A Instantiations of Generalized Gradient Estimator

As discussed in Section 4.1, the generalized gradient estimator in Definition 2 unifies the boundary gradient estimator in HSJA (Chen et al., 2020), QEBA (Li et al., 2020), and our NonLinear-BA. In this section we discuss the instantiations of them in detail.

In the generalized gradient estimator, the $u_1, u_2, \ldots, u_B$ are a sampled subset of orthonormal basis, whereas in practice, all these methods only uniformly sample normalized vectors for efficiency concern. As implied by Lemma 1, when n becomes large, $\langle u_i, v\rangle$'s PDF is highly concentrated at $x = 0$, implying that with high probability the sampled normalized vectors are close to orthogonal. Therefore, the orthonormal basis sampling can be approximated by normalized vector sampling. With this mindset, we express each gradient estimator using generalized gradient estimator (Definition 2).

HSJA. At a boundary-image $x_{adv}^{(t)}$, the HSJA gradient estimator (Chen et al., 2020) is

$$\widetilde{\nabla S(x_{adv}^{(t)})} = \frac{1}{B}\sum_{b=1}^{B} \operatorname{sgn}\left(S\left(x_{adv}^{(t)} + \delta u_b\right)\right) u_b.$$

We define the projection $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ as an identical mapping. The gradient estimator reduces to

$$\widetilde{\nabla S}(\mathbf{f}(x_0)) = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0 + \delta u_i)\right)\right) \mathbf{f}(u_i) = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta u_i\right)\right) u_i, \tag{13}$$

which is exactly the HSJA gradient estimator.

QEBA. At a boundary-image $x_{adv}^{(t)}$, the QEBA gradient estimator (Li et al., 2020) is

$$\widetilde{\nabla S(x_{adv}^{(t)})} = \frac{1}{B}\sum_{b=1}^{B} \operatorname{sgn}\left(S\left(x_{adv}^{(t)} + \mathbf{W}\delta u_b\right)\right) \mathbf{W}u_b.$$

The $\mathbf{W} \in \mathbb{R}^{m\times n}$ is an orthogonal matrix. We define the projection $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ by $\mathbf{f}(v) = \mathbf{W}v + x_0$. Notice that $\mathbf{f}(0) = x_0$ is a boundary-image of difference function S. At the origin, the Equation (5) becomes

$$\widetilde{\nabla \mathbf{f}^\intercal \nabla S} = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{f}(\delta u_i)\right)\right) u_i = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta \mathbf{W}u_i\right)\right) u_i,$$

and the gradient estimator becomes

$$\widetilde{\nabla S}(\mathbf{f}(0)) = \mathbf{W}\widetilde{\nabla \mathbf{f}^\intercal \nabla S} = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta \mathbf{W}u_i\right)\right) \mathbf{W}u_i, \tag{14}$$

which is the QEBA gradient estimator.

NonLinear-BA. In NonLinear-BA, a nonlinear projection $\mathbf{f}$ is already trained. The gradient estimation uses Equation (2). To bridge the gap between Equation (2) and the generalized gradient estimator in Equation (5), we define a new projection $\mathbf{g}$ such that $\mathbf{g}(v) = x_0 + \|v\|\mathbf{f}\left(v/\|v\|\right)$. We assume that $\mathbf{f}$ is highly linear within the L_2 ball $\{r : \|r\| \le 1\}$. Therefore, $\nabla\mathbf{g}(0)$ exists, and for normalized vector u_i, $\mathbf{g}(u_i) - \mathbf{g}(0) \approx \nabla\mathbf{g}(0)u_i$. Notice that $\mathbf{g}(u_i) = x_0 + \mathbf{f}(u_i)$ and $\mathbf{g}(0) = x_0$, so $\mathbf{f}(u_i) \approx \nabla\mathbf{g}(0)u_i$.

