Supplementary Material for
Progressive-Scale Blackbox Attack via Projective Gradient Estimation

Jiawei Zhang [*1] **Linyi Li** [*2] **Huichen Li** [2] **Xiaolu Zhang** [3] **Shuang Yang** [4] **Bo Li** [2]

In Appendix A, we show a summary of our theoretical results, compare it with related work, and demonstrate the complete proofs. In Appendix B, we visualize the key characteristics for improving the gradient estimator, and formally justify the existence of the optimal scale. In Appendix C, we show how the target models, i.e., the models we attacked in the experiments, are prepared. In Appendix D, we include more details about our PSBA-PGAN, such as the architecture and training of the projection model, the detailed algorithmic description of the progressive scaling procedure, and the implementation details. In Appendix E, we show additional quantitative experimental results and ablation studies. Finally, in Appendix F, we randomly sample a few original and attacked image pairs to demonstrate the efficiency of our attack compared with other baselines.

A. Theorems and Proofs

This appendix contains a discussion and comparison of theoretical results and all omitted mathematical proofs.

A.1. An outline of Main Theoretical Results

We summarize our main theoretical results—the lower bound of cosine similarity between the estimated gradient and the true gradient, in Table 2.

In the table:

- "Expectation" indicates the bound of the expected cosine similarity;
- "Concentration" indicates the bound of the cosine similarity that holds with probability at least $1-p$;
- "At Boundary" indicates the case where the estimated point is an exact boundary point, i.e., $S_{x^*}(x)=0$;
- "Approaching Boundary" indicates the general case where the estimated point is away from the decision boundary within a small distance θ (measured along the true gradient direction).

"At Boundary" is actually a special case of "Approaching Boudary" with $\theta=0$. In the main text, we only present Theorems 1 and 2 that are for the general case, i.e., "Approaching Boundary" case.

Table 2. A brief summary of the cosine similarity bounds for the boundary gradient estimator in Section 4.1.

	Expectation	Concentration
At Boundary	Theorem 3	Theorem 4
Approaching Boundary	Theorem 1	Theorem 2

A.2. Comparison of Theoretical Results

We compare our theoretical results with existing work in Table 3. Note that it is better to have fewer assumptions and be applicable to more scenarios. As one can observe, our theoretical result is among the most general ones. Furthermore, as discussed in Section 4.1, ours is also among the tightest ones. From these tightest bounds, under general assumptions, we are able to discover the key characteristics and the existence of the optimal scale. The coarse bounds from the previous work cannot reflect these properties.

Table 3. A brief comparison of the cosine similarity bounds for the boundary gradient estimator in our work with existing work.

	Scenario				Assumption				
	At Boundary		Approaching Boundary		Sampling	Projection			
	Expectation	Concentration	Expectation	Concentration	Orthogonal	Identical	Linear	Orthogonal	No Bias
HSJA (Chen et al., 2020)	✓		✓			✓	✓	✓	✓
QEBA (Li et al., 2020a)	✓				✓		✓	✓	✓
NonLinear-BA (Li et al., 2021)	✓				✓				
Ours	✓	✓	✓	✓					✓

A.3. Proof of Lemma 4.1

Lemma 4.1 ($\nabla\boldsymbol{f}$ Decomposition)**.** *Under the assumption in Section 4.1, there exists a singular value decomposition of* $\nabla\boldsymbol{f}(0) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$ *such that*

$$\boldsymbol{U}_{:,1} = \text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)/\|\nabla S(x_t)\|_2 \text{ or } \boldsymbol{U}_{:,1} = -\text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)/\|\nabla S(x_t)\|_2$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{V} \in \mathbb{R}^{m\times m}$ *are orthogonal matrices;* $\boldsymbol{\Sigma} = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathbb{R}^{n\times m}_{\geq 0}$ *is a rectangular diagonal matrix with* $\alpha_1 > 0$.

Proof of Lemma 4.1. For simplicity, we define $\boldsymbol{M} := \nabla\boldsymbol{f}(0)$. According to the assumption, there exists a column vector $\boldsymbol{M}_{:,c}$ that is not orthogonal with the gradient direction $\nabla S(x_t)$. If $c \neq 1$, we define $\boldsymbol{T} := [e_c\, e_2\, \cdots\, e_{c-1}\, e_1\, e_{c+1}\, \cdots\, e_m]$; otherwise, we let $\boldsymbol{T} := \boldsymbol{I}_m$. Here, $e_i \in \mathbb{R}^m$ is a standard basis vector, i.e., it satisfies $(e_i)_i = 1$ and $(e_i)_j = 0$ for any $j \neq i$. As the result,

$$\boldsymbol{M} = [\boldsymbol{M}_{:,c}\, \boldsymbol{M}_{:,2}\, \boldsymbol{M}_{:,3}\, \cdots\, \boldsymbol{M}_{:,c-1}\, \boldsymbol{M}_{:,1}\, \boldsymbol{M}_{:,c+1}\, \cdots \boldsymbol{M}_{:,m}]\boldsymbol{T} := \boldsymbol{M}'\boldsymbol{T}.$$

Here $\boldsymbol{M}_{:,i}$ stands for the i-th column vector of $\boldsymbol{M}$. $\boldsymbol{M}'$ just exchanges the first column vector with the c-th column vector of $\boldsymbol{M}$. According to the assumption in Section 4.1, $\boldsymbol{M}'_{:,1}$ is aligned with $\text{proj}_{\boldsymbol{M}}\nabla(x_t)$, and for $i \geq 2$, we have $\langle \boldsymbol{M}'_{:,i}, \boldsymbol{M}'_{:,1}\rangle = 0$.

Now, we apply QR decomposition to $\boldsymbol{M}'$ via Gram-Schmidt Process, which yields $\boldsymbol{M}' = \boldsymbol{U}'\boldsymbol{R}$, where $\boldsymbol{U}' \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, and $\boldsymbol{R} \in \mathbb{R}^{n\times m}$ is an upper-triangular matrix. We are going to show two interesting properties of $\boldsymbol{U}'$ and $\boldsymbol{R}$: 1) $\boldsymbol{U}'_{:,1} = \frac{\text{proj}_{\boldsymbol{M}}\nabla S(x_t)}{\|\text{proj}_{\boldsymbol{M}}\nabla S(x_t)\|_2}$ or $\boldsymbol{U}'_{:,1} = -\frac{\text{proj}_{\boldsymbol{M}}\nabla S(x_t)}{\|\text{proj}_{\boldsymbol{M}}\nabla S(x_t)\|_2}$; and 2) $\boldsymbol{R}$ can be written as $\begin{bmatrix}\alpha_1 & 0\\ 0 & \boldsymbol{R}'\end{bmatrix}$ where $\alpha_1 > 0$ and $\boldsymbol{R}'$ is an upper-triangular matrix. The first property is apparent, since in Gram-Schmidt Process, we always have $\boldsymbol{U}'_{:,1} = \boldsymbol{M}'_{:,1}/\|\boldsymbol{M}'_{:,1}\|_2$. Thus, it is equal to $\pm\nabla\text{proj}_{\boldsymbol{M}}S(x_t)/\|\text{proj}_{\boldsymbol{M}}\nabla S(x_t)\|_2$. For the second property, according to the definition of the process, $(\boldsymbol{R})_{1,i} = \langle \boldsymbol{M}'_{:,1}, \boldsymbol{M}'_{:,i}\rangle = 0$. Meanwhile, $\alpha_1 = \|\boldsymbol{M}'_{:,1}\|_2^2 > 0$ since $\boldsymbol{M}'_{:,1}$ aligns with $\nabla S(x_t)$ and it is non-zero.

We apply SVD decomposition to the sub-matrix $\boldsymbol{R}' \in \mathbb{R}^{(n-1)\times(m-1)}$: $\boldsymbol{R}' := \boldsymbol{S}'\boldsymbol{\Sigma}'\boldsymbol{W}'^\intercal$. Here, $\boldsymbol{S}' \in \mathbb{R}^{(n-1)\times(n-1)}$ and $\boldsymbol{W}' \in \mathbb{R}^{(m-1)\times(m-1)}$ are orthogonal matrices, while $\boldsymbol{\Sigma}' \in \mathbb{R}^{(n-1)\times(m-1)}$ is a triangular diagonal matrix. Therefore, $\boldsymbol{R}$ can be decomposed as such:

$$\boldsymbol{R} = \begin{bmatrix}\alpha_1 & 0\\ 0 & \boldsymbol{R}'\end{bmatrix} = \overbrace{\begin{bmatrix}1 & 0\\ 0 & \boldsymbol{S}'\end{bmatrix}}^{\boldsymbol{S}}\overbrace{\begin{bmatrix}\alpha_1 & 0\\ 0 & \boldsymbol{\Sigma}'\end{bmatrix}}^{\boldsymbol{\Sigma}}\overbrace{\begin{bmatrix}1 & 0\\ 0 & \boldsymbol{W}'^\intercal\end{bmatrix}}^{\boldsymbol{W}^\intercal}.$$

It is easy to observe that $\boldsymbol{\Sigma} = \text{diag}(\alpha_1, \alpha_2, \cdots, \alpha_m)$ is a rectangular diagonal matrix with $\alpha_1 > 0$, and $\boldsymbol{V}$ is an orthogonal matrix. Notice that

$$\begin{aligned}\nabla\boldsymbol{f}(0) = \boldsymbol{M} = \boldsymbol{M}'\boldsymbol{T} &= \boldsymbol{U}'\boldsymbol{R}\boldsymbol{T} = \boldsymbol{U}'\boldsymbol{S}\boldsymbol{\Sigma}\boldsymbol{W}^\intercal\boldsymbol{T}\\ &= \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{U}'_{:,1} & \boldsymbol{U}'_{:,2} & \cdots & \boldsymbol{U}'_{:,m} \\ | & | & \cdots & | \end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \boldsymbol{S}'\end{bmatrix}\boldsymbol{\Sigma}\boldsymbol{W}^\intercal\boldsymbol{T}\\ &= \underbrace{\begin{bmatrix}\boldsymbol{U}'_{:,1} & \boldsymbol{U}'_{:,2:m}\boldsymbol{S}'\end{bmatrix}}_{\boldsymbol{U}}\boldsymbol{\Sigma}\underbrace{\boldsymbol{W}^\intercal\boldsymbol{T}}_{\boldsymbol{V}^\intercal}.\end{aligned}$$

We have already shown $\boldsymbol{U}'_{:,1} = \frac{\text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)}{\|\text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}$ or $\boldsymbol{U}'_{:,1} = -\frac{\text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)}{\|\text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}$. To finish the proof, we only need to verify that $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices. Since $\boldsymbol{U}'$ and $\boldsymbol{S}'$ are both orthogonal matrices, the $\boldsymbol{U}'_{:,2:m}\boldsymbol{S}'$ is a semi-orthogonal matrix (the column vectors are unitary and orthogonal). Furthermore, $\boldsymbol{U}'_{:,1} \perp \mathrm{span}(\boldsymbol{U}'_{:,2:m})$ because $\boldsymbol{U}'$ is an orthogonal matrix. Thus, $\boldsymbol{U}'_{:,1} \perp \mathrm{span}(\boldsymbol{U}'_{:,2:m}\boldsymbol{S}')$. As the result, $\boldsymbol{U}$ is an orthogonal matrix. For $\boldsymbol{V}$, $\boldsymbol{V}^\intercal \boldsymbol{V} = \boldsymbol{W}^\intercal \boldsymbol{T}\boldsymbol{T}^\intercal \boldsymbol{W} = \boldsymbol{I}_m$ so it is an orthogonal matrix. □

A.4. Warmup: Expectation Bound at Boundary

As a warm-up, we begin with the special case where the point is exactly the boundary point.

The proof of the following theorems require the following lemma.

Lemma A.1 (Cosine Similarity in Projected Subspace)**.** *Let $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ be a matrix. The vector $w \in \mathbb{R}^n$ is in* $\mathrm{span}(\boldsymbol{W})$*, and the vector $v \in \mathbb{R}^n$ has non-zero projection in* $\mathrm{span}(\boldsymbol{W})$*, i.e.,* $\mathrm{proj}_{\boldsymbol{W}} v \neq 0$*. Then,*

$$\cos\langle w, v\rangle = \cos\langle w, \mathrm{proj}_{\boldsymbol{W}} v\rangle \cdot \frac{\|\mathrm{proj}_{\boldsymbol{W}} v\|_2}{\|v\|_2}. \tag{5}$$

Proof of Lemma A.1. The lemma can be illustrated by simple geometry. To be rigorous, here we give an algebraic proof.

$$\cos\langle w, v\rangle = \frac{\langle w, v\rangle}{\|w\|_2\|v\|_2} = \frac{\langle w, v\rangle}{\|w\|_2\|\mathrm{proj}_{\boldsymbol{W}} v\|_2} \cdot \frac{\|\mathrm{proj}_{\boldsymbol{W}} v\|_2}{\|v\|_2}.$$

We notice that $w \in \mathrm{span}(w)$, and $v = \mathrm{proj}_{\boldsymbol{W}} v + (v - \mathrm{proj}_{\boldsymbol{W}} v)$ where $(v - \mathrm{proj}_{\boldsymbol{W}} v)$ is orthogonal to w. Thus,

$$\langle w, v\rangle = \langle w, \mathrm{proj}_{\boldsymbol{W}} v\rangle.$$

So

$$\cos\langle w, v\rangle = \cos\langle w, \mathrm{proj}_{\boldsymbol{W}} v\rangle \cdot \frac{\|\mathrm{proj}_{\boldsymbol{W}} v\|_2}{\|v\|_2}.$$

□

Remark. The lemma reveals that for any vector w in $\mathrm{span}(\boldsymbol{W})$, the maximum possible cosine similarity between w and v is $\|\mathrm{proj}_{\boldsymbol{W}} v\|_2/\|v\|_2$. This is achieved by setting $w = k \cdot \mathrm{proj}_{\boldsymbol{W}} v$.

Theorem 3 (Expected cosine similarity; at boundary)**.** *The difference function S and the projection $\boldsymbol{f}$ are as defined before. For a boundary point x_t such that $S(x_t) = 0$, let estimated gradient $\widetilde{\nabla S}(x_t)$ be as computed by Definition 2 with step size δ and sampling size B. Over the randomness of the sampled vectors $\{u_b\}_{i=1}^B$,*

$$\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \geq \frac{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2} \cdot \left(1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \sqrt{\frac{1}{m-1}}1.58\beta_{\boldsymbol{f}}\right)^2\right), \tag{6}$$

where

$$\gamma := \beta_{\boldsymbol{f}} + \frac{\beta_S\left(\max_{i\in[m]}\alpha_i + {}^1\!/\!_2\,\delta\beta_{\boldsymbol{f}}\right)^2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}. \tag{7}$$

Proof of Theorem 3. We begin with an important lemma.

Lemma A.2. *We let*

$$w := \frac{1}{2}\delta\left(\beta_{\boldsymbol{f}}\|\nabla S(x_t)\|_2 + \beta_S\left(\|\nabla \boldsymbol{f}(0)\|_2 + \frac{1}{2}\delta\beta_{\boldsymbol{f}}\right)^2\right). \tag{8}$$

On the point x_t such that $S(x_t) = 0$, for any $\delta > 0$ and unit vector $u \in \mathbb{R}^n$,

$$\langle \nabla S(x_t),\, \nabla \boldsymbol{f}(0)\cdot u\rangle > w \Longrightarrow \phi(x_t + \Delta \boldsymbol{f}(\delta u)) = 1,$$
$$\langle \nabla S(x_t),\, \nabla \boldsymbol{f}(0)\cdot u\rangle < -w \Longrightarrow \phi(x_t + \Delta \boldsymbol{f}(\delta u)) = -1.$$

Proof of Lemma A.2. We prove the lemma by Taylor expansion and the smoothness condition on S and $\boldsymbol{f}$. First, from Taylor expansion on function $\Delta f(\delta u)$ at the origin,

$$\Delta \boldsymbol{f}(\delta u) = \boldsymbol{f}(\delta u) - \boldsymbol{f}(0) = \nabla \boldsymbol{f}(0) \cdot (\delta u) + \nicefrac{1}{2}(\delta u)^\intercal \nabla^2 \boldsymbol{f}(\xi)(\delta u), \tag{9}$$

where ξ is a point on the segment between the origin and (δu). Since $\boldsymbol{f}$ is $\beta_{\boldsymbol{f}}$-smooth, $\|\nicefrac{1}{2}(\delta u)^\intercal \nabla^2 \boldsymbol{f}(\xi)(\delta u)\|_2 \le \nicefrac{1}{2}\beta_{\boldsymbol{f}}\delta^2$. Thus,

$$\langle \nabla S(x_t), \Delta \boldsymbol{f}(\delta u)\rangle \in \delta \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u\rangle \pm \nicefrac{1}{2}\delta^2 \|\nabla S(x_t)\|_2 \beta_{\boldsymbol{f}}.$$

Also we have

$$\|\Delta \boldsymbol{f}(\delta u)\|_2 \le \|\nabla \boldsymbol{f}(0) \cdot (\delta u)\|_2 + \|\nicefrac{1}{2}(\delta u)^\intercal \nabla^2 \boldsymbol{f}(\xi)(\delta u)\|_2 \le \delta \|\nabla \boldsymbol{f}(0)\|_2 + \nicefrac{1}{2}\beta_{\boldsymbol{f}}\delta^2.$$

Easily seen, this also applies to any point ξ' between the origin and (δu). Now, we apply Taylor expansion on function $S\left(x_t + \Delta \boldsymbol{f}(\delta u)\right)$ at point x_t, and get

$$\begin{aligned}
S(x_t + \Delta \boldsymbol{f}(\delta u)) &= \overbrace{S(x_t)}^{0} + \langle \nabla S(x_t), \Delta \boldsymbol{f}(\delta u)\rangle + \nicefrac{1}{2}(\Delta \boldsymbol{f}(\xi'))^\intercal \nabla^2 S(x_t)(\Delta \boldsymbol{f}(\xi')) \\
&\in \delta \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u\rangle \pm \nicefrac{1}{2}\delta^2 \|\nabla S(x_t)\|_2 \beta_{\boldsymbol{f}} \pm \nicefrac{1}{2}\beta_S \left(\delta \|\nabla \boldsymbol{f}(0)\|_2 + \nicefrac{1}{2}\beta_{\boldsymbol{f}}\delta^2\right)^2 \\
&= \delta \left(\langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u\rangle \pm \nicefrac{1}{2}\delta \left(\|\nabla S(x_t)\|_2 \beta_{\boldsymbol{f}} + \beta_S \left(\|\nabla \boldsymbol{f}(0)\|_2 + \nicefrac{1}{2}\delta\beta_{\boldsymbol{f}}\right)^2 \right)\right) \\
&= \delta \left(\langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u\rangle \pm w \right).
\end{aligned}$$

Since the step size $\delta > 0$,

$$\begin{aligned}
\langle \nabla S(x_t),\, \nabla \boldsymbol{f}(0) \cdot u\rangle > w &\Longrightarrow S(x_t + \Delta \boldsymbol{f}(\delta u)) > 0, \\
\langle \nabla S(x_t),\, \nabla \boldsymbol{f}(0) \cdot u\rangle < -w &\Longrightarrow S(x_t + \Delta \boldsymbol{f}(\delta u)) < 0.
\end{aligned}$$

Observing that ϕ is the sign function of S according to Definition 1, we conclude the proof. □

Remark. This lemma shows the connection between the value of sign function and the direction alignment between $\nabla \boldsymbol{f}(0) \cdot u$ and the true gradient.