We apply generalized gradient estimator with projection $\mathbf{g}$ at the boundary-image $\mathbf{g}(0) = x_0$:

$$\widetilde{\nabla S}(\mathbf{g}(0)) = \nabla\mathbf{g}(0)\left(\frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{g}(\delta u_i)\right)\right) u_i\right) = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta\mathbf{f}(u_i)\right)\right) \nabla\mathbf{g}(0)u_i \tag{15}$$

$$\approx \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta\mathbf{f}(u_i)\right)\right) \mathbf{f}(u_i), \tag{16}$$

where the Equation (16) is the NonLinear-BA gradient estimator in Equation (2). We implement NonLinear-BA gradient estimator by Equation (16) instead of the precise Equation (15) to avoid gradient computation and improve the efficiency.

Notice that in all these methods we perform boundary attack iterations in the raw input space. However, for the gradient estimation, QEBA and NonLinear-BA use low dimension space while HSJA uses raw input space. To reflect the boundary point x_0 found in raw input space, in QEBA and NonLinear-BA, the projection is defined as the difference from the bounadry image x_0, i.e., $\mathbf{f}(0) = x_0$ and the gradient estimation is for $\mathbf{f}(0)$. In this way, we circumvent the possible sparsity of the boundary-images in low dimension space.

In summary, all these gradient estimators are instances of generalized gradient estimator in Definition 2. Moreover, we can observe that HSJA and QEBA use linear projection, and NonLinear-BA permits nonlinear projection.

B Proof of Cosine Similarity Bounds

In this section, we prove the universal cosine similarity bounds as shown in Theorem 1. The proof is derived from careful analysis of the distribution of randomly sampled orthonormal basis, combining with Taylor expansion and breaking down the cosine operator.

Lemma 1. *Let $u_1, u_2, \ldots, u_B$ be randomly chosen subset of orthonormal basis of $\mathbb{R}^n$ ($B \leq n$). Let v be any fixed unit vector in $\mathbb{R}^n$. For any $i \in [B]$, define $a_i := \langle u_i, v\rangle$. Then each a_i follows the distribution p_a with PDF*

$$p_a(x) := \frac{(1-x^2)^{(n-3)/2}}{\mathcal{B}\left(\frac{n-1}{2}, \frac{1}{2}\right)}, \quad x \in [-1, 1], \tag{17}$$

where $\mathcal{B}$ is the Beta function.

Remark. Lemma 1 shows the distribution of projection of orthonormal base vector on arbitrary normalized vector. Later we will apply the lemma to any normalized vector.

Proof of Lemma 1. Since u_i is the randomly chosen orthonormal base vector, the marginal distribution of each u_i is the uniform distribution sampled from $(n-1)$-unit sphere. As a result, for any unit vector v, the distribution of $\langle u_i, v\rangle$ should be the same. Consider $e_1 = (1, 0, 0, \ldots, 0)^\intercal$,

$$a_i = \langle u_i, e_1\rangle = u_{i1}. \tag{18}$$

Now consider the distribution of u_{i1}, i.e., the first component of u_i. We know that $u_{i1} = x_1/\sqrt{x_1^2 + \cdots + x_n^2}$ where each $x_i \sim \mathcal{N}(0,1)$ independently (Muller, 1959; Marsaglia et al., 1972). Therefore, let $X \sim \mathcal{N}(0,1)$, and $Y \sim \chi^2(n-1)$, $u_{i1} = X/\sqrt{X^2+Y}$. Denote $f(x)$ to the PDF of u_{i1}, from calculus, we obtain

$$f(x) = \int_0^\infty \frac{y^{\frac{n-1}{2}-1}\exp\left(-\frac{y}{2}\right)}{2^{\frac{n-1}{2}}\Gamma\left(\frac{n-1}{2}\right)} \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2 y}{2(1-x^2)}\right) \frac{\sqrt{y}}{(1-x^2)^{-3/2}} \mathrm{d}y = \frac{(1-x^2)^{\frac{n-3}{2}}}{\mathcal{B}\left(\frac{n-1}{2}, \frac{1}{2}\right)} \tag{19}$$

for $x \in (-1, 1)$. Combining Equation (18) and Equation (19), we have

$$p_a(x) = \frac{(1-x^2)^{(n-3)/2}}{\mathcal{B}\left(\frac{n-1}{2}, \frac{1}{2}\right)}, \; x \in [-1, 1].$$