Then, we study the distribution of $\nabla \boldsymbol{f}(0) \cdot u$.

Lemma A.3.

$$\langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u\rangle \sim \alpha_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 \left(2\mathrm{Beta}\left(\frac{m-1}{2}, \frac{m-1}{2}\right) - 1\right). \tag{10}$$

Meanwhile, for any $c \in [-\|\nabla S(x_t)\|_2, +\|\nabla S(x_t)\|_2]$,

$$\mathbb{E}\left[\nabla \boldsymbol{f}(0) \cdot u \,|\, \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u\rangle = c\right] = c \frac{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2}. \tag{11}$$

Proof of Lemma A.3. According to Lemma 4.1, $\nabla \boldsymbol{f}(0) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$. Since $\boldsymbol{V}$ is an orthongonal basis of $\mathbb{R}^m$, and u is uniformly sampled from the uniform sphere S^{m-1}, we let $v = \boldsymbol{V}^\intercal u$ and $v \sim \mathrm{Unif}(S^{m-1})$ too. Then,

$$\begin{aligned}
\langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u\rangle &= \langle \nabla S(x_t), \boldsymbol{U}\boldsymbol{\Sigma} v\rangle = \left\langle \nabla S(x_t), \sum_{i=1}^{m} \alpha_i v_i \boldsymbol{U}_{:,i} \right\rangle \\
&= \alpha_1 v_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 \left\langle \frac{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}, \boldsymbol{U}_{:,1} \right\rangle \overset{(*)}{=} \pm \alpha_1 v_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2,
\end{aligned} \tag{12}$$

where $(*)$ follows from $\boldsymbol{U}$ is an orthogonal basis with $\boldsymbol{U}_{:,1} = \pm \nabla \mathrm{proj}_{\nabla \boldsymbol{f}(0)} S(x_t) / \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2$ as Lemma 4.1 shows.

From (Yang et al., 2020) (Lemma I.23), $\frac{1+v_1}{2} \sim \mathrm{Beta}\left(\frac{m-1}{2}, \frac{m-1}{2}\right)$, where $\mathrm{Beta}(\cdot,\cdot)$ stands for the Beta distribution. As the result,

$$\alpha_1 v_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 \sim \alpha_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 \left(2\mathrm{Beta}\left(\frac{m-1}{2}, \frac{m-1}{2}\right) - 1\right).$$

Observing that this is a symmetric distribution centered at 0, we have

$$\langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle \sim \alpha_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 \left(2\mathrm{Beta}\left(\frac{m-1}{2}, \frac{m-1}{2} \right) - 1 \right),$$

which proves the first part of the lemma.

For the second part, hereinafter, we condition the distribution of u on $\langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle = c$. According to Eq. (12), the condition means that

$$v_1 = c_1 := \frac{c}{\alpha_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} \left\langle \frac{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}, \boldsymbol{U}_{:,1} \right\rangle.$$

Here we define the constant c_1. Since $v \sim \mathrm{Unif}(S^{m-1})$, it means that under this condition, $v' = (v_2, v_3, \cdots, v_m)$ is uniformly sampled from the $(m-1)$-dimension hypersphere with radius $r = \sqrt{1 - c_1^2}$. Therefore,

$$\begin{aligned} & \mathbb{E}[\nabla \boldsymbol{f}(0) \cdot u \,|\, \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle = c] = \mathbb{E}[\boldsymbol{U\Sigma} v \,|\, v_1 = c_1] \\ = & \mathbb{E}\left[\alpha_1 v_1 \boldsymbol{U}_{:,1} + \sum_{i=2}^{m} \alpha_i v_i \boldsymbol{U}_{:,i} \,\middle|\, v_1 = c_1 \right] = \alpha_1 c_1 \boldsymbol{U}_{:,1} + \sum_{i=2}^{m} \mathbb{E}\left[\alpha_i v_i \boldsymbol{U}_{:,i} | v_1 = c_1 \right] \\ = & c \frac{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2} + \sum_{i=2}^{m} \alpha_i \boldsymbol{U}_{:,i} \cdot \mathbb{E}[v_i | v_1 = c_1]. \end{aligned} \tag{13}$$

Since under this condition, $v' = (v_2, \cdots, v_m)$ is uniformly sampled from a hypersphere centered at the origin (with radius r), we have $\mathbb{E}[v_i | v_1 = c_1] = 0$ for $i \geq 2$ by the symmetry of Beta distribution. Thus,

$$\mathbb{E}[\nabla \boldsymbol{f}(0) \cdot u \,|\, \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle = c] = c \frac{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2}.$$

□

Remark. The lemma considers the distribution of sampled vector u after the transformation by $\boldsymbol{f}$ approximated in the first-order. The first equation of the lemma, Eq. (10), reveals the distribution of the projection (dot product) onto the true gradient direction. The distribution is a linearly scaled Beta distribution. The second equation of the lemma, Eq. (11), reveals that the sampled vector is *unbiased* on any direction orthogonal to the true gradient, i.e., conditioned on the same projected length on the true gradient direction, the expectation of the sampled vector aligns with the projected true gradient direction without any directional bias.

According to Lemma 4.1, we write $\nabla \boldsymbol{f}(0) = \boldsymbol{U\Sigma V}^\intercal$. For notation simplicity, we let $\widehat{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t)$ denote the normalized true gradient: $\widehat{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t) := \mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t) / \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2$. Furthermore, we let $s := \langle \widehat{\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t), \boldsymbol{U}_{:,1} \rangle \in \{\pm 1\}$ denote the sign between these two aligned vectors.

With respect to the randomness of $u \sim \mathrm{Unif}(S^{m-1})$, we denote v to $\boldsymbol{V}^\intercal u \sim \mathrm{Unif}(S^{m-1})$, and we define the following three events E^-, E^o, and E^+:

$$E^- : \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle \in (-\infty, -w), \tag{14}$$

$$E^o : \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle \in [-w, +w], \tag{15}$$

$$E^+ : \langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle \in (+w, +\infty). \tag{16}$$

From Lemma A.3, we denote p to $\Pr[E^o]$, and by the symmetry of Beta distribution, $\Pr[E^-] = \Pr[E^+] = (1-p)/2$. From Eq. (12), we know $\langle \nabla S(x_t), \nabla \boldsymbol{f}(0) \cdot u \rangle = \alpha_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 s v_1$. Therefore, with events E^- and E^+, $|v_1| > \frac{w}{\alpha_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}$ while the signs of v_1 are different between the two events; and with event E^o, $|v_1| \leq \frac{w}{\alpha_1 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}$.

According to the definition of the gradient estimator in Definition 2, we have $\mathbb{E}\widetilde{\nabla S}(x_t) = \mathbb{E}[\phi(x_t + \Delta \boldsymbol{f}(\delta u)) \Delta \boldsymbol{f}(\delta u)]$. According to Eq. (9), for any unit vector u,

$$\phi(x_t + \Delta \boldsymbol{f}(\delta u)) \Delta \boldsymbol{f}(\delta u) = \phi(x_t + \Delta \boldsymbol{f}(\delta u)) \left(\delta \nabla \boldsymbol{f}(0) \cdot u + \xi_{\delta u} \right) = \delta \phi(x_t + \Delta \boldsymbol{f}(\delta u)) \nabla \boldsymbol{f}(0) \cdot u + \xi'_{\delta u}$$

where $\xi_{\delta u}$ and $\xi'_{\delta u}$ are vectors depended by δu with length $\leq 1/2\beta_{\boldsymbol{f}}\delta^2$. Therefore,

$$\|\mathbb{E}\widetilde{\nabla S}(x_t) - \delta\mathbb{E}[\phi(x_t + \nabla \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0) \cdot u]\|_2 \leq 1/2\beta_{\boldsymbol{f}}\delta^2. \tag{17}$$

Now, we inspect $\mathbb{E}[\phi(x_t + \nabla \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0) \cdot u]$:

$$\mathbb{E}[\phi(x_t+\nabla \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0)\cdot u] = \overbrace{p\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0) \cdot u \,|\, E^o\right]}^{(*)} + \overbrace{\frac{1-p}{2}\left(\mathbb{E}\left[-\nabla \boldsymbol{f}(0) \cdot u \,|\, E^-\right] + \mathbb{E}\left[\nabla \boldsymbol{f}(0) \cdot u \,|\, E^+\right]\right)}^{(**)}. \tag{18}$$

According to Lemma A.3 (Eq. (11)),

$$\mathbb{E}[\nabla \boldsymbol{f}(0) \cdot u \,|\, E^+] = \mathbb{E}[|v_1| \,|\, E^+]\alpha_1\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t),\ \mathbb{E}[\nabla \boldsymbol{f}(0) \cdot u \,|\, E^-] = \mathbb{E}[-|v_1| \,|\, E^-]\alpha_1\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t).$$

By symmetry of Beta distribution, $\mathbb{E}\left[|v_1| \,|\, E^+\right] = \mathbb{E}\left[|v_1| \,|\, E^-\right]$, and

$$(**) = \frac{1-p}{2}\left(\mathbb{E}\left[-\nabla \boldsymbol{f}(0) \cdot u \,|\, E^-\right] + \mathbb{E}\left[\nabla \boldsymbol{f}(0) \cdot u \,|\, E^+\right]\right) = (1-p)\alpha_1\mathbb{E}\left[|v_1| \,|\, E^+\right]\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t).$$

For $(*)$, we notice that

$$\begin{aligned}\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0) \cdot u \,|\, E^o\right] &\overset{\text{Eq. 13}}{=} \mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))\left(\alpha_1 v_1 s \cdot \widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t) + \sum_{i=2}^m \alpha_i v_i \boldsymbol{U}_{:,i}\right) \,\middle|\, E^o\right] \\ &= \alpha_1 s\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_1 \,|\, E^o\right]\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t) + \sum_{i=2}^m \alpha_i\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_i \,|\, E^o\right]\boldsymbol{U}_{:,i}.\end{aligned}$$

Combining them with Eq. (18), we have

$$\begin{aligned}\mathbb{E}[\phi(x_t + \nabla \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0) \cdot u] =&\alpha_1\left(ps\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_1 \,|\, E^o\right] + (1-p)\mathbb{E}\left[|v_1| \,|\, E^+\right]\right)\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t) \\ &+ p\sum_{i=2}^m \alpha_i\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_i \,|\, E^o\right]\boldsymbol{U}_{:,i}.\end{aligned}$$

We notice that $\{\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t), \boldsymbol{U}_{:,2}, \cdots, \boldsymbol{U}_{:,m}\}$ is an orthogonal basis of $\mathbb{R}^n$. Thus,

$$\begin{aligned}&\left\|\mathbb{E}[\phi(x_t + \nabla \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0) \cdot u] - \alpha_1\mathbb{E}\left[|v_1|\right]\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t)\right\|_2 \\ =&\left\|\alpha_1\left(ps\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_1 \,|\, E^o\right] + (1-p)\mathbb{E}\left[|v_1| \,|\, E^+\right] - \mathbb{E}\left[|v_1|\right]\right)\widehat{\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S}(x_t)\right. \\ &\qquad\left. + p\sum_{i=2}^m \alpha_i\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_i \,|\, E^o\right]\boldsymbol{U}_{:,i}\right\|_2 \\ =&\sqrt{\alpha_1^2\underbrace{\left(ps\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_1 \,|\, E^o\right] + (1-p)\mathbb{E}\left[|v_1| \,|\, E^+\right] - \mathbb{E}\left[|v_1|\right]\right)^2}_{\text{(I)}} + \underbrace{\sum_{i=2}^m \alpha_i^2 p^2\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_i \,|\, E^o\right]^2}_{\text{(II)}}.}\end{aligned}$$

We bound the two terms (I) and (II) individually. For the first term, we notice that

$$\begin{aligned}&\left|ps\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_1 \,|\, E^o\right] + (1-p)\mathbb{E}\left[|v_1| \,|\, E^+\right] - \mathbb{E}\left[|v_1|\right]\right| \\ \leq& p\left|\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_1 \,|\, E^o\right]\right| + \left|(1-p)\mathbb{E}\left[|v_1| \,|\, E^+\right] - (1-p)\mathbb{E}\left[|v_1| \,|\, E^+\right] - p\mathbb{E}\left[|v_1| \,|\, E^o\right]\right| \\ \leq& p\mathbb{E}\left[|v_1| \,|\, E^o\right] + p\mathbb{E}\left[|v_1| \,|\, E^o\right] = 2p\mathbb{E}\left[|v_1| \,|\, E^o\right] \leq \frac{2pw}{\alpha_1\|\text{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2}.\end{aligned}$$

For the second term, we have

$$\text{(II)} = p^2\sum_{i=2}^m \alpha_i^2\mathbb{E}\left[\phi(x_t + \nabla \boldsymbol{f}(\delta u))v_i \,|\, E^o\right]^2 \leq p^2\sum_{i=2}^m \alpha_i^2\mathbb{E}\left[|v_i| \,|\, E^o\right]^2 \leq p^2\sum_{i=2}^m \alpha_i^2\mathbb{E}\left[v_i^2 \,|\, E^o\right].$$

Since $v = (v_1, v_2, \dots, v_m)^\intercal$ is uniformly sampled from S^{m-1}, conditioned on every sampled v_1, the vector $(v_2, \dots, v_m)^\intercal$ is uniformly sampled from the $\sqrt{1-v_1^2}S^{m-2}$. We let $v' = (v_2', \dots, v_m')^\intercal$ be uniformly sampled from S^{m-2}. Thus, for every sampled v_1, we always have $\mathbb{E}\left[v_i^2 \,|\, v_1\right] \le \mathbb{E}\left[{v_i'}^2\right]$ because v' is sampled from a larger hypersphere. By stacking all sampled v_1's that forms the event E^o, we have (II) $\le p^2 \sum_{i=2}^m \alpha_i^2 \mathbb{E}\left[{v_i'}^2\right]$. According to (Yang et al., 2020), $\frac{1+v_i'}{2} \sim$ Beta $\left(\frac{m}{2}-1, \frac{m}{2}-1\right)$, whose variance Var $\left(\frac{1+v_i'}{2}\right) = \frac{1}{4(m-1)}$. Since Var $\left(\frac{1+v_i'}{2}\right) = \mathbb{E}\left[\left(\frac{1+v_i'}{2}\right)^2\right] - \mathbb{E}\left[\frac{1+v_i'}{2}\right]^2 = \frac{1}{4}\mathbb{E}\left[{v_i'}^2\right]$, we have $\mathbb{E}\left[{v_i'}^2\right] = \frac{1}{m-1}$. Thus,

$$\text{(II)} \le p^2 \sum_{i=2}^m \alpha_i^2 \mathbb{E}\left[{v_i'}^2\right] = \frac{1}{m-1} p^2 \sum_{i=2}^m \alpha_i^2.$$

As a result,

$$\left\|\mathbb{E}[\phi(x_t + \Delta \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0)\cdot u] - \alpha_1 \mathbb{E}\left[|v_1|\right] \widehat{\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t)\right\|_2 \le \sqrt{\frac{4p^2w^2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2} + \frac{p^2\sum_{i=2}^m \alpha_i^2}{m-1}}. \tag{19}$$

Combining Eq. (19) with Eq. (17), we have

$$\|\mathbb{E}\widetilde{\nabla S}(x_t) - \delta\alpha_1 \mathbb{E}\left[|v_1|\right] \widehat{\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t)\|_2 \le p\delta\sqrt{\frac{4w^2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2} + \frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + {}^1\!/_2 \beta_{\boldsymbol{f}}\delta^2. \tag{20}$$

It means

$$\begin{aligned}
&\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\rangle \\
=&\cos\left\langle \mathbb{E}\widetilde{\nabla S}(x_t),\ \delta\alpha_1 \mathbb{E}\left[|v_1|\right] \widehat{\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t)\right\rangle \\
\ge&1 - \frac{1}{2}\left(\frac{p\delta\sqrt{\frac{4w^2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2} + \frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \frac{1}{2}\beta_{\boldsymbol{f}}\delta^2}{\delta\alpha_1 \mathbb{E}\left[|v_1|\right]}\right)^2 \qquad (21)\\
\ge&1 - \frac{1}{2}\left(\frac{2pw}{\alpha_1 \mathbb{E}\left[|v_1|\right] \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{p}{\alpha_1 \mathbb{E}\left[|v_1|\right]}\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \frac{\delta\beta_{\boldsymbol{f}}}{2\alpha_1 \mathbb{E}\left[|v_1|\right]}\right)^2.
\end{aligned}$$

To this point, we need to unfold p and $\mathbb{E}\left[|v_1|\right]$. From the definition of event E^o(Eq. (15)),

$$p = \Pr\left[-w/(\alpha_1\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2) \le v_1 \le w/(\alpha_1\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2)\right] = \Pr\left[v_1^2 \le w^2/(\alpha_1^2\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|^2)\right],$$

where $v_1^2 \sim \text{Beta}(1/2, (m-1)/2)$ (Chen et al., 2020). Thus, let $B(\cdot,\cdot)$ be the Beta function,

$$p = \int_0^{\frac{w^2}{\alpha_1^2\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2}} \frac{x^{-1/2}(1-x)^{\frac{m-3}{2}}}{B\left(\frac{1}{2}, \frac{m-1}{2}\right)} \mathrm{d}x \le \frac{2w}{B(\frac{1}{2}, \frac{m-1}{2})\alpha_1\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}. \tag{22}$$

Also, from (Li et al., 2021) (Lemma 1), we have

$$\mathbb{E}\left[|v_1|\right] = 2\int_0^1 \frac{x(1-x^2)^{\frac{m-3}{2}}}{B(\frac{1}{2}, \frac{m-1}{2})}\mathrm{d}x = \frac{2}{(m-1)\cdot B(\frac{1}{2}, \frac{m-1}{2})}. \tag{23}$$

Plugging them into Eq. (21):

$$\begin{aligned}
&\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\rangle \ge \\
&1 - \frac{1}{2}\left(\frac{m-1}{\alpha_1^2\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}\left(\frac{2w^2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + w\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}}\right) + \frac{\delta\beta_{\boldsymbol{f}}(m-1)B(\frac{1}{2}, \frac{m-1}{2})}{4\alpha_1}\right)^2. \qquad (24)
\end{aligned}$$

We apply Stirling's approximation with error bound to $(m-1)B(\frac{1}{2}, \frac{m-1}{2})$:

$$\sqrt{2\pi(m-1)} \le (m-1)B\left(\frac{1}{2}, \frac{m-1}{2}\right) \le 1.26\sqrt{2\pi(m-1)} \text{ for all } m \ge 1,$$

and plug w (Eq. (8)) in (we also replace $\|\nabla \boldsymbol{f}(0)\|_2$ by $\max_{i\in[m]} \alpha_i$):

$$\begin{aligned}
&\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\rangle \\
\ge& 1 - \frac{1}{2}\left(\frac{m-1}{\alpha_1^2 \|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}\left(\frac{2w^2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + w\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}}\right) + 0.315\sqrt{2\pi(m-1)}\frac{\delta\beta_{\boldsymbol{f}}}{\alpha_1}\right)^2 \\
=& 1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta}{\alpha_1}\left(\frac{\beta_{\boldsymbol{f}}\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 + \beta_S\left(\max_{i\in[m]}\alpha_i + {}^1\!/\!_2\delta\beta_{\boldsymbol{f}}\right)^2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}\right)^2\right. \\
&\left. + \frac{1}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}}\frac{\beta_{\boldsymbol{f}}\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2 + \beta_S\left(\max_{i\in[m]}\alpha_i + {}^1\!/\!_2\delta\beta_{\boldsymbol{f}}\right)^2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + 0.63\sqrt{\frac{2\pi}{m-1}}\beta_{\boldsymbol{f}}\right)^2 \\
\ge& 1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \sqrt{\frac{1}{m-1}}1.58\beta_{\boldsymbol{f}}\right)^2.
\end{aligned} \tag{25}$$

According to Lemma A.1, we have

$$\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \ge \frac{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2} \cdot \left(1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \sqrt{\frac{1}{m-1}}1.58\beta_{\boldsymbol{f}}\right)^2\right). \tag{26}$$

□

A.5. Warmup: Concentration Bound at Boundary

Now we consider the concentration bound for the boundary point gradient estimation.

Theorem 4 (Concentration of cosine similarity; at boundary)**.** *Under the same setting as Theorem 3, over the randomness of the sampled vector $\{u_b\}_{i=1}^B$, with probability $1-p$,*

$$\begin{aligned}
&\cos\langle \widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \\
\ge& \frac{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2} \cdot \left(1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \frac{1.58\beta_{\boldsymbol{f}} + \frac{1}{\delta}\sqrt{\sum_{i=1}^m \alpha_i^2} \cdot \sqrt{\frac{2}{B}\ln(\frac{m}{\epsilon})}}{\sqrt{m-1}}\right)^2\right),
\end{aligned} \tag{27}$$

where

$$\gamma := \beta_{\boldsymbol{f}} + \frac{\beta_S\left(\max_{i\in[m]}\alpha_i + {}^1\!/\!_2\delta\beta_{\boldsymbol{f}}\right)^2}{\|\nabla S(x_t)\|_2}. \tag{28}$$

Proof of Theorem 4. The key idea for the proof is to bound the ℓ_2 distance from the estimated gradient vector to the expectation of the estimated gradient vector via concentration bounds. Then, the bound is combined to the proof of Theorem 3, concretely, Eq. (19), to derive the required result.

In the proof, to distinguish the "p" in probability bound from the definition in Eq. (15), we change this variable to "ϵ", i.e., the bound reads "with probability $1-\epsilon, \ldots$".

We note that the gradient estimator per Definition 2 is

$$\widetilde{\nabla S}(x_t) = \frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))\Delta f(\delta u_b)$$

where $u_b \sim \text{Unif}(S^{m-1})$. Thus, let $u \sim \text{Unif}(S^{m-1})$,

$$\begin{aligned}
&\left\|\widetilde{\nabla S}(x_t) - \delta\mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u\right]\right\|_2 \\
\overset{(i.)}{=}&\left\|\frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))\nabla\boldsymbol{f}(0)\cdot(\delta u_b) + \frac{1}{B}\sum_{b=1}^{B}\xi_{\delta u_b} - \delta\mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u\right]\right\|_2 \\
\le&{}^1\!/_2\beta_{\boldsymbol{f}}\delta^2 + \delta\left\|\frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))\nabla\boldsymbol{f}(0)\cdot u_b - \mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u\right]\right\|_2 \\
=&{}^1\!/_2\beta_{\boldsymbol{f}}\delta^2 + \delta\left\|\frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))\boldsymbol{U\Sigma}v_b - \mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\boldsymbol{U\Sigma}v\right]\right\|_2 && \text{(Lemma 4.1, } v = \boldsymbol{V}^\intercal u\text{)} \\
=&{}^1\!/_2\beta_{\boldsymbol{f}}\delta^2 + \delta\left\|\frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{m}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))\boldsymbol{U}_{:,i}\alpha_i v_{b,i} - \sum_{i=1}^{m}\mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\boldsymbol{U}_{:,i}\alpha_i v_i\right]\right\|_2 \\
=&{}^1\!/_2\beta_{\boldsymbol{f}}\delta^2 + \delta\left\|\sum_{i=1}^{m}\alpha_i\boldsymbol{U}_{:,i}\left(\frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))v_{b,i} - \mathbb{E}[\phi(x_t + \Delta\boldsymbol{f}(\delta u))v_i]\right)\right\|_2 \\
\overset{(ii.)}{=}&{}^1\!/_2\beta_{\boldsymbol{f}}\delta^2 + \delta\sqrt{\sum_{i=1}^{m}\alpha_i^2\left(\frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))v_{b,i} - \mathbb{E}[\phi(x_t + \Delta\boldsymbol{f}(\delta u))v_i]\right)^2}.
\end{aligned}$$

In $(i.)$, $\|\xi_{\delta u_b}\|_2 \le {}^1\!/_2\beta_{\boldsymbol{f}}\delta^2$ from Taylor expansion as in Eq. (12). In $(ii.)$, $\boldsymbol{U}_{:,i}$'s are orthogonal basis vectors.

For each i, since the $\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))v_{b,i}$'s for different b's are independent, and within range $[-1, 1]$, we apply Hoeffding's inequality and yield

$$\Pr\left[\left|\frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))v_{b,i} - \mathbb{E}[\phi(x_t + \Delta\boldsymbol{f}(\delta u))v_i]\right| \le \sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)}\right] \ge 1 - \frac{\epsilon}{m}.$$

From union bound, with probability $1-\epsilon$, for any $i \in [m]$, we have

$$\left|\frac{1}{B}\sum_{b=1}^{B}\phi(x_t + \Delta\boldsymbol{f}(\delta u_b))v_{b,i} - \mathbb{E}[\phi(x_t + \Delta\boldsymbol{f}(\delta u))v_i]\right| \le \sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)}.$$

Under this condition, we have

$$\left\|\widetilde{\nabla S}(x_t) - \delta\mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u\right]\right\|_2 \le {}^1\!/_2\beta_{\boldsymbol{f}}\delta^2 + \delta\sqrt{\sum_{i=1}^{m}\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)}. \tag{29}$$

Note that a tighter concentration may be achieved by replacing the Hoeffding's inequality by other tailored tail bounds for Beta-distributed v_i's. But due to the uncertainty brought by the sign term $\phi(\cdot)$, it is challenging. On the other hand, for these i.i.d. random variable's concentration, the Hoeffding's inequality is tight in terms of orders due to central limit theorem.

Now, we combine Eq. (29) with Eq. (19) and get

$$\|\widetilde{\nabla S}(x_t) - \delta\alpha_1\mathbb{E}\left[|v_1|\right]\widehat{\text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)}\|_2 \le p\delta\sqrt{\frac{4w^2}{\|\text{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2^2} + \frac{\sum_{i=2}^{m}\alpha_i^2}{m-1}} + \delta\sqrt{\sum_{i=1}^{m}\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)} + {}^1\!/_2\beta_{\boldsymbol{f}}\delta^2, \tag{30}$$

where $\widehat{\nabla S}$ is the normalized tue gradient, $p := \Pr[E^o]$, and E^o is as defined in Eq. (15). Similar as Eq. (21):

$$\cos\langle \widetilde{\nabla S}(x_t), \mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\rangle \geq 1 - \frac{1}{2}\left(\frac{p\delta\sqrt{\frac{4w^2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2^2} + \frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \delta\sqrt{\sum_{i=1}^m \alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)} + \frac{1}{2}\beta_{\boldsymbol{f}}\delta^2}{\delta\alpha_1\mathbb{E}\left[|v_1|\right]}\right)^2.$$

Following the similar process as in the proof of Theorem 3, we get

$$\begin{aligned}&\cos\langle \widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \\ \geq &\frac{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2}\cdot\left(1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}} + \frac{1.58\beta_{\boldsymbol{f}} + \frac{1}{\delta}\sqrt{\sum_{i=1}^m\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln(\frac{m}{\epsilon})}}{\sqrt{m-1}}\right)^2\right).\end{aligned} \tag{31}$$

□

A.6. Main Result: Expectation Bound Near Boundary (Theorem 1)

Theorem 1 (Expected cosine similarity; approaching boundary)**.** *The difference function S and the projection $\boldsymbol{f}$ are as defined before. For a point x_t that is θ-close to the boundary, i.e., there exists $\theta' \in [-\theta, \theta]$ such that $S(x_t + \theta'\nabla S(x_t)/\|\nabla S(x_t)\|_2) = 0$, let estimated gradient $\widetilde{\nabla S}(x_t)$ be as computed by Definition 2 with step size δ and sampling size B. Over the randomness of the sampled vectors $\{u_b\}_{i=1}^B$,*

$$\begin{aligned}\cos\langle\mathbb{E}\widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \geq &\frac{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2}\cdot \\ &\left(1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}} + \frac{1.58\beta_{\boldsymbol{f}}}{\sqrt{m-1}} + \frac{\gamma\theta}{\alpha_1\delta}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2}\right)^2\right),\end{aligned} \tag{32}$$

where

$$\gamma := \beta_{\boldsymbol{f}} + \frac{\beta_S\left(\max_{i\in[m]}\alpha_i + {}^1\!/_2\delta\beta_{\boldsymbol{f}}\right)^2 + \beta_S\theta^2/\delta^2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2}. \tag{33}$$

Proof of Theorem 1. The high-level idea is similar to the proof of Theorem 3: we build the connection between $\langle\nabla S(x_t, \nabla \boldsymbol{f}(0)\cdot u\rangle$ and $\phi(x_t + \Delta\boldsymbol{f}(\delta u))$. Then, due to the unbiased sampling, the expectation of estimated gradient is close to the true gradient direction with bounded error.

For simplicity, from the symmetry and monotonicity, we let $\theta' = \theta$, i.e., $S(x_t + \theta\nabla S(x_t)/\|\nabla S(x_t)\|_2) = 0$.

Lemma A.4. *We let*

$$w := \frac{1}{2}\delta\left(\beta_{\boldsymbol{f}}\|\nabla S(x_t)\|_2 + \beta_S\left(\|\nabla\boldsymbol{f}(0)\|_2 + \frac{1}{2}\delta\beta_{\boldsymbol{f}}\right)^2\right) + \frac{\beta_S\theta^2}{2\delta}. \tag{34}$$

On the point x_t such that $S(x_t + \theta\nabla S(x_t)/\|\nabla S(x_t)\|_2) = 0$, for any $\delta > 0$ and unit vector $u \in \mathbb{R}^n$,

$$\begin{aligned}\langle\nabla S(x_t),\, \nabla\boldsymbol{f}(0)\cdot u\rangle > w + \frac{\theta\|\nabla S(x_t)\|_2}{\delta} &\Longrightarrow \phi(x_t + \Delta\boldsymbol{f}(\delta u)) = 1, \\ \langle\nabla S(x_t),\, \nabla\boldsymbol{f}(0)\cdot u\rangle < -w + \frac{\theta\|\nabla S(x_t)\|_2}{\delta} &\Longrightarrow \phi(x_t + \Delta\boldsymbol{f}(\delta u)) = -1.\end{aligned}$$

Proof of Lemma A.4. We do Taylor expansion on $S(x_t + \Delta\boldsymbol{f}(\delta u))$ at point x_t:

$$S(x_t + \Delta\boldsymbol{f}(\delta u)) \in S(x_t) + \delta\left(\langle\nabla S(x_t), \nabla\boldsymbol{f}(0)\cdot u\rangle \pm {}^1\!/_2\delta\left(\|\nabla S(x_t)\|_2\beta_{\boldsymbol{f}} + \beta_S\left(\|\nabla\boldsymbol{f}(0)\|_2 + {}^1\!/_2\delta\beta_{\boldsymbol{f}}\right)^2\right)\right).$$

Notice that $S(x_t + \theta\nabla S(x_t)/\|\nabla S(x_t)\|_2)$ can also be expanded at point x_t:

$$0 = S(x_t + \theta\nabla S(x_t)/\|\nabla S(x_t)\|_2) \in S(x_t) + \theta\|\nabla S(x_t)\|_2 \pm \frac{1}{2}\beta_S\theta^2,$$

i.e.,

$$S(x_t) \in -\theta\|\nabla S(x_t)\|_2 \pm \frac{1}{2}\beta_S\theta^2.$$

Therefore,

$$S(x_t+\Delta \boldsymbol{f}(\delta u)) \in \delta\left(\langle\nabla S(x_t), \nabla\boldsymbol{f}(0)\cdot u\rangle - \frac{\theta\|\nabla S(x_t)\|_2}{\delta} \pm \frac{1}{2}\delta\left(\|\nabla S(x_t)\|_2\beta_{\boldsymbol{f}} + \beta_S\left(\|\nabla \boldsymbol{f}(0)\|_2 + \frac{1}{2}\delta\beta_{\boldsymbol{f}}\right)^2\right) \pm \frac{\beta_S\theta^2}{2\delta}\right).$$

Noticing that $\phi(x_t+\Delta\boldsymbol{f}(\delta u)) = \operatorname{sgn}\left(S(x_t+\Delta \boldsymbol{f}(\delta u))\right)$, we conclude the proof. □

According to Lemma 4.1, we write $\Delta\boldsymbol{f}(0) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$. We let $\widehat{\nabla S}(x_t)$ denote the normalized true gradient: $\widehat{\nabla S}(x_t) := \nabla S(x_t)/\|\nabla S(x_t)\|_2$. Furthermore, we define $s := \langle \widehat{\nabla S}(x_t), \boldsymbol{U}_{:,1}\rangle \in \{\pm1\}$, which is the sign between these two aligned vectors.

With respect to the randomness of $u \sim \mathrm{Unif}(S^{m-1})$, we let v denote $\boldsymbol{V}^\intercal u \sim \mathrm{Unif}(S^{m-1})$, and we define the following three events E^-, E^o, and E^+ with probability p^-, p^o and p^+ respectively:

$$\begin{aligned}
E^- &: \langle\nabla S(x_t), \nabla\boldsymbol{f}(0)\cdot u\rangle \in (-\infty, -w + \theta\|\nabla S(x_t)\|_2/\delta), & p^- &:= \Pr[E^-] \quad (35)\\
E^o &: \langle\nabla S(x_t), \nabla\boldsymbol{f}(0)\cdot u\rangle \in [-w + \theta\|\nabla S(x_t)\|_2/\delta, +w + \theta\|\nabla S(x_t)\|_2/\delta], & p^o &:= \Pr[E^o] \quad (36)\\
E^+ &: \langle\nabla S(x_t), \nabla\boldsymbol{f}(0)\cdot u\rangle \in (+w + \theta\|\nabla S(x_t)\|_2/\delta, +\infty). & p^+ &:= \Pr[E^+] \quad (37)
\end{aligned}$$

We notice that Lemma A.3 still holds since u is still uniformly sampled from hypersphere S^{m-1} and Lemma 4.1 still holds for $\nabla\boldsymbol{f}(0)$. Thus,

$$\langle\nabla S(x_t), \nabla\boldsymbol{f}(0)\cdot u\rangle = \alpha_1\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2 s v_1.$$

And with event E^o, we have

$$|v_1| \le \frac{w}{\alpha_1\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1\delta}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}. \tag{38}$$

Now, we can start to bound the error between expectation of estimated gradient and a scaled true gradient. As the first step, according to Eq. (17), we have

$$\|\mathbb{E}\widetilde{\nabla S}(x_t) - \delta\mathbb{E}[\phi(x_t+\Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u]\|_2 \le \frac{1}{2}\beta_{\boldsymbol{f}}\delta^2. \tag{39}$$

Then we decompose $\mathbb{E}[\phi(x_t+\Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u]$ to connect it with the (projected) true gradient:

$$\begin{aligned}
&\mathbb{E}[\phi(x_t+\Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u]\\
=&p^o\mathbb{E}[\phi(x_t+\Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u \,|\, E^o] + p^+\mathbb{E}[\nabla\boldsymbol{f}(0)\cdot u \,|\, E^+] + p^-\mathbb{E}[-\nabla\boldsymbol{f}(0)\cdot u \,|\, E^-]\\
=&p^o\sum_{i=1}^m \alpha_i\mathbb{E}\left[\phi(x_t+\Delta\boldsymbol{f}(\delta u))v_i \,|\, E^o\right]\boldsymbol{U}_{:,i} + p^+\alpha_1\widehat{\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S}(x_t)s\mathbb{E}[v_1 \,|\, E^+] + p^-\alpha_1\widehat{\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S}(x_t)s\mathbb{E}[-v_1 \,|\, E^-]\\
=&\alpha_1 s\left(p^o\mathbb{E}[\phi(x_t+\Delta\boldsymbol{f}(\delta u))v_1 \,|\, E^o] + p^+\mathbb{E}[v_1 \,|\, E^+] + p^-\mathbb{E}[-v_1 \,|\, E^-]\right)\widehat{\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S}(x_t)\\
&\quad + p^o\sum_{i=2}^m \alpha_i\mathbb{E}[\phi(x_t+\Delta\boldsymbol{f}(\delta u))v_i \,|\, E^o]\boldsymbol{U}_{:,i}.
\end{aligned} \tag{40}$$

We notice that

$$\left|s\left(p^o\mathbb{E}[\phi(x_t+\Delta\boldsymbol{f}(\delta u))v_1 \,|\, E^o] + p^+\mathbb{E}[v_1 \,|\, E^+] + p^-\mathbb{E}[-v_1 \,|\, E^-]\right) - \mathbb{E}[|v_1|]\right|$$

$$
\begin{aligned}
&= p^o \left|\mathbb{E}[\phi(x_t + \Delta \boldsymbol{f}(\delta u))v_1 - |v_1| \,|\, E^o]\right| \le 2p^o \mathbb{E}[|v_1| \,|\, E^o] \\
&\le 2p^o \left(\frac{w}{\alpha_1 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1 \delta} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} \right). \qquad \text{(Eq. (38))}
\end{aligned}
$$

For any $i \ge 2$, we define a new vector $v' = (v'_2, v'_3, \cdots, v'_m) \sim \text{Unif}(S^{m-2})$. Thus,

$$\mathbb{E}[\phi(x_t + \Delta \boldsymbol{f}(\delta u))v_i \,|\, E^o] \le \mathbb{E}[|v_i| \,|\, E^o] \le \mathbb{E}[|v'_i|].$$