□

Lemma 2. *Define ω as in Definition 5. Let $\mathbf{f}(x_0)$ be a boundary-image. The projection $\mathbf{f}$ and the difference function S satisfy the assumptions in Section 4.1. Let*

$$J := \nabla\mathbf{f}(x_0),\ \nabla S := \nabla S\left(\mathbf{f}(x_0)\right),\ \text{and } v := \frac{J^\intercal \nabla S}{\|J^\intercal \nabla S\|_2}.$$

When $0 < \delta \ll 1$, for any unit vector $u \in \mathbb{R}^n$,

$$\langle u, v\rangle > \frac{\omega}{\|J^\intercal \nabla S\|_2} \implies \operatorname{sgn}\left(S\left(\mathbf{f}(x_0 + \delta u)\right)\right) = 1,$$
$$\langle u, v\rangle < -\frac{\omega}{\|J^\intercal \nabla S\|_2} \implies \operatorname{sgn}\left(S\left(\mathbf{f}(x_0 + \delta u)\right)\right) = -1.$$

Remark. The Lemma 2 reveals that $\langle u, v\rangle$ in some degree aligns with the sign of $S(\mathbf{f}(x_0+\delta u))$. Later, we will write the cosine similarity as the sum of the product $\langle u, v\rangle \mathrm{sgn}\left(\mathbf{f}(x_0+\delta u)\right)$. Such alignment, along with Lemma 1, provides the bound for this sum of the product.

Proof of Lemma 2. We do Taylor expansion at point x_0 and $\mathbf{f}(x_0)$ for $\mathbf{f}$ and S to the second order respectively using Lagrange remainder:

$$\mathbf{f}(x_0+\delta u) = \mathbf{f}(x_0) + J\cdot\delta u + \frac{1}{2}\sum_{i=1}^{n}(\theta\delta u)^\intercal \mathbf{T}(x_0)_i(\theta\delta u) = \mathbf{f}(x_0) + \delta J u + \frac{1}{2}\beta_{\mathbf{f}}\delta^2\epsilon, \tag{20}$$

$$S\left(\mathbf{f}(x_0+\delta u)\right) = S\left(\mathbf{f}(x_0)\right) + \nabla S^\intercal\left(\delta J u + \frac{1}{2}\beta_{\mathbf{f}}\delta^2\epsilon\right) + \frac{1}{2}\beta_S\left(\delta L_{\mathbf{f}} + \frac{1}{2}\beta_{\mathbf{f}}\delta^2\right)^2\theta_1 \tag{21}$$

$$= \delta\nabla S^\intercal J u + \delta^2\left(\frac{1}{2}\beta_{\mathbf{f}}L_S + \frac{1}{2}\beta_S L_{\mathbf{f}}^2 + \frac{1}{2}\delta\beta_{\mathbf{f}}\beta_S L_{\mathbf{f}} + \frac{1}{8}\delta^2\beta_S\beta_{\mathbf{f}}^2\right)\theta_2. \tag{22}$$

In above expressions, $\theta\in[0,\,1]$, $\theta_1,\theta_2\in[-1,\,1]$, $\epsilon\in\mathbb{R}^m$ is an error vector such that $\|\epsilon\|_2\le 1$.

In Equation (20), we use the smoothness condition of $\mathbf{f}$, which leads to $\|\sum_{i=1}^n v^\intercal\mathbf{T}(x_0)_i v\|_2 \le \beta_{\mathbf{f}}\|v\|_2^2$, where $\mathbf{T}$ is the second-order gradient tensor, i.e., $\mathbf{T}(x)_{ijk} = \partial\mathbf{f}(x)_i/\left(\partial x_j\partial x_k\right)$. In Equation (21), similarly, the smoothness condition of S leads to $v^\intercal\mathbf{H}v\le\beta_S\|v\|_2^2$ where $\mathbf{H}$ is the Hessian matrix of S and its spectral radius is bounded by β_S. We let $v=\delta J u+\frac{1}{2}\beta_{\mathbf{f}}\delta^2\epsilon$ and observe that $\|v\|_2\le\|\delta J u\|_2+\frac{1}{2}\beta_{\mathbf{f}}\delta^2\le\delta L_{\mathbf{f}}+\frac{1}{2}\beta_{\mathbf{f}}\delta^2$. From Taylor expansion we get Equation (21). Equation (22) follows from $S(\mathbf{f}(x_0))=0$ by the boundary condition and $\nabla S^\intercal v\le L_S\|v\|_2$ by the Lipschitz condition.

Consider the expression in the parenthesis of Equation (22), we have

$$0\le\frac{1}{2}\beta_{\mathbf{f}}L_S+\frac{1}{2}\beta_S L_{\mathbf{f}}^2+\frac{1}{2}\delta\beta_{\mathbf{f}}\beta_S L_{\mathbf{f}}+\frac{1}{8}\delta^2\beta_S\beta_{\mathbf{f}}^2=\omega/\delta,$$

where ω is as defined in Definition 5. As a result, we rewrite Equation (22) as

$$S(\mathbf{f}(x_0+\delta u)) = \delta\nabla S^\intercal J u+\delta\omega\theta_2.$$

Given that $\theta_2\in[-1,\,1]$, $S\left(\mathbf{f}(x_0+\delta u)\right)$ can be bounded:

$$\delta\nabla S^\intercal J u-\delta\omega\le S\left(\mathbf{f}(x_0+\delta u)\right)\le\delta\nabla S^\intercal J u+\delta\omega.$$

Since $\nabla S^\intercal J u=(J^\intercal\nabla S)^\intercal u=\|J^\intercal\nabla S\|_2\langle u,\,v\rangle$, we rewrite the bound as:

$$\delta\left(\|J^\intercal\nabla S\|_2\langle u,\,v\rangle-\omega\right)\le S\left(\mathbf{f}(x_0+\delta u)\right)\le\delta\left(\|J^\intercal\nabla S\|_2\langle u,\,v\rangle+\omega\right).$$

Thus, when $\|J^\intercal\nabla S\|_2\langle u,\,v\rangle-\omega>0$, i.e., $\langle u,\,v\rangle>\omega/\|J^\intercal\nabla S\|_2$, $S\left(\mathbf{f}(x_0+\delta u)\right)>0$; when $\|J^\intercal\nabla S\|_2\langle u,\,v\rangle+\omega<0$, i.e., $\langle u,\,v\rangle<-\omega/\|J^\intercal\nabla S\|_2$, $S\left(\mathbf{f}(x_0+\delta u)\right)<0$, which concludes the proof. □

Lemma 3. *Let $\mathbf{f}(x_0)$ be a boundary-image, i.e., $S\left(\mathbf{f}(x_0)\right)=0$. The projection $\mathbf{f}$ and the difference function S satisfy the assumptions in Section 4.1. Over the randomness of the sampling of orthogonal basis subset $u_1,u_2,\ldots,u_B$ for $\mathbb{R}^n$ space, The expectation of cosine similarity between $\widetilde{\nabla\mathbf{f}^\intercal\nabla S}$ (defined as Equation (5)) and $\nabla\mathbf{f}(x_0)^\intercal\nabla S\left(\mathbf{f}(x_0)\right)$ ($\nabla\mathbf{f}^\intercal\nabla S$ for short) satisfies*

$$\left(2\left(1-\frac{\omega^2}{\|\nabla\mathbf{f}^\intercal\nabla S\|_2^2}\right)^{(n-1)/2}-1\right)\cdot\frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\,\frac{1}{2}\right)\cdot(n-1)}\le\mathbb{E}\cos\langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\,\nabla\mathbf{f}^\intercal\nabla S\rangle\le\frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\,\frac{1}{2}\right)\cdot(n-1)}. \tag{23}$$

Here, ω is as defined in Definition 5, and we assume $\omega\le\|\nabla\mathbf{f}^\intercal\nabla S\|_2$.

Remark. This theorem directly relates the intermediate gradient estimation $\widetilde{\nabla\mathbf{f}^\intercal\nabla S}$ to the mapped true gradient $\nabla\mathbf{f}^\intercal\nabla S$ by providing general cosine similarity bounds between them. The assumption that $\omega\le\|\nabla\mathbf{f}^\intercal\nabla S\|_2$ can be easily achieved since δ is typically small and $\lim_{\delta\to0}\omega/\delta$ is a constant.