Combining the above equations to Eq. (40), we get

$$
\begin{aligned}
&\left\|\mathbb{E}[\phi(x_t + \Delta \boldsymbol{f}(\delta u))\nabla \boldsymbol{f}(0) \cdot u] - \alpha_1 \mathbb{E}\left[|v_1|\right] \widehat{\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t)\right\|_2 \\
&\le \sqrt{4\alpha_1^2 (p^o)^2 \left(\frac{w}{\alpha_1 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1 \delta} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} \right)^2 + (p^o)^2 \sum_{i=2}^m \alpha_i^2 \mathbb{E}[|v'_i|]^2} \\
&\le p^o \sqrt{4\alpha_1^2 \left(\frac{w}{\alpha_1 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1 \delta} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} \right)^2 + \sum_{i=2}^m \frac{\alpha_i^2}{m-1}}. \qquad (41)
\end{aligned}
$$

Combining with Eq. (39):

$$
\begin{aligned}
&\left\|\mathbb{E}\widetilde{\nabla S}(x_t) - \delta \alpha_1 \mathbb{E}\left[|v_1|\right] \widehat{\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S}(x_t)\right\|_2 \\
&\le \delta p^o \sqrt{4\alpha_1^2 \left(\frac{w}{\alpha_1 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1 \delta} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} \right)^2 + \sum_{i=2}^m \frac{\alpha_i^2}{m-1}} + \frac{1}{2}\beta_{\boldsymbol{f}} \delta^2. \qquad (42)
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\rangle \\
\ge& 1 - \frac{1}{2}\left(\frac{2p^o w}{\alpha_1 \mathbb{E}\left[|v_1|\right] \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{2p^o \theta}{\alpha_1 \delta \mathbb{E}\left[|v_1|\right]} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{p^o}{\alpha_1 \mathbb{E}\left[|v_1|\right]} \sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \frac{\delta \beta_{\boldsymbol{f}}}{2\alpha_1 \mathbb{E}\left[|v_1|\right]} \right)^2. \qquad (43)
\end{aligned}
$$

We notice that v_1 is distributed around 0 with symmetry and concentration, so

$$
\begin{aligned}
p^o = \Pr[E^o] = \Pr\Bigg[s v_1 \in \Bigg[&-\frac{w}{\alpha_1 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1 \delta} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}, \\
&\frac{w}{\alpha_1 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1 \delta} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} \Bigg]\Bigg] \\
\le \Pr\left[v_1^2 \le \frac{w^2}{\alpha_1^2 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2^2} \right] &\overset{\text{Eq.22}}{\le} \frac{2w}{B(\frac{1}{2}, \frac{m-1}{2}) \alpha_1 \|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}.
\end{aligned}
$$

And from Eq. (23),

$$\mathbb{E}\left[|v_1|\right] = \frac{2}{(m-1) \cdot B(\frac{1}{2}, \frac{m-1}{2})}.$$

Insert them into Eq. (43), we get

$$
\begin{aligned}
\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \ge& \frac{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2} \cdot \\
&\left(1 - \frac{(m-1)^2 \delta^2}{8\alpha_1^2} \left(\frac{\delta \gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1} \sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \frac{\gamma \theta}{\alpha_1 \delta} \cdot \frac{\|\nabla S(x_t)\|_2}{\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2} + \frac{1.58 \beta_{\boldsymbol{f}}}{\sqrt{m-1}} \right)^2 \right), \qquad (44)
\end{aligned}
$$

□

A.7. Main Result: Concentration Bound Near Boundary (Theorem 2)

Theorem 2 (Concentration of cosine similarity; approaching boundary). *Under the same setting as Theorem 1, over the randomness of the sampled vector $\{u_b\}_{i=1}^B$, with probability $1-p$, Under the same setting as Theorem 1, over the randomness of the sampled vector $\{u_b\}_{i=1}^B$, with probability $1-p$,*

$$\cos\langle \mathbb{E}\widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \ge \frac{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2} \cdot$$
$$\left(1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m \alpha_i^2}{m-1}} + \frac{1.58\beta_{\boldsymbol{f}}}{\sqrt{m-1}} + \frac{\gamma\theta}{\alpha_1\delta}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2} + \frac{\frac{1}{\delta}\sqrt{\sum_{i=1}^m \alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln(\frac{m}{p})}}{\sqrt{m-1}}\right)^2\right), \tag{45}$$

where

$$\gamma := \beta_{\boldsymbol{f}} + \frac{\beta_S\left(\max_{i\in[m]}\alpha_i + {}^1\!/\!{}_2\delta\beta_{\boldsymbol{f}}\right)^2 + \beta_S\theta^2/\delta^2}{\|\mathrm{proj}_{\nabla \boldsymbol{f}(0)}\nabla S(x_t)\|_2}. \tag{46}$$

Proof of Theorem 2. The proof follows the similar way as how we extend Theorem 3 to Theorem 4.

Similar as the proof in Theorem 4, by applying Hoefdding bound, with probability $1-\epsilon$,

$$\left\|\widetilde{\nabla S}(x_t) - \delta\mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u\right]\right\|_2 \le {}^1\!/\!{}_2\beta_{\boldsymbol{f}}\delta^2 + \delta\sqrt{\sum_{i=1}^m \alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)}.$$

When this holds, combining it with Eq. (41), we get

$$\begin{aligned}&\left\|\widetilde{\nabla S}(x_t) - \delta\mathbb{E}\left[\phi(x_t + \Delta\boldsymbol{f}(\delta u))\nabla\boldsymbol{f}(0)\cdot u\right]\right\|_2\\ \le&\frac{1}{2}\beta_{\boldsymbol{f}}\delta^2 + \delta\sqrt{\sum_{i=1}^m\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)} + \delta p^o\sqrt{4\alpha_1^2\left(\frac{w}{\alpha_1\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} + \frac{\theta}{\alpha_1\delta}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}\right)^2 + \sum_{i=2}^m\frac{\alpha_i^2}{m-1}},\end{aligned}$$

where p^o is as defined in Eq. (36). Thus,

$$\begin{aligned}&\cos\langle\widetilde{\nabla S}(x_t), \mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\rangle\\ \ge&1 - \frac{1}{2}\left(\frac{2p^o w}{\alpha_1\mathbb{E}\left[|v_1|\right]\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} + \frac{2p^o\theta}{\alpha_1\delta\mathbb{E}\left[|v_1|\right]}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}\right.\\ &\left.+\frac{p^o}{\alpha_1\mathbb{E}\left[|v_1|\right]}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}} + \frac{\delta\beta_{\boldsymbol{f}}}{2\alpha_1\mathbb{E}\left[|v_1|\right]} + \frac{\sqrt{\sum_{i=1}^m\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln\left(\frac{m}{\epsilon}\right)}}{\alpha_1\mathbb{E}\left[|v_1|\right]}\right)^2\\ \ge&1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}} + \frac{1.58\beta_{\boldsymbol{f}}}{\sqrt{m-1}} + \frac{\gamma\theta}{\alpha_1\delta}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} + \frac{\frac{1}{\delta}\sqrt{\sum_{i=1}^m\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln(\frac{m}{\epsilon})}}{\sqrt{m-1}}\right)^2.\end{aligned}$$

We conclude the proof by observing that

$$\cos\langle\widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle = \frac{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2}\cos\langle\widetilde{\nabla S}(x_t), \mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\rangle$$

from Lemma A.1. □

A.8. Bound in Big-$\mathcal{O}$ Notation

We mainly simplify the bound in Theorems 1 and 2 by omitting the terms with smaller orders of m. In boundary attack, following HSJA (Chen et al., 2017), we set binary search precision $\theta = (m\sqrt{m})^{-1}$ and step size $\delta_t = \|x_t - x^*\|_2/m =$

$\Theta(1/m)$. Therefore, $\theta/\delta = \Theta(1/\sqrt{m})$. We first simplify γ in Theorems 1 and 2:

$$\begin{aligned}\gamma &= \beta_{\boldsymbol{f}} + \frac{\beta_S\left(\max_{i\in[m]}\alpha_i + {}^1\!/\!_2\delta\beta_{\boldsymbol{f}}\right)^2 + \beta_S\theta^2/\delta^2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} \\ &= \beta_{\boldsymbol{f}} + \frac{\beta_S\left(\max_{i\in[m]}\alpha_i + \Theta(\beta_{\boldsymbol{f}}/m)\right)^2 + \beta_S\Theta(1/m)}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} = \beta_{\boldsymbol{f}} + \mathcal{O}\left(\frac{\beta_S\max_{i\in[m]}\alpha_i^2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}\right).\end{aligned}$$

We plug in the γ into Theorem 2:

$$\begin{aligned}&\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}} + \frac{1.58\beta_{\boldsymbol{f}}}{\sqrt{m-1}} + \frac{\gamma\theta}{\alpha_1\delta}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} + \frac{\frac{1}{\delta}\sqrt{\sum_{i=1}^m\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln(\frac{m}{p})}}{\sqrt{m-1}} \\ =&\mathcal{O}\left(\frac{\beta_{\boldsymbol{f}}^2}{m\alpha_1}\right) + \mathcal{O}\left(\frac{\beta_S^2\max_{i\in[m]}\alpha_i^4}{m\alpha_1\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2^2}\right) + \mathcal{O}\left(\frac{\beta_{\boldsymbol{f}}}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}}\right) + \mathcal{O}\left(\frac{\beta_S\max_{i\in[m]}\alpha_i^2}{\alpha_1\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}}\right) \\ &+ \mathcal{O}\left(\frac{\beta_{\boldsymbol{f}}}{\sqrt{m}}\right) + \mathcal{O}\left(\frac{\beta_{\boldsymbol{f}}\|\nabla S(x_t)\|_2}{m\alpha_1}\right) + \mathcal{O}\left(\frac{\beta_S\max_{i\in[m]}\alpha_i^2\|\nabla S(x_t)\|_2}{m\alpha_1\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}\right) + \mathcal{O}\left(\frac{1}{\delta}\frac{\alpha_1\sqrt{\frac{2}{B}\ln(\frac{m}{p})}}{\sqrt{m-1}}\right) \\ &+ \mathcal{O}\left(\frac{1}{\delta}\sqrt{\frac{2}{B}\ln\left(\frac{m}{p}\right)}\cdot\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}}\right).\end{aligned}$$

We can discard all terms with negative order of m (since they will be negligible with respect to other terms when m is not too small) and get

$$\begin{aligned}&\frac{\delta\gamma^2}{\alpha_1} + \frac{\gamma}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}} + \frac{1.58\beta_{\boldsymbol{f}}}{\sqrt{m-1}} + \frac{\gamma\theta}{\alpha_1\delta}\cdot\frac{\|\nabla S(x_t)\|_2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2} + \frac{\frac{1}{\delta}\sqrt{\sum_{i=1}^m\alpha_i^2}\cdot\sqrt{\frac{2}{B}\ln(\frac{m}{p})}}{\sqrt{m-1}} \\ =&\mathcal{O}\left(\frac{\beta_{\boldsymbol{f}}}{\alpha_1}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}}\right) + \mathcal{O}\left(\frac{\beta_S\max_{i\in[m]}\alpha_i^2}{\alpha_1\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}}\right) + \mathcal{O}\left(\frac{1}{\delta}\sqrt{\frac{2}{B}\ln\left(\frac{m}{p}\right)}\cdot\sqrt{\frac{\sum_{i=2}^m\alpha_i^2}{m-1}}\right).\end{aligned}$$

Therefore, the bound in Theorem 2 becomes:

$$\begin{aligned}&\cos\langle\mathbb{E}\widetilde{\nabla S}(x_t), \nabla S(x_t)\rangle \\ \geq&\frac{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2}\cdot \\ &\left(1 - \frac{(m-1)^2\delta^2}{8\alpha_1^2}\left(\mathcal{O}\left(\frac{\beta_{\boldsymbol{f}}^2}{\alpha_1^2}\cdot\frac{\sum_{i=2}^m\alpha_i^2}{m-1}\right) + \mathcal{O}\left(\frac{\beta_S^2\max_{i\in[m]}\alpha_i^4}{\alpha_1^2\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2^2}\cdot\frac{\sum_{i=2}^m\alpha_i^2}{m-1}\right) + \mathcal{O}\left(\frac{1}{\delta^2}\cdot\frac{2}{B}\ln\left(\frac{m}{p}\right)\cdot\frac{\sum_{i=2}^m\alpha_i^2}{m-1}\right)\right)\right) \\ =&\frac{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2}\cdot\left(1 - \mathcal{O}\left(m^2\cdot\frac{\sum_{i=2}^m\alpha_i^2}{m-1}\left(\frac{\delta^2\beta_{\boldsymbol{f}}^2}{\alpha_1^4} + \frac{\alpha_{\max}^4}{\alpha_1^4}\cdot\frac{\delta^2\beta_S^2}{\|\mathrm{proj}_{\nabla\boldsymbol{f}(0)}\nabla S(x_t)\|_2^2} + \frac{\ln(\frac{m}{p})}{B\alpha_1}\right)\right)\right).\end{aligned} \tag{47}$$

It recovers the simplified bound in Fig. 3. Note the the last term $\frac{\ln(m/p)}{B\alpha_1}$ is the extra term of Theorem 2 that guarantees the $1-p$ holding probability, and discarding this term yields the version bound for the expectation bound (Theorem 1).

B. Discussion on Key Characteristics and Optimal Scale

This section illustrates the key characteristics for improving the gradient estimator, and discusses the existence of the optimal scale.

B.1. Illustration of Key Characteristics

The figure illustrates the key characteristics, or the optimization goals, for improving the projection-based gradient estimator as discussed in Section 4.1.

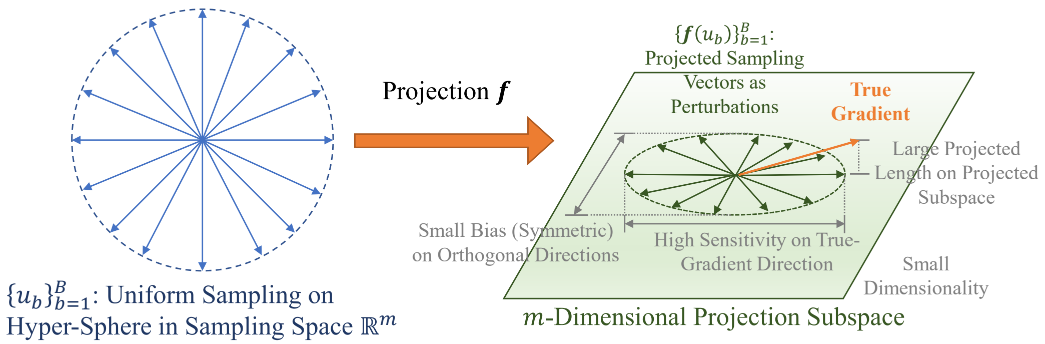
Projection $\boldsymbol{f}$
$\{u_b\}_{b=1}^{B}$: Uniform Sampling on Hyper-Sphere in Sampling Space $\mathbb{R}^m$
$\{\boldsymbol{f}(u_b)\}_{b=1}^{B}$: Projected Sampling Vectors as Perturbations
True Gradient
Large Projected Length on Projected Subspace
Small Bias (Symmetric) on Orthogonal Directions
High Sensitivity on True-Gradient Direction
Small Dimensionality
m-Dimensional Projection Subspace

Figure 9. An illustration of key characteristics for a good projection-based gradient estimator.

B.2. Existence of Optimal Scale

As discussed in Section 4.2, the scale is mapped to the dimensionality of the projection subspace—m.

Due to the trade-off between large $\|\text{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\|_2$ and small m, from Eq. (47), we can intuitively learn that for a given projection $\boldsymbol{f}$, the optimal scale m_{opt} always exists. Now we define this formally. We first explicitly show that $\boldsymbol{f}$ relies on the dimensionality of the projection subspace. To do so, we use $\boldsymbol{f}_m : \mathbb{R}^m \to \mathbb{R}^n$ instead of the general notion $\boldsymbol{f}$. $\boldsymbol{f}_m$ can be viewed as being drawn from a pre-defined projection function family $\mathcal{F} = \{\boldsymbol{f}_i : i \in [n]\}$. Then, the optimal scale m_{opt} can be then explicit expressed as such:

$$m_{\text{opt}} = \arg\max_{m \in [n]} \frac{\|\text{proj}_{\nabla \boldsymbol{f}_m(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2} \cdot \left(1 - \mathcal{O}\left(m^2 \cdot \frac{\sum_{i=2}^{m} \alpha_i^2}{m-1} \left(\frac{\delta^2 \beta_{\boldsymbol{f}_m}^2}{\alpha_1^4} + \frac{\alpha_{\max}^4}{\alpha_1^4} \cdot \frac{\delta^2 \beta_S^2}{\|\text{proj}_{\nabla \boldsymbol{f}_m(0)} \nabla S(x_t)\|_2^2} + \frac{\ln(\frac{m}{p})}{B\alpha_1}\right)\right)\right).$$

The objective function in above $\arg\max$ encodes the precise bound in Theorem 2.

A Simplified Form for Linear Case

When both the projection function $\boldsymbol{f}_m$ and the difference function S are locally linear, i.e., $\beta_{\boldsymbol{f}_m} = \beta_S = 0$, we can simplify the above equation as such:

$$m_{\text{opt}} = \arg\max_{m \in [n]} \frac{\|\text{proj}_{\nabla \boldsymbol{f}_m(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2} \left(1 - Cm^2 \ln\left(\frac{m}{p}\right)\right),$$

where $0 < C < 1$ is a constant.

Now, the existence of optimal scale becomes more apparent. While increasing m can increase $\frac{\|\text{proj}_{\nabla \boldsymbol{f}_m(0)} \nabla S(x_t)\|_2}{\|\nabla S(x_t)\|_2}$, this term has its upper bound 1. On the other hand, the $m^2 \ln(m/p)$ in the second term will also be increased, and it is unbounded. Therefore, an optimal m should be non-zero but not large, i.e., an optimal scale m_{opt} usually exists.

The optimal scale depends on the actual function family $\boldsymbol{f}_m$ and the difference function S. For common and practical cases, as shown in Fig. 4, the objective function for $\arg\max$ is usually unimodal so that the progressive scaling is guaranteed to find the optimal scale. We leave it as our future work to theoretically analyze on what cases this objective function is strictly unimodal.

C. Target Models

In this section, we introduce the target models used in the experiments including the implementation details and the target model performance.

C.1. Implementation Details

Offline Models. Following (Li et al., 2020a; 2021), the pretrained ResNet-18 models are used here as the target models. We also evaluate model ResNeXt50_32×4d (Xie et al., 2017) to demonstrate the generalization ability. For models that are finetuned, cross entropy error is employed as the loss function and is implemented as ‘`torch.nn.CrossEntropyLoss`’ in PyTorch.

For ImageNet, no finetuning is performed as the pretrained target model is just trained exactly on ImageNet. The model is loaded with PyTorch command ‘`torchvision.models.resnet18(pretrained=True)`’ or ‘`torchvision.models.resnext50_32x4d(pretrained=True)`’ following the documentation (PyTorch, 2020).

For CelebA, the target model is fine-tuned to do binary classification on image attributes. Among the 40 binary attributes associated with each image, the top-5 balanced attributes are ‘Attractive’, ‘Mouth_Slightly_Open’, ‘Smiling’, ‘Wearing_Lipstick’, ‘High_Cheekbones’. Though the ‘Attractive’ attribute is the most balanced one, however, it is more subjective than objective, thus we instead choose the second attribute ‘Mouth_Slightly_Open’.

For MNIST and CIFAR-10 datasets, we first resize the original images to 224×224 by linear interpolation, then the target model is finetuned to do 10-way classification. One reason for doing interpolation is that it can provide us more spatial scales to explore. The another reason is that the linear interpolation step also makes image sizes consistent among all the tasks and experiments.

Commercial Online API. Among all the APIs provided by the Face++ platform (MEGVII, 2020c), the ‘Compare’ API (MEGVII, 2020a) which takes two images as input and returns a confidence score indicating whether they contain the faces from the same person. This is also consistent with the same online attacking in (Li et al., 2020a; 2021). In implementation during the attack process, the two image arrays with floating number values are first converted to integers and stored as jpg images on disk. Then they are encoded as base64 binary data and sent as POST request to the request URL (MEGVII, 2020b). We set the similarity threshold as 50% in the experiments following (Li et al., 2020a; 2021): when the confidence score is equal to or larger than 50%, we consider the two faces to belong to the ‘same person’, vice versa.

For source-target images that are from two different persons, the goal of the attack is to get an adversarial image that looks like the target image (has low MSE between the adversarial image and target image), but is predicted as ‘same person’ with the source image. We randomly sample 50 source-target image pairs from the CelebA dataset that are predicted as different persons by the ‘Compare’ API. Then we apply the PSBA pipeline with various perturbation vector generators for comparison.

C.2. Performance of Target Models

The benign accuracy of the target models finetuned on different datasets is shown in Table 4.

Table 4. The benign model accuracy of the target models.

Model	MNIST	CelebA	CIFAR-10	ImageNet
ResNet-18	99.55%	93.77%	88.15%	69.76%
ResNeXt50_32×4d	99.33%	94.00%	90.26%	77.62%

D. Details on PSBA-PGAN

In this section, we introduce the details of Progressive-Scale based projection models including the architecture of Progressive GAN, the training procedure, the algorithm description for progressive scaling and gradient estimation, and the implementation details.

D.1. The Architecture of Progressive GAN

Progressive GAN is a method developed by Karras et. al. (Karras et al., 2018) allowing gradually generating the image from low resolution images to high resolution images. Here, we adopt the implementation of PGAN from pytorch_GAN_zoo (Research, 2020) to help us explore the influence of different scales on attacking performance.
The Conv2d(n_kernel, n_stride, n_pad) here applies He’s constant (He et al., 2015b) at runtime, and for simplicity, the LeakyReLU(negative_slope = 0.2) is denoted as LReLU. Besides, for the generator, we actually utilize bilinear interpolation to implement ‘Unsample’ and utilize average pool to implement ‘Downsample’. Then, the detailed model network structures

for the generator and discriminator with the maximum scale 224×224 are listed in Table 5 and Table 6.

Table 5. The detailed model structure for generator in PGAN.

Generator	Act	Output shape
Latent vector	-	$9408 \times 1 \times 1$
Fully-connected	LReLU	$8192 \times 1 \times 1$
Resize	-	$512 \times 4 \times 4$
Conv(3, 1, 1)	LReLU	$512 \times 4 \times 4$
Upsample	-	$512 \times 7 \times 7$
Conv(3, 1, 1)	LReLU	$256 \times 7 \times 7$
Conv(3, 1, 1)	LReLU	$256 \times 7 \times 7$
Upsample	-	$256 \times 14 \times 14$
Conv(3, 1, 1)	LReLU	$256 \times 14 \times 14$
Conv(3, 1, 1)	LReLU	$256 \times 14 \times 14$
Upsample	-	$256 \times 28 \times 28$
Conv(3, 1, 1)	LReLU	$128 \times 28 \times 28$
Conv(3, 1, 1)	LReLU	$128 \times 28 \times 28$
Upsample	-	$128 \times 56 \times 56$
Conv(3, 1, 1)	LReLU	$64 \times 56 \times 56$
Conv(3, 1, 1)	LReLU	$64 \times 56 \times 56$
Upsample	-	$64 \times 56 \times 56$
Conv(3, 1, 1)	LReLU	$32 \times 112 \times 112$
Conv(3, 1, 1)	LReLU	$32 \times 112 \times 112$
Upsample	-	$32 \times 224 \times 224$
Conv(3, 1, 1)	LReLU	$16 \times 224 \times 224$
Conv(3, 1, 1)	LReLU	$16 \times 224 \times 224$
Conv(1, 1, 0)	Tanh	$3 \times 224 \times 224$

D.2. Projection Model Training Procedure

First, we need to prepare the datasets for PGAN training, which comprise the gradient images generated from a set of *reference models*. Generally, the reference models are assumed to have *different structures* compared with the blackbox target model. Nonetheless, attacker-trained reference models can generate accessible gradients and provide valuable information on the distribution of the target model gradients.

In our case, with the same setting as in (Li et al., 2021), there are five reference models (i.e., DenseNet-121 (Huang et al., 2018), ResNet-50 (He et al., 2015a), VGG16 (Simonyan & Zisserman, 2015), GoogleNet (Szegedy et al., 2014) and WideResNet (Zagoruyko & Komodakis, 2017)) with different backbones compared with the target model, while the implementation and training details are similar with the target model in Section C.1. The benign test accuracy results of these five reference models for MNIST, CIFAR-10 and CelebA datasets are shown in Table 7. After the reference models are trained, their gradients with respect to the training data points are generated with PyTorch automatic differentiation function with command 'loss.backward()'. The loss is the cross entropy between the prediction scores and the ground truth labels.

For ImageNet and CelebA, we randomly sample $500,000$ gradient images ($100,000$ per reference model) for each of ImageNet and CelebA and fix them throughout the experiments for fair comparison.

For CIFAR-10 and MNIST, there are fewer images and so we use the whole dataset and generate $250,000$ gradient images for CIFAR-10 ($50,000$ per reference model) and $300,000$ ($60,000$ per reference model) gradient images for MNIST.

For AE and VAE, they are directly trained on the original gradient datasets, and the training details can be found in (Li et al., 2020a). However, for PGAN, since the size of the images from original gradient datasets is 224×224, we down-scale them by average pool first and then as the new training datasets when the PGAN is trained to build the low resolution image. Actually, this training procedure is just the same as in (Karras et al., 2018), the only difference here is the so-called images are gradient images generated from reference models instead of the real-world pictures.

Table 6. The detailed model structure for discriminator in PGAN.

Discriminator	Act	Output shape
Input image	-	$3 \times 224 \times 224$
Conv(1, 1, 0)	-	$16 \times 224 \times 224$
Conv(3, 1, 1)	LReLU	$16 \times 224 \times 224$
Conv(3, 1, 1)	LReLU	$32 \times 224 \times 224$
Downsample	-	$32 \times 112 \times 112$
Conv(3, 1, 1)	LReLU	$32 \times 112 \times 112$
Conv(3, 1, 1)	LReLU	$64 \times 112 \times 112$
Downsample	-	$64 \times 56 \times 56$
Conv(3, 1, 1)	LReLU	$64 \times 56 \times 56$
Conv(3, 1, 1)	LReLU	$128 \times 56 \times 56$
Downsample	-	$128 \times 28 \times 28$
Conv(3, 1, 1)	LReLU	$128 \times 28 \times 28$
Conv(3, 1, 1)	LReLU	$256 \times 28 \times 28$
Downsample	-	$256 \times 14 \times 14$
Conv(3, 1, 1)	LReLU	$256 \times 7 \times 7$
Conv(3, 1, 1)	LReLU	$512 \times 7 \times 7$
Downsample	-	$512 \times 4 \times 4$
Minibatch stddev	-	$513 \times 4 \times 4$
Conv(3, 1, 1)	LReLU	$512 \times 4 \times 4$
Fully-connected	LReLU	$512 \times 1 \times 1$
Fully-connected	Linear	$1 \times 1 \times 1$

D.3. Reference Model Performance

Intuitively, with well-trained reference models that perform comparatively with the target models, the attacker can get gradient images that are in a more similar distribution with the target model's gradients for training, thus increasing the chance of an attack with higher quality. The reference model performance in terms of prediction accuracy for MNIST, CIFAR-10, CelebA and ImageNet datasets are shown in Table 7. The model performance is comparable to that of the target models.

D.4. Algorithm Description

We provide the pseudocode for the progressive-scale process with the PSBA-PGAN in Algorithm 1 (once the optimal scale determined, it will be used across all the pairs of source and target images) and for frequency reduction gradient estimation with the PGAN224 in Algorithm 2.

D.5. Attack Implementation

The goal is to generate an adversarial image that looks similar as the target image, i.e., as close as to the target image, but is predicted as the label of the target image. We fix the random seed to 0 so that the samples are consistent across different runs and various methods to ensure reproducibility and to facilitate fair comparison.

Table 7. The benign model accuracy of the reference models on four datasets. For the dataset MNIST and CIFAR10, the images are linearly interpolated to size 224×224; for the dataset CelebA, the attribute is chosen as 'mouth_slightly_open'

Dataset	DenseNet-121	ResNet-50	VGG16	GoogleNet	WideResNet
MNIST	98.99%	99.43%	99.16%	99.46%	98.59%
CIFAR10	92.73%	88.47%	92.67%	92.26%	85.19%
CelebA	93.81%	94.02%	94.13%	91.77%	93.79%
ImageNet	74.65%	76.15%	71.59%	69.78%	78.51%

Algorithm 1 The Process for Searching the Optimal Scale for PSBA-PGAN.

```
**Input:** a validation set which comprises ten pairs of source-target images, the PGANs with different output scales, access to query the decision of target model.
**Output:** the optimal scale for attacking the target model.
$optimal\_scale \leftarrow 7 \times 7$
$lowest\_distance \leftarrow \infty$
**for** $s = 7 \times 7$ to $224 \times 224$ **do**
    Take the PGAN generator with output scale $s$ as the gradient estimator to attack the target model.
    $current\_distance \leftarrow$ the MSE of the the ten adversarial images to the corresponding target images after 10 step attack. The number of sampled perturbation vectors per step is set to $100$.
    **if** $current\_distance \leq lowest\_distance$ **then**
        $lowest\_distance \leftarrow current\_distance$
        $optimal\_scale \leftarrow s$
    **else**
        **return** $optimal\_scale$
    **end if**
**end for**
**return** $optimal\_scale$
```

Algorithm 2 Frequency Reduction Gradient Estimation

```
**Input:** a data point on the decision boundary $x \in \mathbb{R}^m$, nonlinear projection function $\boldsymbol{f}$, number of random sampling $B$, access to query the decision of target model $\phi(\cdot) = \text{sgn}(S(\cdot))$.
**Output:** the approximated gradient $\widetilde{\nabla S}(x_{adv}^{(t)})$.
Sample $B$ random Gaussian vectors of the lower dimension: $v_b \in \mathbb{R}^n$.
Use PGAN224 to project the random vectors to the gradient space: $u_b = \boldsymbol{f}(v_b) \in \mathbb{R}^m$.
Do DCT transformation on each channel of $u_b$ and get the frequency representation: $d_b = \text{DCT}(u_b)$
Save the $k \times k$ signals on the upper left corner and set other signals to zero : $d'_b = \text{Filter}(d_b)$
Map the signals back to the original space by Inverse DCT transformation: $u'_b = \text{IDCT}(d'_b)$
Get query points by adding perturbation vectors with the original point on the decision boundary $x_{adv}^{(t)} + \delta u'_b$.
Monte Carlo approximation for the gradient:
$\widetilde{\nabla S}(x_{adv}^{(t)}) = \frac{1}{B}\sum_{b=1}^{B} \phi\left(x_{adv}^{(t)} + \delta u'_b\right) u'_b = \frac{1}{B}\sum_{b=1}^{B} \text{sgn}\left(S\left(x_{adv}^{(t)} + \delta u'_b\right)\right) u'_b$
**return** $\widetilde{\nabla S}(x_{adv}^{(t)})$
```

Gradient Estimation. For convenience and precision concern, we just use $\delta_t \boldsymbol{f}(u_b)$ instead of the theoretical representation $\Delta \boldsymbol{f}(\delta_t u_b)$, i.e., $\boldsymbol{f}(\delta_t u_b) - \boldsymbol{f}(0)$ in the actual calculation of estimated gradient $\widehat{\nabla S}(x_t)$ and the $\boldsymbol{f}(u_b)$ is normalized here. Besides, the variance reduction balancing adopted in (Chen et al., 2020) is also applied in our gradient estimation out of the concern for the accuracy of estimation.

Offline Models. During the attack, we randomly sample source-target pairs of images from each of the corresponding datasets. We query the offline models with the sampled images to make sure both source image and target image are predicted as their ground truth labels and the labels are different so that the attack is nontrivial. For the same dataset, the results of different attack methods are reported as the average of the same 50 randomly sampled pairs.

Online API. For the online API attacks, the source-target pairs are sampled from the dataset CelebA. The results of different attack methods are also reported as the average of the same 50 randomly sampled pairs.

E. Quantitative Results

E.1. Attack Setup

We randomly select 50 pairs of source and target images from test set that are predicted by the target model as different classes for both offline attack and online attack. The goal here is to move the source image gradually to the target image under the measure of MSE while maintaining being predicted as source label by the target model. In this process, the number of sampled perturbation vectors at each step (B in Definition 2) is controlled as 100 for every gradient generator in the Monte Carlo algorithm to estimate the gradient for fairness (except EA, in which the B is set to 1) . The optimal dimensions chosen on the search space for EA are shown in Table 8, and other hyper-parameters are the same with the setting in (Dong et al., 2019).

Table 8. The optimal dimension of the search space for EA on different datasets and target models.

Model	MNIST	CIFAR-10	CelebA	ImageNet
ResNet-18	$30 \times 30 \times 1$	$30 \times 30 \times 3$	$112 \times 112 \times 3$	$30 \times 30 \times 3$
ResNeXt50_32×4d	$30 \times 30 \times 1$	$45 \times 45 \times 3$	$45 \times 45 \times 3$	$45 \times 45 \times 3$

E.2. Time and Resource Consumption

The optimal scale is usually small and relatively stable for an assigned dataset as shown in Appendix E.9. Indeed, from our experimental observation, it is enough to just use 10 scr-tgt image pairs to determine the optimal scale within 10 minutes on one 2080 Ti GPU. Besides, the model is trained before we start to attack, and the optimal scale is also determined before the attack. So in fact, no matter in the training or attacking stage, the time and resource consumption are almost the same with other generative model-based attacks (Li et al., 2021; Tu et al., 2020). The PGAN training time on scale 28×28 for ImageNet is about one day with two RTX 2080 Ti GPUs and that for the scale 224×224 is about two days. We note that the PGAN training can be done offline and is one-time training for attacking different models. Besides, the averaged attack time with $10,000$ queries on ImageNet of HSJA is 114.2s, and that of PSBA is 136.4s. We remark that the bulk of PSBA attack time is on the resize operation similar to the baseline QEBA-S (see (Li et al., 2020a)).

E.3. Attack Performance for Different Datasets and Target Models

The complete attack performance results for different datasets and target models are shown in Fig. 10. On complex datasets like ImageNet, the major challenge is the excessive number of categories (1,000 on ImageNet). Thus, we suspect that the geometry of model's decision boundary is more non-smooth and complex, and therefore the general boundary attacks should be harder to improve.

The 'successful attack' is defined as the x_{adv} reaching some pre-defined distance threshold under the metric of MSE. Note that because the complexity of tasks and images varies between datasets, we set different MSE thresholds for the datasets. For example, ImageNet images are the most complicated so the task is most difficult, thus we set larger (looser) threshold for it. The corresponding numerical results for small query number constrains are shown in Table 9 and Table 10, the visualized attack success rates for different target models are shown in Fig. 11. Since in practice, we care about the efficiency more,

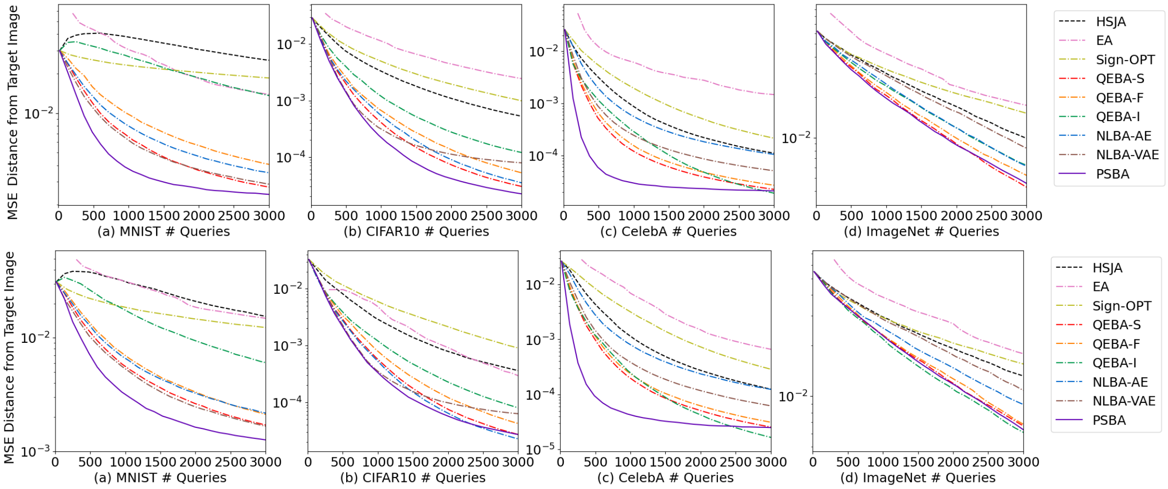

Figure 10. Perturbation magnitude (MSE) w.r.t. query numbers. Row 1: For attacks on ResNet-18; Row 2: For attacks on ResNeXt50_32×4d.