Proof of Lemma 3. According to Equation (5),

$$\widetilde{\nabla \mathbf{f}^\intercal \nabla S} = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right) u_i.$$

Define $J := \nabla \mathbf{f}(x_0)$. Since $u_1, u_2, \ldots, u_B$ is a subset of the orthonormal basis,

$$\begin{aligned}\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S},\, \nabla \mathbf{f}^\intercal \nabla S\rangle &= \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right) \langle J^\intercal \nabla S,\, u_i\rangle \\ &= \frac{\|J^\intercal \nabla S\|_2}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right) \left\langle \frac{J^\intercal \nabla S}{\|J^\intercal \nabla S\|_2},\, u_i\right\rangle.\end{aligned}$$

Let $v := J^\intercal \nabla S / \|J^\intercal \nabla S\|_2$. Note that $\left\|\widetilde{\nabla \mathbf{f}^\intercal \nabla S}\right\|_2 = \sqrt{\sum_{i=1}^{B}(1/B)^2} = 1/\sqrt{B}$, we have

$$\cos\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S},\, \nabla \mathbf{f}^\intercal \nabla S\rangle = \frac{\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S},\, \nabla \mathbf{f}^\intercal \nabla S\rangle}{\|\widetilde{\nabla \mathbf{f}^\intercal \nabla S}\|_2 \|\nabla \mathbf{f}^\intercal \nabla S\|_2} = \frac{1}{\sqrt{B}}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right) \langle v,\, u_i\rangle. \tag{24}$$

According to Lemma 1, $\langle v,\, u_i\rangle$ follows the distribution p_a. Intuitively, we know that $\langle v,\, u_i\rangle$ in some degree decides $\operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)$.

Consider each component $\left(\operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)\langle v,\, u_i\rangle\right)$. By Lemma 2, in the worst case, only when $\|\langle v,\, u_i\rangle\| > \omega/\|J^\intercal \nabla S\|_2$, the $\operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)$ is aligned with the sign of $\langle v,\, u_i\rangle$, otherwise their signs are always different. Since $\omega/\|J^\intercal \nabla S\|_2 \le 1$,

$$\begin{aligned}&\mathbb{E}_{u_i} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)\langle v,\, u_i\rangle \\ \ge& \int_{-1}^{-\omega/\|J^\intercal \nabla S\|_2} -x p_a(x)\mathrm{d}x + \int_{-\omega/\|J^\intercal \nabla S\|_2}^{0} x p_a(x)\mathrm{d}x + \int_{0}^{\omega/\|J^\intercal \nabla S\|_2} -x p_a(x)\mathrm{d}x + \int_{\omega/\|J^\intercal \nabla S\|_2}^{1} x p_a(x)\mathrm{d}x \\ =& \int_{0}^{\omega/\|J^\intercal \nabla S\|_2} -2x p_a(x)\mathrm{d}x + \int_{\omega/\|J^\intercal \nabla S\|_2}^{1} 2x p_a(x)\mathrm{d}x \\ =& \frac{2}{\mathcal{B}\left(\frac{n-1}{2},\, \frac{1}{2}\right)\cdot(n-1)}\left(2\left(1-\frac{\omega^2}{\|\nabla \mathbf{f}^\intercal \nabla S\|_2^2}\right)^{(n-1)/2} - 1\right).\end{aligned}$$

Here we use the fact that p_a is symmetric. Inject it into Equation (24):

$$\mathbb{E}\cos\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S},\, \nabla \mathbf{f}^\intercal \nabla S\rangle \ge \frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\, \frac{1}{2}\right)\cdot(n-1)}\left(2\left(1-\frac{\omega^2}{\|\nabla \mathbf{f}^\intercal \nabla S\|_2^2}\right)^{(n-1)/2} - 1\right). \tag{25}$$

On the other hand, the upper bound can be obtained by forcing $\langle v,\, u_i\rangle$ and $S\left(\mathbf{f}(x_0+\delta u_i)\right)$ be of the same sign everywhere, which means that

$$\mathbb{E}_{u_i} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)\langle v,\, u_i\rangle \le \int_{-1}^{0} -x p_a(x)\mathrm{d}x + \int_{0}^{1} x p_a(x)\mathrm{d}x = \int_{0}^{1} 2x p_a(x) = \frac{2}{\mathcal{B}\left(\frac{n-1}{2},\, \frac{1}{2}\right)\cdot(n-1)}.$$