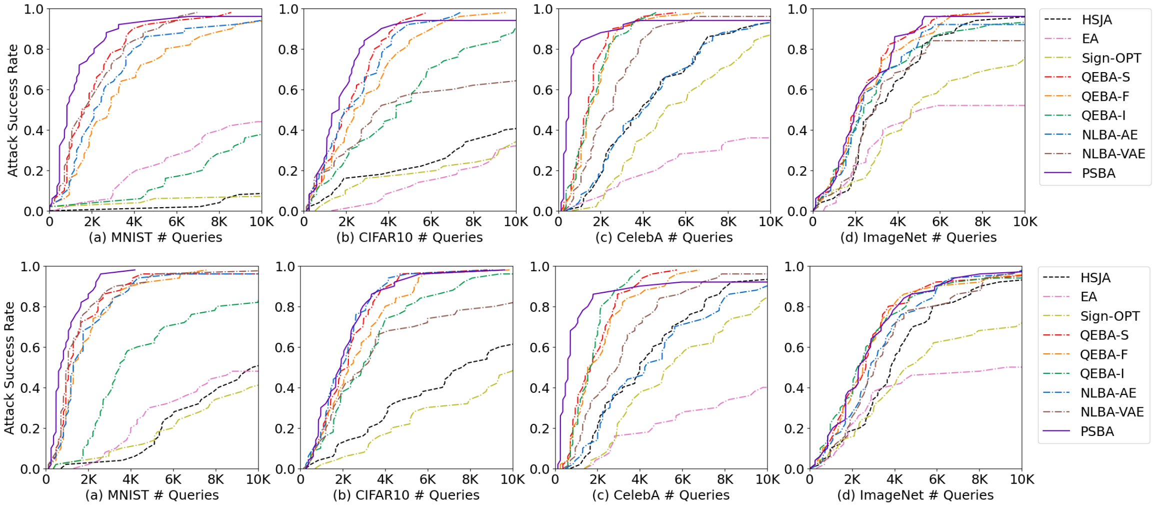

Figure 11. The attack success rate w.r.t. number of queries on four different datasets. Row 1: For attacks on ResNet-18; Row 2: For attacks on ResNeXt50_32×4d.

Table 9. Comparison of the attack success rate for different attacks at query number 1K (the perturbation magnitude under MSE for each dataset are: MNIST: $5e-3$; CIFAR10: $5e-4$; CelebA: $1e-4$; ImageNet: $1e-2$).

Data	Model	# Queries = 1K								
		HSJA	EA	Sign-OPT	QEBA-S	QEBA-F	QEBA-I	NLBA-AE	NLBA-VAE	PSBA
MNIST	ResNet	2%	4%	4%	40%	18%	4%	26%	44%	**64%**
	ResNeXt	4%	0%	4%	44%	30%	4%	28%	50%	**64%**
CIFAR10	ResNet	16%	2%	14%	52%	42%	30%	50%	50%	**62%**
	ResNeXt	12%	0%	8%	44%	38%	30%	48%	**52%**	48%
CelebA	ResNet	8%	6%	2%	48%	36%	40%	10%	22%	**86%**
	ResNeXt	2%	0%	0%	32%	24%	22%	6%	12%	**72%**
ImageNet	ResNet	10%	8%	12%	**22%**	18%	**22%**	18%	10%	20%
	ResNeXt	12%	6%	10%	16%	14%	**24%**	14%	10%	20%

that is, we usually focus on the attack performance when the query number is small, like 1K or 2K. But we still provide the results for large query number constraints in Table 11, Table 12 and Table 13, showing that PSBA achieves the highest or comparable ASR to other approaches. Note that PSBA converges significantly faster than baselines ($\leq 3K$) (Fig.6 & 10 in paper), which leads to its high attack success rate with small number of queries. On the other hand, when large number of queries are allowed, baselines such as QEBA will eventually converge to similar result (e.g., close to 100% ASR), which

Table 10. Comparison of the attack success rate for different attacks at query number 3K (the perturbation magnitude under MSE for each dataset are: MNIST: $5e-3$; CIFAR10: $5e-4$; CelebA: $1e-4$; ImageNet: $1e-2$).

Data	Model	# Queries = 3K								
		HSJA	EA	Sign-OPT	QEBA-S	QEBA-F	QEBA-I	NLBA-AE	NLBA-VAE	PSBA
MNIST	ResNet	2%	12%	4%	80%	58%	6%	66%	76%	**90%**
	ResNeXt	4%	10%	10%	88%	80%	38%	82%	90%	**98%**
CIFAR10	ResNet	40%	34%	24%	96%	90%	74%	90%	**96%**	**96%**
	ResNeXt	50%	2%	30%	96%	90%	82%	96%	**100%**	96%
CelebA	ResNet	40%	16%	24%	92%	92%	88%	44%	68%	**94%**
	ResNeXt	38%	18%	24%	88%	78%	90%	40%	52%	**90%**
ImageNet	ResNet	54%	36%	30%	**68%**	66%	64%	64%	56%	**68%**
	ResNeXt	36%	38%	28%	**68%**	64%	64%	54%	48%	66%

Table 11. Comparison of the attack success rate for different attacks at query number 5K (the perturbation magnitude under MSE for each dataset are: MNIST: $5e-3$; CIFAR10: $5e-4$; CelebA: $1e-4$; ImageNet: $1e-2$).

Data	Model	# Queries = 5K								
		HSJA	EA	Sign-OPT	QEBA-S	QEBA-F	QEBA-I	NLBA-AE	NLBA-VAE	PSBA
MNIST	ResNet	2%	24%	8%	94%	76%	12%	88%	90%	**96%**
	ResNeXt	12%	32%	14%	98%	94%	62%	96%	96%	**100%**
CIFAR10	ResNet	22%	12%	20%	**96%**	86%	54%	94%	58%	94%
	ResNeXt	36%	2%	22%	**98%**	86%	82%	96%	70%	94%
CelebA	ResNet	66%	24%	50%	98%	98%	**100%**	66%	92%	96%
	ResNeXt	62%	24%	42%	98%	90%	**100%**	54%	86%	92%
ImageNet	ResNet	74%	50%	54%	90%	86%	82%	84%	78%	**92%**
	ResNeXt	72%	48%	54%	**88%**	**88%**	86%	78%	80%	**88%**

Table 12. Comparison of the attack success rate for different attacks at query number 8K (the perturbation magnitude under MSE for each dataset are: MNIST: $5e-3$; CIFAR10: $5e-4$; CelebA: $1e-4$; ImageNet: $1e-2$).

Data	Model	# Queries = 8K								
		HSJA	EA	Sign-OPT	QEBA-S	QEBA-F	QEBA-I	NLBA-AE	NLBA-VAE	PSBA
MNIST	ResNet	4%	40%	8%	98%	90%	30%	92%	**100%**	98%
	ResNeXt	40%	46%	36%	98%	**100%**	80%	98%	98%	**100%**
CIFAR10	ResNet	36%	22%	24%	**100%**	98%	82%	**100%**	64%	96%
	ResNeXt	54%	2%	36%	**98%**	**98%**	94%	**98%**	80%	**98%**
CelebA	ResNet	88%	36%	72%	**100%**	**100%**	**100%**	90%	98%	96%
	ResNeXt	90%	34%	72%	**100%**	**100%**	**100%**	86%	98%	94%
ImageNet	ResNet	96%	54%	70%	**98%**	**98%**	92%	94%	86%	**98%**
	ResNeXt	90%	48%	70%	94%	92%	94%	96%	88%	**96%**

again demonstrates the importance of evaluation under small query budget.

E.4. Comparison with RayS Attack

RayS attack (Chen & Gu, 2020) performs blackbox attack by enumerating gradient signs at different scales, which is an efficient attack strategy for untargeted attack under ℓ_∞ norm. While PSBA can perform both untargeted and targeted attacks, the gradient generator used in our experiments is mainly designed for the targeted attack under MSE measure, which is a more practical and tougher task. To customize our PSBA for ℓ_∞ norm bounded attack scenario, we believe some specific design for the generator is needed. But currently, even if we directly use the original PGAN generator, our method can still compete with the RayS attack when compared under the ℓ_∞ based untargeted attack scenario. The corresponding additional experiments conducted on ImageNet dataset are shown in Table 14, demonstrating that the PSBA always outperforms RayS in targeted attacks, while RayS achieves slightly better results for untargeted attack under ℓ_∞.

Table 13. Comparison of the attack success rate for different attacks at query number 10K (the perturbation magnitude under MSE for each dataset are: MNIST: $5e-3$; CIFAR10: $5e-4$; CelebA: $1e-4$; ImageNet: $1e-2$).

Data	Model	# Queries = 10K								
		HSJA	EA	Sign-OPT	QEBA-S	QEBA-F	QEBA-I	NLBA-AE	NLBA-VAE	PSBA
MNIST	ResNet	10%	46%	8%	**100%**	94%	38%	96%	**100%**	98%
	ResNeXt	52%	50%	42%	98%	**100%**	84%	98%	98%	**100%**
CIFAR10	ResNet	42%	34%	36%	**100%**	**100%**	92%	**100%**	66%	96%
	ResNeXt	62%	2%	50%	**100%**	**100%**	98%	**100%**	82%	**100%**
CelebA	ResNet	94%	38%	88%	**100%**	**100%**	**100%**	94%	98%	96%
	ResNeXt	94%	42%	86%	**100%**	**100%**	**100%**	92%	98%	94%
ImageNet	ResNet	96%	54%	76%	**100%**	**100%**	94%	94%	86%	98%
	ResNeXt	94%	52%	72%	96%	96%	96%	**98%**	98%	**98%**

Table 14. Comparison of Attack Success Rate (ASR) on 100 randomly chosen ImageNet images for ResNet-18 with different perturbation thresholds ϵ and attack types (the query budget = 10000).

Attack Type	ϵ	Methods	Avg. Queries	Med. Queries	ASR (%)
Targeted Attack	0.30 (ℓ_∞)	HSJA	1909.9	853.0	45
		RayS	4677.0	4677.0	2
		PSBA	**1437.7**	**502.0**	**96**
	0.01 (MSE)	HSJA	3182.8	2726.0	**99**
		RayS	2479.0	2479.0	2
		PSBA	**2460.4**	**1924.0**	**99**
Untargeted Attack	0.05 (ℓ_∞)	HSJA	1782.8	595.5	88
		RayS	**528.7**	**214.5**	**100**
		PSBA	596.9	270.0	99
	0.0001 (MSE)	HSJA	2208.4	1305.0	92
		RayS	1798.1	625.0	83
		PSBA	**1151.5**	**486.0**	**96**

Since the RayS in targeted setting is not mentioned in the original paper, we have tried two ways to implement it: 1) initialize the perturbation using the same setting as that in untargeted attack; 2) initialize the perturbation using the images from the target class. The values recorded in the table are the better ones between these two ways and there may still exist a better way to do it. The other thing to note is that currently all the methods are not so powerful on ImageNet for targeted attack with the perturbation threshold ϵ set to 0.05 under ℓ_∞ norm.

E.5. Cosine Similarity Measure for Offline Models

The cosine similarity between the estimated gradient and the true gradient is a significant measure of the quality of the gradient estimation, and is highly correlated with the actual attack performance. The cosine similarity for the different boundary attack methods are shown in Fig. 12. As we can see, our approach PSBA-PGAN usually achieves higher cosine similarity especially when the number of queries is limited.

E.6. Long Tail Distribution for the Gradients Generated from Different Models on Frequency Domain

As mentioned in the Section 5.2, the gradients generated by the target model ResNet-18 tend to focus on the low-frequency region. However, this pattern actually exists on other models as well and the experiments conducted here are in a more statistical sense: first, gradients from $1,000$ images are generated from these six models on different datasets respectively; then, by transforming them into frequency domain by DCT transformation, we average the absolute value of the coefficients on the corresponding basis components and smooth them by the Savitzky-Golay filter. As a result, if we draw these components from low-frequency to high-frequency on x-axis, we will see the interesting long tail distribution as shown in the Fig. 13. This extensively existed phenomenon, as justified in Section 4.2, indicates that the attack performance would be

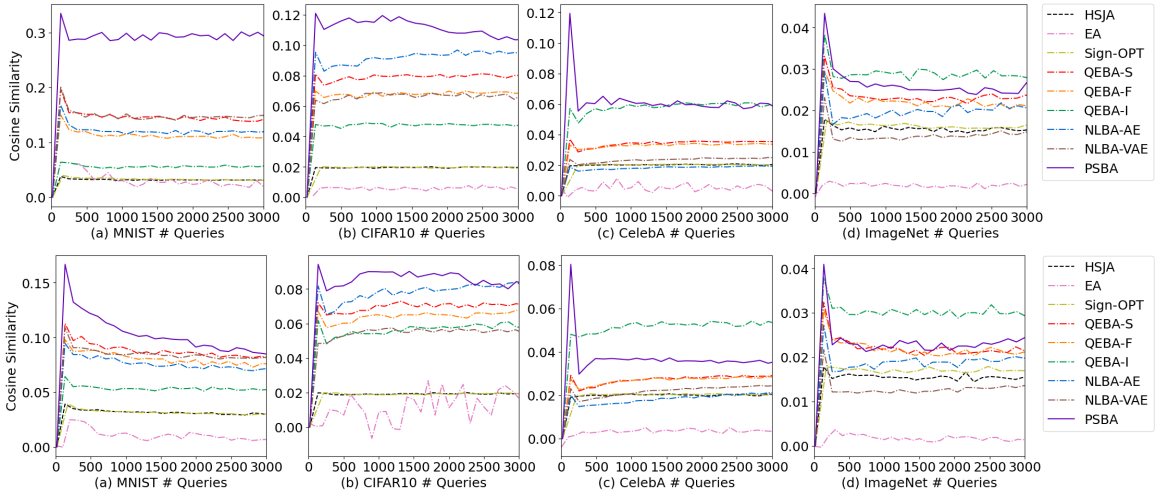

Figure 12. The cosine similarity between the estimated and true gradient w.r.t. the number of queries during attacking models on different datasets. Row 1: For attacks on ResNet-18; Row 2: For attacks on ResNeXt50_32×4d.

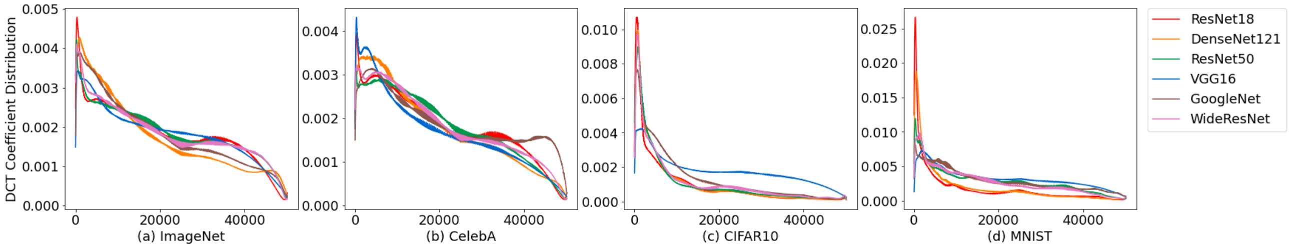

Figure 13. The long-tailed distribution for the coefficients of the gradients represented on DCT basis on different models and datasets.

improved if we just save the low-frequency part of the generated gradient images. This conjecture has already been proved by our experiments in Appendix E.8.

E.7. High Sensitivity on Gradient Direction

Hereinafter, the target model is specified to ResNet-18 for reducing the redundancy. The tendency on ResNeXt models or other models is the same, that is, the results shown below are actually decoupled with the structure of the target model.

Verification of High Sensitivity. We empirically verify that the trained PGAN has higher sensitivity on the projected true gradient as discussed in Section 5.2. The gradient estimator chosen here is the PGAN generator with the optimal scale 28×28 for each dataset. Inspired by the definition in Lemma 4.1, the value α_1^2 is approximately calculated by $\frac{1}{B}\sum_{b=1}^{B}\cos^2\langle \Delta \boldsymbol{f}(\delta_t u_b), \mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\rangle$, where the number of queries, i.e., B, is set to $10,000$ instead of the original 100 for better estimation and verification. The value $\frac{\sum_{i=2}^{m}\alpha_i^2}{m-1}$ is then approximately calculated by $\frac{1}{B(m-1)}\sum_{b=1}^{B}(1 - \cos^2\langle \Delta \boldsymbol{f}(\delta_t u_b), \mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)\rangle)$. Since the actual output scale of $\boldsymbol{f}$ is 28×28 here, the ground gradient $\nabla S(x_t)$ is resized to 28×28 first (thus denoted by $\mathrm{proj}_{\nabla \boldsymbol{f}(0)} \nabla S(x_t)$) and the value m is equal to $n\_channel \times 28 \times 28$ here. All these values are averaged on 50 pairs of source-target images with 10-step attack on ResNet-18.

As shown in Fig. 14, across all four datasets, we observe that the sensitivity on the projected true gradient direction (blue bars) is significantly higher than the (averaged) sensitivity on other orthogonal directions (purple bars).

Adjust Sensitivity on Different Directions. Here, we deliberately adjust the sensitivity on different directions to show the correlation between the attack performance and sensitivity by changing the weight of the components which are orthogonal to the ground gradient. In other words, we replace the $\Delta \boldsymbol{f}(\delta_t u_b)$ in the original calculation of estimated gradient $\widetilde{\nabla S}(x_t)$ with $\left(\frac{\langle \Delta \boldsymbol{f}(\delta_t u_b), \nabla S(x_t)\rangle}{\|\nabla S(x_t)\|}\nabla S(x_t) + k\left(\Delta \boldsymbol{f}(\delta_t u_b) - \frac{\langle \Delta \boldsymbol{f}(\delta_t u_b), \nabla S(x_t)\rangle}{\|\nabla S(x_t)\|}\nabla S(x_t)\right)\right)$ and then repeat the attack on the target model. Lower value of k means less weight is put on the orthogonal components. Empirically, the range of k is set between 0.96 to 1.04 and it is worth noting that when the value of k is set to 1, the new gradient estimation adopted here is just the same with the original gradient estimation. We choose the projection 'PGAN28' and dataset ImageNet. As shown in Fig. 15, aligned with our theoretical analysis, lower k results in better attack performance, and vise versa.

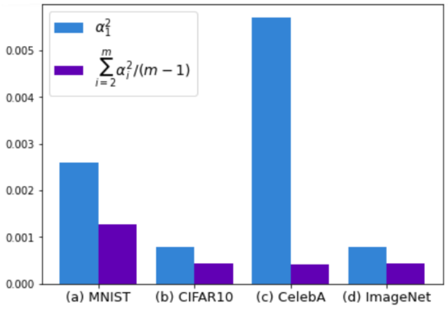

Figure 14. The α value on diverse datasets.

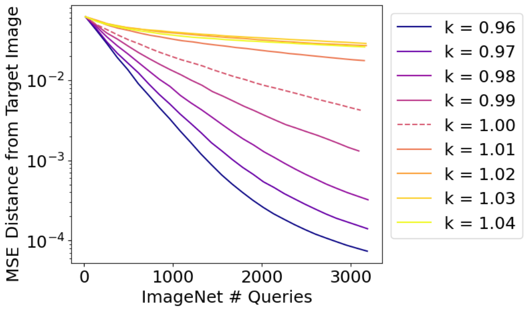

Figure 15. The perturbation magnitude w.r.t. different number of queries for different k values on ImageNet.

E.8. PSBA with Different Domains

In this subsection, we demonstrate the performance of PSBA when applied to other domains like frequency and spectrum.Since the PGAN is originally designed for spatial expansion, it would be also beneficial for us to adopt some similar training strategy to expand progressively on both frequency and spectrum domain. However, for both convenience and effectiveness, we just take PGAN224, whose attack performance is almost the worst compared to other PGAN with smaller scale, as the gradient generator in the experiments here. As a result, the attack performance on these other domains can be further improved by progressively expanding training strategy.

With this in mind, in this subsection, the main attention is concentrated on: 1) the existence of the optimal scale on both frequency and spectrum domain.2) whether the original attack performance of PGAN224 would be improved a lot by just selecting the optimal scale on specific frequency or spectrum domain. Since the conclusions are consistent with any model, we just show the experiment results on ResNet-18 below as an instance.

Frequency Domain. As discussed in Appendix E.6, with applying the DCT transformation on the output of PGAN in 224×224 scale, the low-frequency components will be concentrated at the upper left corner, i.e., low-frequency subspace. Then, we let 'PGAN224dk' denote the adjusted attack process where we just save the $k \times k$ signals on the upper left corner of the frequency representation of the output of PGAN224 and transform it back to the original space by the Inverse DCT to continue the attack. In other words, we just use some low-frequency part of the original gradient images generated from PGAN224 to estimate the ground gradient and the pseudocode is also provided in Algorithm 2 for making it more clear. The final attack performance is shown in Fig. 16, and as we can see, in some cases like when the src-tgt images are sampled from MNIST, PSBA-freq even outperforms all other baselines with a simple adjustment on the output of the inherent bad gradient estimator PGAN224. The results corresponding to different choices of frequency region and their induced changes of cosine similarity are shown in Fig. 17 and Fig. 18. Besides, this simple strategy can also work well on the attack to the online API, which is shown in Fig. 19 and Fig. 20.

As we can see, even this simple "gating" strategy can improve the attack performance a lot compared with original PGAN224. Though it is not as competitive as the progressive scaling in the spatial domain due to lack of projection model finetuning. Furthermore, the existence of optimal scale is pronounced.

Spectrum Domain. Here, we sample 40,000 gradient images generated from PGAN224, and then use PCA to decompose them to get $9,408$ main components. It may seem to be great if we project the original gradient images generated from PGAN224 to just part of the main components, however, the computation cost is a little unacceptable, since there are a lot of dot product operations between two $150,000$-dimensional vectors required. Therefore, for efficiency, the gradient images generated here are actually composed by the combination of the top-k main components among the total $9,408$ components with the coefficients sampled from normal distribution. Thus, for simplicity, we denote 'PGAN224pk' as the attack with the combination of the top-k main components decomposed by PCA. By progressively increase the value of k, the attack performance are shown in Fig. 21. The result on different spectrum scales and corresponding changes of cosine similarity

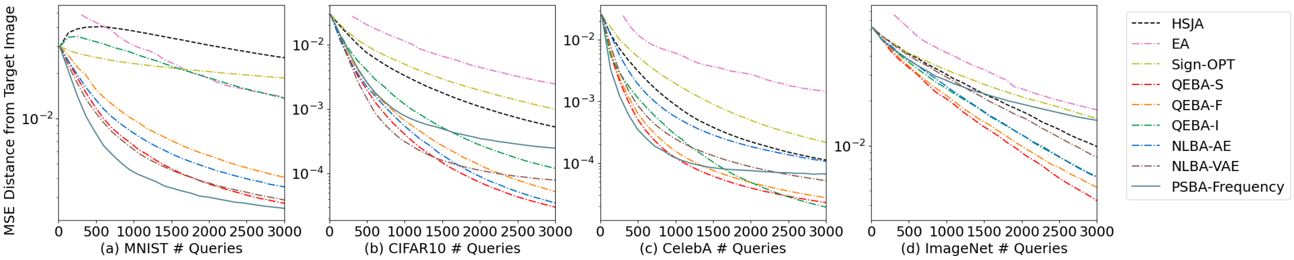

Figure 16. The perturbation magnitude w.r.t. different number of queries for different methods. The PSBA here is applied on frequency domain.

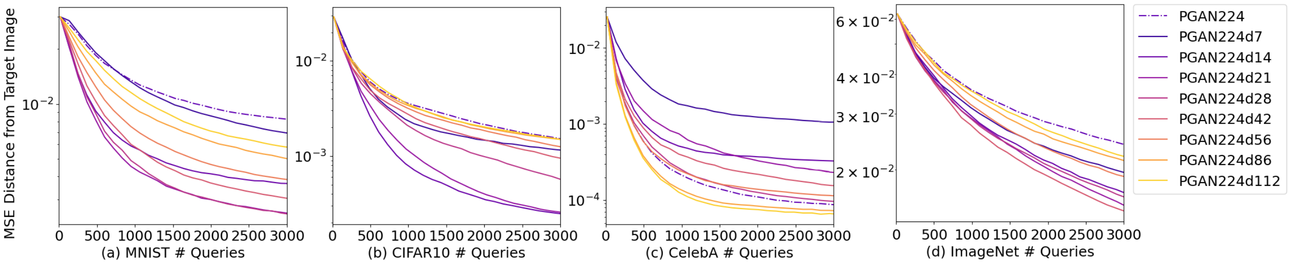

Figure 17. The perturbation magnitude w.r.t. different number of queries for different scales chosen on frequency domain.

are shown in Fig. 22 and Fig. 23. Again, we observe an apparent improvement over the original PGAN224 and the existence of the optimal scale.

E.9. Optimal Scale across Different Model Structures

One may think that the optimal scale may be influenced a lot by the specific structure of the target model. In other words, the depth of the model, the existence of the residual connection and batchnorm layer, and so on, would affect the optimal scale. However, the result as shown in Table 15 demonstrates that the optimal scale is actually stable, which is usually the small scale 28×28.

It is our future work to look into this phenomenon and analyze the optimal properties such as stability of optimal scale. Besides, together with the improved attack performance owing to the removing of the high frequency part or the focus on the most informative spectrum of the generated images of PGAN224, it will be promising to explore the benefits brought from gradient sparsification and devise a more efficient algorithm in the future.

F. Qualitative Results

In this section, we present the qualitative results for attacking both offline models and online APIs.

F.1. Offline Models

The goal of the attack is to generate an adversarial image that looks like the target image but has the same label with source image. We report qualitative results that show how the adversarial image changes during the attack process in Figure 24, Figure 25, Figure 26, and Figure 27 for the four datasets respectively. The target model chosen here is ResNet-18. In the figures, the left-most column has two images: the source image and the target image. They are randomly sampled from the corresponding dataset. We make sure that the images in the sampled pairs have different ground truth labels (otherwise the attack is trivial). The other five columns each represents the adversarial image at certain number of queries as indicated by $\#q$ at the bottom line. In other words, all images in these five columns can successfully attack the target model. Each row represents one method as shown on the right. The d value under each image shows the MSE between the adversarial image and the target image. The smaller the d is, the better the attack is.

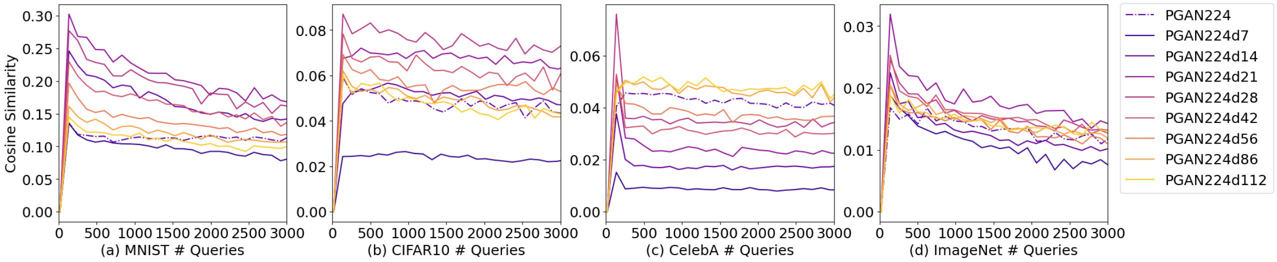

Figure 18. The cosine similarity between the estimated and true gradients for different scales chosen on frequency domain.

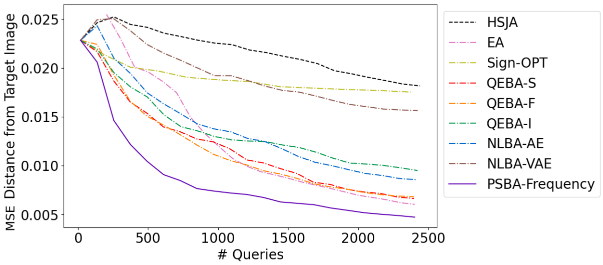

Figure 19. The perturbation magnitude w.r.t. different queries against Face++ ‘Compare’ API, the PSBA here is applied on frequency domain.

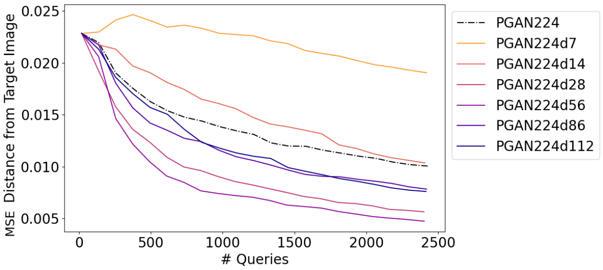

Figure 20. The perturbation magnitude w.r.t. different number of queries for different scales chosen on frequency domain against Face++ ‘Compare’ API.

F.2. Commercial Online API Attack

As discussed in Section 5, the goal is to generate an adversarial image that looks like the target image but is predicted as ‘same person’ with the source image. In this case, we want to get images that looks like the man but is actually identified as the woman. The qualitative results of attacking the online API Face++ ‘compare’ is shown in Figure 28. In the figure, the source image and target image are shown on the left-most column.

Table 15. The optimal scale across different model structures.

Dataset Model	MNIST	CIFAR-10	CelebA	Imagenet
ResNet-18	28×28	28×28	28×28	28×28
ResNet-34	28×28	28×28	28×28	56×56
ResNet-152	28×28	28×28	56×56	56×56
ResNext50_32x4d	28×28	28×28	28×28	56×56
Vgg11	28×28	28×28	28×28	56×56
Vgg19	28×28	28×28	28×28	56×56
Vgg11_bn	112×112	28×28	28×28	56×56
Vgg19_bn	112×112	28×28	28×28	56×56
DenseNet161	28×28	14×14	28×28	56×56

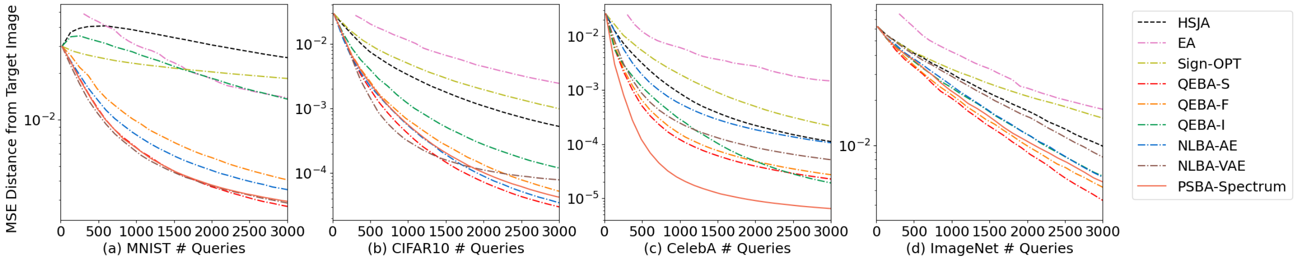

Figure 21. The perturbation magnitude w.r.t. different number of queries for different scales chosen on spectrum domain.

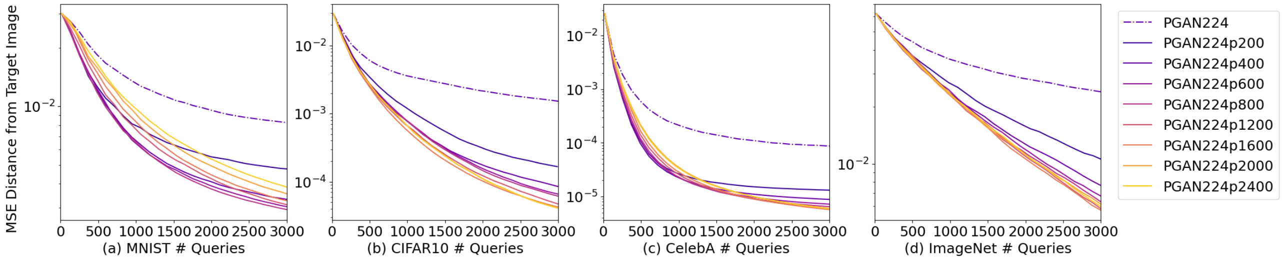

Figure 22. The perturbation magnitude w.r.t. different number of queries for different methods, the PSBA here is applied on spectrum domain.

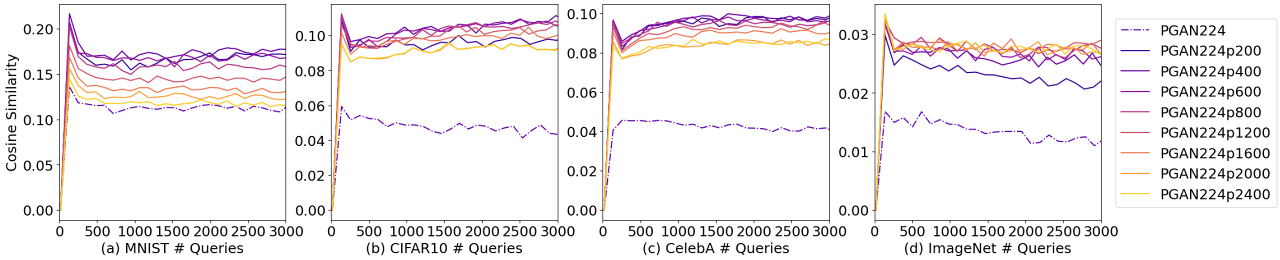

Figure 23. The cosine similarity between the estimated and true gradients for different scales chosen on spectrum domain.

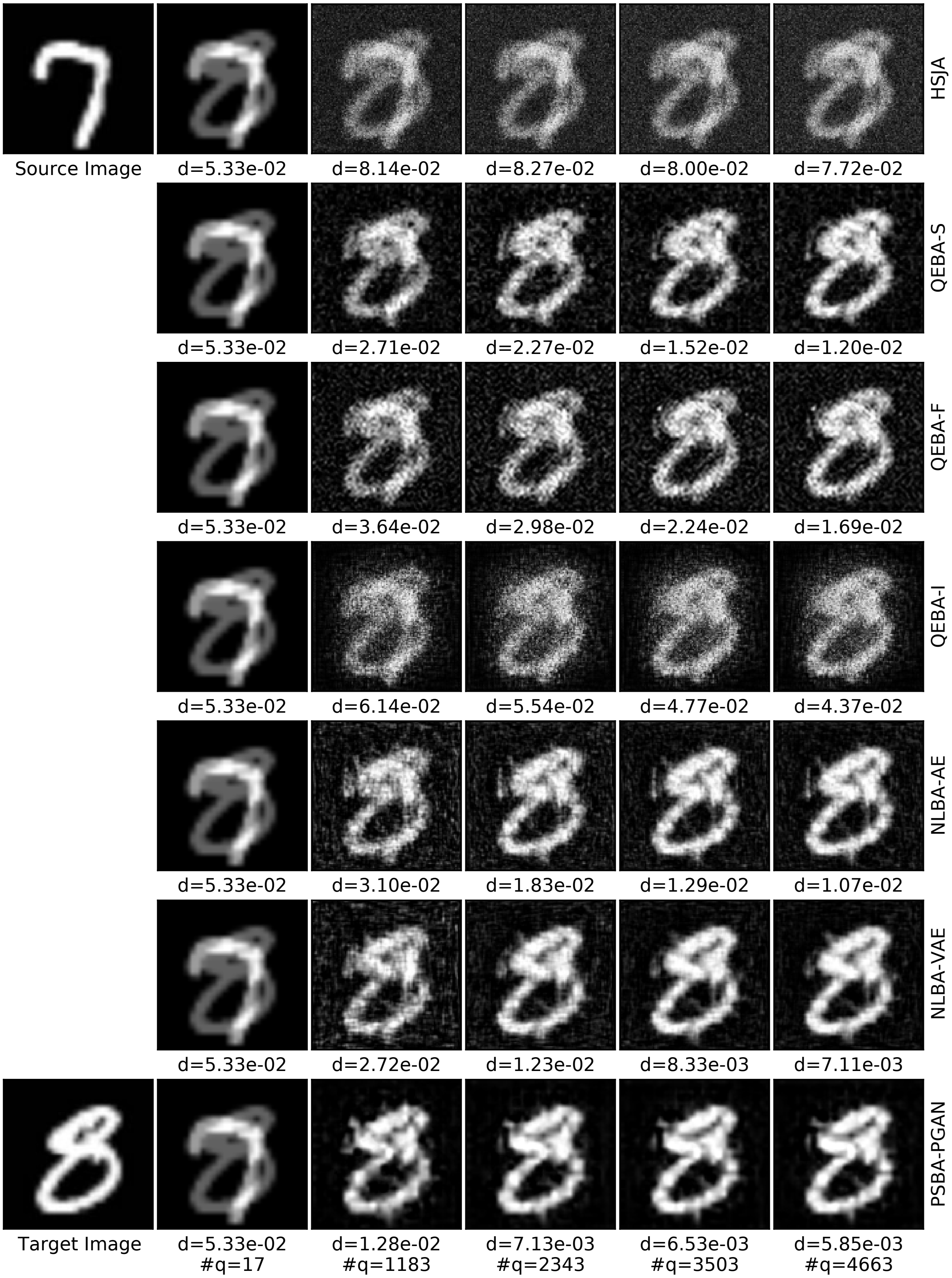

Figure 24. The qualitative case study of attacking ResNet-18 model on MNIST dataset.