Inject it into Equation (24):

$$\mathbb{E}\cos\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S},\, \nabla \mathbf{f}^\intercal \nabla S\rangle \le \frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\, \frac{1}{2}\right)\cdot(n-1)}. \tag{26}$$

□

Lemma 4. *For any positive integer $n \ge 2$, define*

$$c_n := \frac{2\sqrt{n}}{\mathcal{B}\left(\frac{n-1}{2},\, \frac{1}{2}\right)\cdot(n-1)},$$

where $\mathcal{B}$ is the Beta function. We have $c_n \in (2/\pi,\, 1)$ and $c_{n+2} < c_n$.

Remark. Using Lemma 4, we can simplify the term $2\sqrt{B}/\left(\mathcal{B}(\frac{n-1}{2}, \frac{1}{2})\cdot(n-1)\right)$ in Lemma 3 to $c_n\sqrt{B/n}$.

Proof of Lemma 4. Let $d_n := \Gamma\left(\frac{n}{2}\right)/\Gamma\left(\frac{n-1}{2}\right)$, where $\Gamma(\cdot)$ is the Gamma function. Notice that

$$c_n = \frac{2\sqrt{n}}{\mathcal{B}\left(\frac{n-1}{2}, \frac{1}{2}\right)\cdot(n-1)} = \frac{2\sqrt{n}\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})\sqrt{\pi}\cdot(n-1)} = d_n\frac{2\sqrt{n}}{(n-1)\sqrt{\pi}}.$$

(I.) For $n \geq 5$, $d_n = \dfrac{\Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{n-1}{2}\right)} = \dfrac{n-2}{n-3}\cdot\dfrac{\Gamma\left(\frac{n-2}{2}\right)}{\Gamma\left(\frac{n-3}{2}\right)} = \dfrac{n-2}{n-3}d_{n-2}$. Notice that

$$\frac{d_n}{\sqrt{n-2}} = \frac{\sqrt{n-2}}{n-3}d_{n-2} = \frac{\sqrt{(n-2)\cdot(n-4)}}{n-3}\cdot\frac{d_{n-2}}{\sqrt{n-4}} \leq \frac{d_{n-2}}{\sqrt{n-4}},$$

and

$$\frac{d_3}{\sqrt{1}} = \frac{\sqrt{\pi}}{2}, \frac{d_4}{\sqrt{2}} = \frac{2}{\sqrt{\pi}},$$

we have $\dfrac{d_n}{\sqrt{n-2}} \leq \dfrac{\sqrt{\pi}}{2}$ for $n \geq 3$. Therefore,

$$c_n = d_n\frac{2\sqrt{n}}{(n-1)\sqrt{\pi}} \leq \frac{\sqrt{\pi}}{2}\cdot\frac{2\sqrt{n(n-2)}}{(n-1)\sqrt{\pi}} < 1$$

for $n \geq 3$. When $n = 2$, $c_n = \dfrac{2\sqrt{2}}{\pi} < 1$. So $c_n < 1$ holds for any $n \geq 2$.

(II.) Similarly, notice that

$$\frac{d_n}{\sqrt{n-1}} = \frac{n-2}{(n-3)\sqrt{n-1}}d_{n-2} = \frac{n-2}{\sqrt{(n-3)(n-1)}}\cdot\frac{d_{n-2}}{\sqrt{n-3}} \geq \frac{d_{n-2}}{\sqrt{n-3}},$$

and

$$\frac{d_3}{\sqrt{2}} = \frac{1}{4}\sqrt{2\pi}, \frac{d_2}{\sqrt{1}} = \frac{1}{\sqrt{\pi}},$$

we have $\dfrac{d_n}{\sqrt{n-1}} \geq \dfrac{1}{\sqrt{\pi}}$ for $n \geq 2$. Therefore,

$$c_n = d_n\frac{2\sqrt{n}}{(n-1)\sqrt{\pi}} \geq \sqrt{\frac{n-1}{\pi}}\cdot\frac{2\sqrt{n}}{(n-1)\sqrt{\pi}} = \frac{2}{\pi}\sqrt{\frac{n}{n-1}} > \frac{2}{\pi}.$$