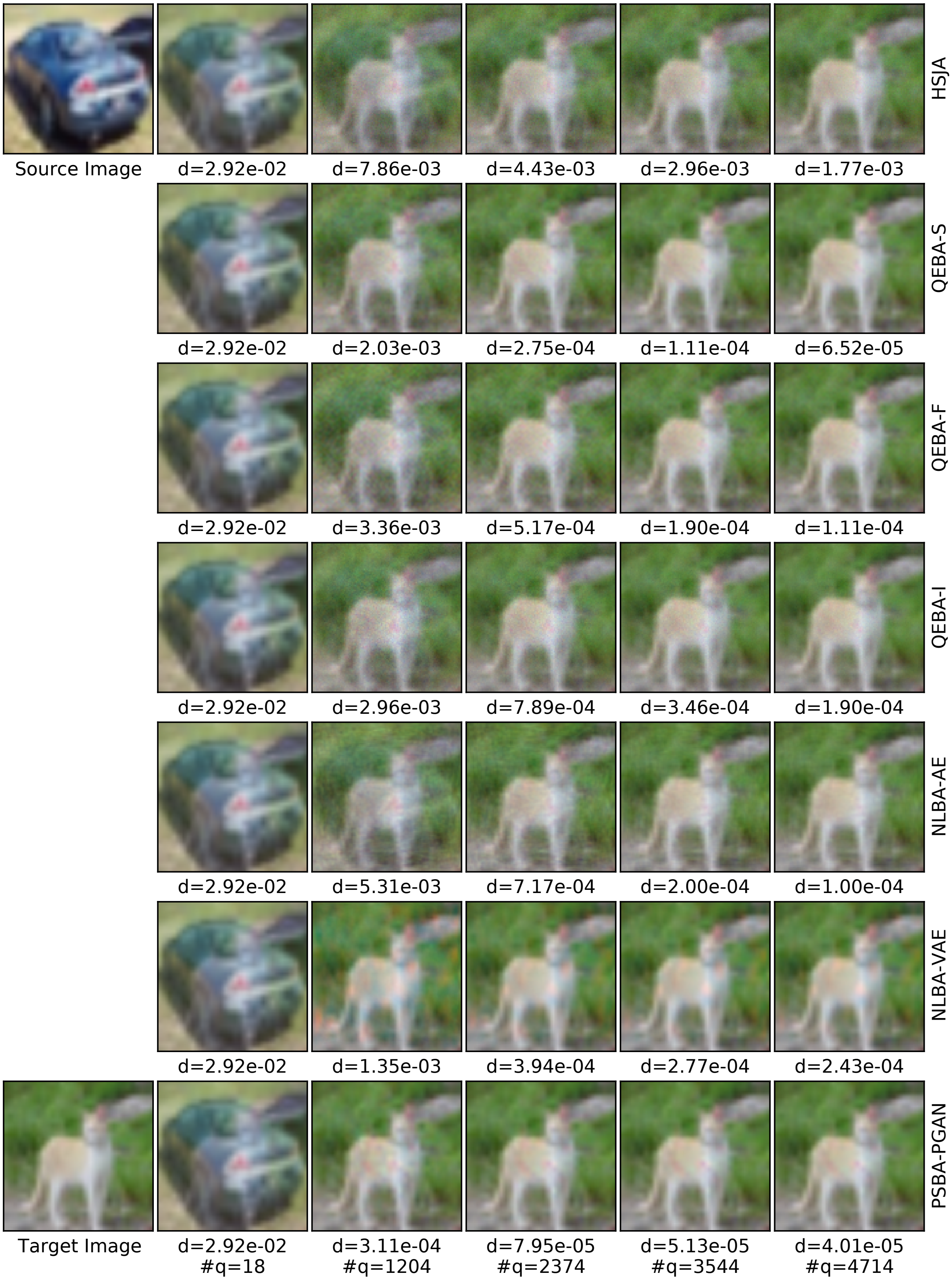

Figure 25. The qualitative case study of attacking ResNet-18 model on CIFAR-10 dataset.

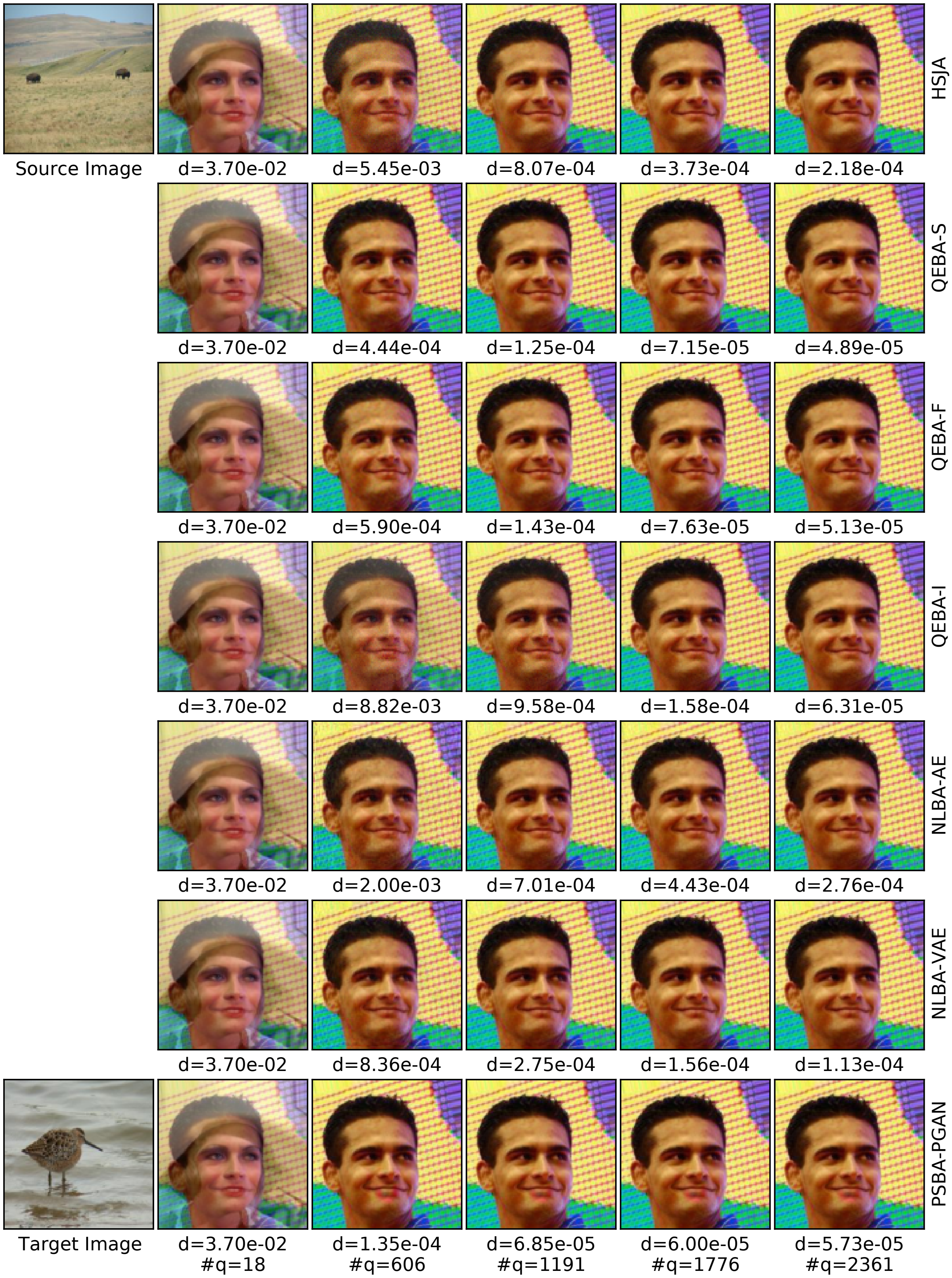

Figure 26. The qualitative case study of attacking ResNet-18 model on CelebA dataset.

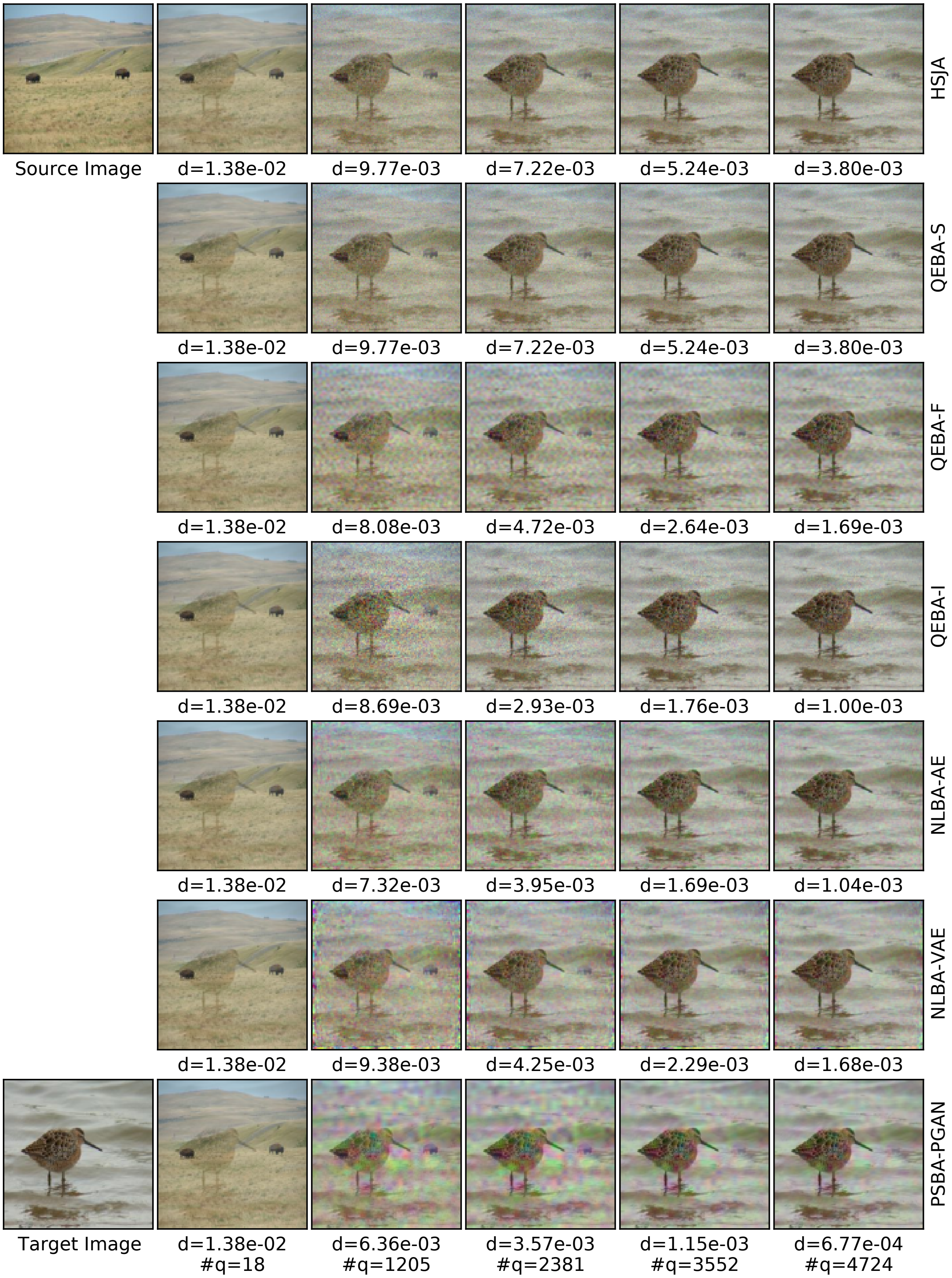

Figure 27. The qualitative case study of attacking ResNet-18 model on ImageNet dataset.

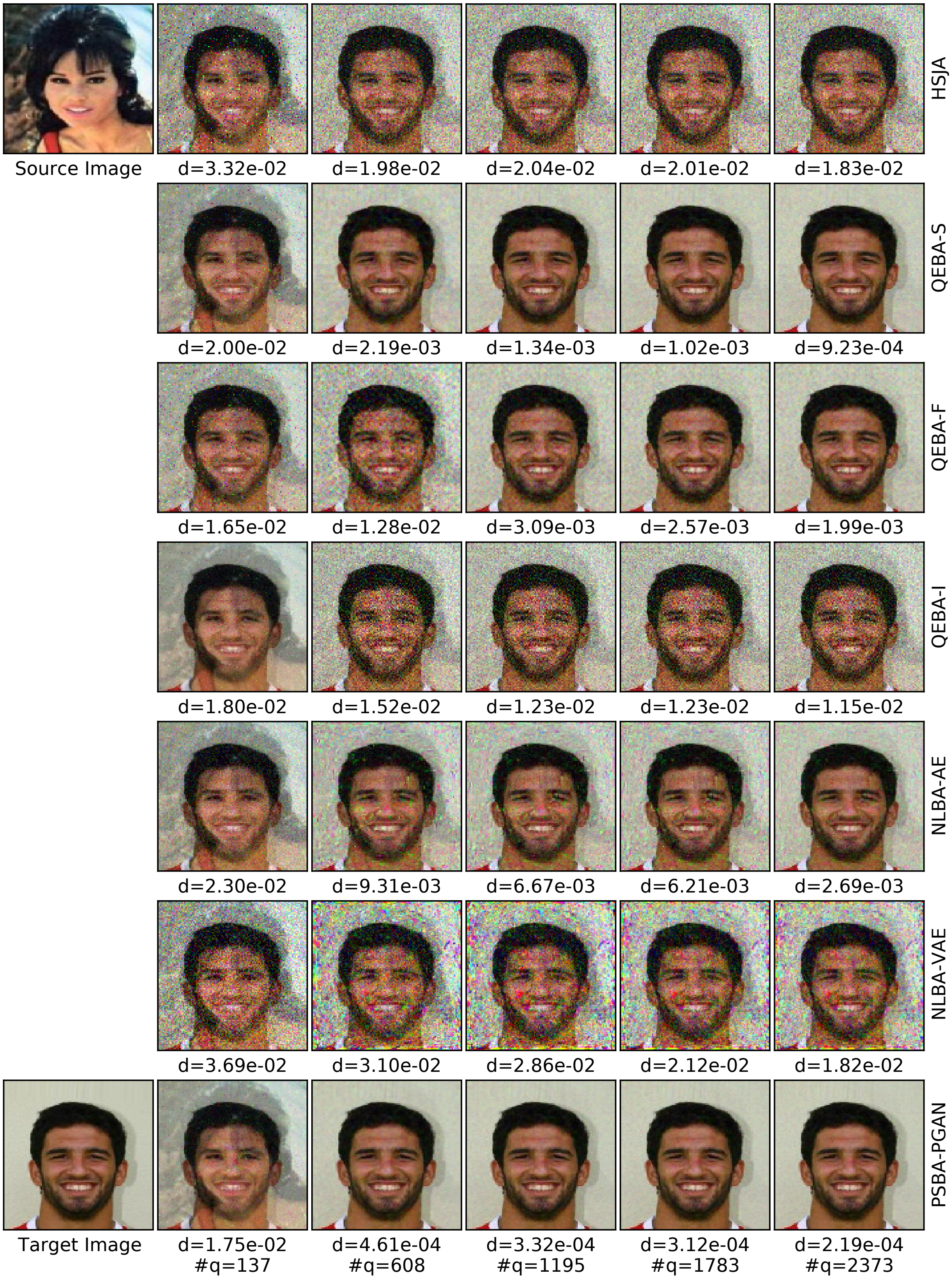

Figure 28. A case study of Face++ online API attack process. The source-target image pair is randomly sampled from CelebA dataset (ID: 019862 and 168859).

References

Bhagoji, A. N., He, W., Li, B., and Song, D. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *European Conference on Computer Vision*, pp. 158–174. Springer, 2018.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

Cai, Z., Fan, Q., Feris, R. S., and Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pp. 354–370. Springer, 2016.

Chen, J. and Gu, Q. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1739–1747, 2020.

Chen, J., Jordan, M. I., and Wainwright, M. J. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE symposium on security and privacy (SP)*, pp. 1277–1294. IEEE, 2020.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. 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*, pp. 15–26, 2017.

Cheng, M., Singh, S., Chen, P. H., Chen, P.-Y., Liu, S., and Hsieh, C.-J. Sign-opt: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2019a.

Cheng, M., Singh, S., Chen, P., Chen, P.-Y., Liu, S., and Hsieh, C.-J. Sign-opt: A query-efficient hard-label adversarial attack, 2020.

Cheng, S., Dong, Y., Pang, T., Su, H., and Zhu, J. Improving black-box adversarial attacks with a transfer-based prior. In *Advances in Neural Information Processing Systems*, pp. 10932–10942, 2019b.

Clarke, F. H., Ledyaev, Y. S., Stern, R. J., and Wolenski, P. R. *Nonsmooth analysis and control theory*, volume 178. Springer Science & Business Media, 2008.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., and Zhu, J. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7714–7722, 2019.

Fiers, T. Why are randomly drawn vectors nearly perpendicular in high dimensions. Mathematics Stack Exchange, 2018. URL https://math.stackexchange.com/q/995678. URL:https://math.stackexchange.com/q/995678 (version: 2018-05-15).

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Guo, Y., Yan, Z., and Zhang, C. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015a.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks, 2018.

Huang, L., Liu, L., Zhu, F., Wan, D., Yuan, Z., Li, B., and Shao, L. Controllable orthogonalization in training dnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6429–6438, 2020.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pp. 2137–2146, 2018.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation, 2018.

Krizhevsky, A., Hinton, G., et al. *Learning multiple layers of features from tiny images*. Citeseer, 2009.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, H., Xu, X., Zhang, X., Yang, S., and Li, B. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020a.

Li, H., Li, L., Xu, X., Zhang, X., Yang, S., and Li, B. Nonlinear gradient estimation for query efficient blackbox attack. In *Proceedings of 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Li, J., Ji, R., Liu, H., Liu, J., Zhong, B., Deng, C., and Tian, Q. Projection & probability-driven black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 362–371, 2020b.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Ma, W., Wu, Y., Cen, F., and Wang, G. Mdfn: Multi-scale deep feature learning network for object detection. *Pattern Recognition*, 100:107149, 2020.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

MEGVII. Face++ facial recognition 'compare' API documentation. https://console.faceplusplus.com/documents/5679308, 2020a.

MEGVII. Face++ facial recognition 'compare' API query URL. https://api-us.faceplusplus.com/facepp/v3/compare, 2020b.

MEGVII. Face++. https://www.faceplusplus.com/, 2020c.

Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

PyTorch. Torchvision.models. https://pytorch.org/docs/stable/torchvision/models.html, 2020.

Research, F. Pytorch GAN zoo. https://github.com/facebookresearch/pytorch_GAN_zoo, 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

Suya, F., Chi, J., Evans, D., and Tian, Y. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 1327–1344, 2020.

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

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions, 2014.

Tashiro, Y., Song, Y., and Ermon, S. Diversity can be transferred: Output diversification for white-and black-box attacks. *Advances in Neural Information Processing Systems*, 33, 2020.

Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 742–749, 2019.

Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NIPS*, 2017.

Wu, R., Zhang, G., Lu, S., and Chen, T. Cascade ef-gan: Progressive facial expression editing with local focuses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5021–5030, 2020.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks, 2017.

Yang, G., Duan, T., Hu, E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, 2020.

Yin, D., Gontijo Lopes, R., Shlens, J., Cubuk, E. D., and Gilmer, J. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32:13276–13286, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks, 2017.

Zhang, D. and Khoreva, A. Progressive augmentation of gans. In *Advances in Neural Information Processing Systems*, pp. 6249–6259, 2019.

Zhang, J., Xie, Z., Sun, J., Zou, X., and Wang, J. A cascaded r-cnn with multiscale attention and imbalanced samples for traffic sign detection. *IEEE Access*, 8:29742–29754, 2020.