(III.) Since $d_{n+2} = d_n\cdot n/(n-1)$ and $c_n = d_n\cdot(2\sqrt{n})\,/\,((n-1)\sqrt{\pi})$, we have

$$\frac{c_{n+2}}{c_n} = \frac{d_{n+2}}{d_n}\cdot\frac{\sqrt{n+2}}{n+1}\cdot\frac{n-1}{\sqrt{n}} = \frac{n}{n-1}\cdot\frac{\sqrt{n+2}}{n+1}\cdot\frac{n-1}{\sqrt{n}} = \frac{\sqrt{n(n+2)}}{n+1} < 1.$$

In summary, for any positive integer $n \geq 2$, we have shown $2/\pi < c_n < 1$ and $c_{n+2} < c_n$. □

Now we are ready to prove the main theorem which provides the general cosine similarity bounds for our gradient estimator.

Theorem 1 (restated)**.** *Let $\mathbf{f}(x_0)$ be a boundary-image, i.e., $S\left(\mathbf{f}(x_0)\right) = 0$. The projection $\mathbf{f}$ and the difference function S satisfy the assumptions in Section 4.1. Over the randomness of the sampling of orthogonal basis subset $u_1, u_2, \ldots, u_B$ for $\mathbb{R}^n$ space, the expectation of cosine similarity between $\widetilde{\nabla S}\left(\mathbf{f}(x_0)\right)$ ($\widetilde{\nabla S}$ for short) and $\nabla S\left(\mathbf{f}(x_0)\right)$ (∇S for short) satisfies*

$$\left(2\left(1-\frac{\omega^2}{\|\nabla\mathbf{f}^\intercal\nabla S\|_2^2}\right)^{(n-1)/2} - 1\right)\frac{\|\nabla\mathbf{f}^\intercal\nabla S\|_2}{L_{\mathbf{f}}\|\nabla S\|_2}\sqrt{\frac{B}{n}}c_n \leq \mathbb{E}\cos\langle\widetilde{\nabla S}, \nabla S\rangle \leq \frac{\|\nabla\mathbf{f}^\intercal\nabla S\|_2}{l_{\mathbf{f}}\|\nabla S\|_2}\sqrt{\frac{B}{n}}c_n, \tag{27}$$

where ω is as defined in Definition 5, and we assume $\omega \leq \|\nabla\mathbf{f}^\intercal\nabla S\|_2$; $c_n \in (2/\pi, 1)$ is a constant depended on n; $L_{\mathbf{f}}$ is as defined in assumptions in Section 4.1; and $l_{\mathbf{f}} := \lambda_{\min}(\nabla\mathbf{f}(x_0))$.

Proof of Theorem 1. According to Equation (6), we know $\widetilde{\nabla S} = \nabla\mathbf{f}\widetilde{\nabla\mathbf{f}^\intercal\nabla S}$, where $\nabla\mathbf{f}$ is the shorthand of $\nabla\mathbf{f}(x_0)$. Thus,

$$\langle\widetilde{\nabla S},\, \nabla S\rangle = \widetilde{\nabla S}^\intercal\nabla S = \widetilde{\nabla\mathbf{f}^\intercal\nabla S}^\intercal\nabla\mathbf{f}^\intercal\nabla S = \langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\, \nabla\mathbf{f}^\intercal\nabla S\rangle = \cos\langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\, \nabla\mathbf{f}^\intercal\nabla S\rangle\cdot\left\|\widetilde{\nabla\mathbf{f}^\intercal\nabla S}\right\|_2\|\nabla\mathbf{f}^\intercal\nabla S\|_2.$$

Therefore,

$$\cos\langle\widetilde{\nabla S},\, \nabla S\rangle = \cos\langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\, \nabla\mathbf{f}^\intercal\nabla S\rangle\frac{\left\|\widetilde{\nabla\mathbf{f}^\intercal\nabla S}\right\|_2\|\nabla\mathbf{f}^\intercal\nabla S\|_2}{\left\|\widetilde{\nabla S}\right\|_2\|\nabla S\|_2}. \tag{28}$$

According to the estimation formula of $\widetilde{\nabla\mathbf{f}^\intercal\nabla S}$ (Equation (5)), $\left\|\widetilde{\nabla\mathbf{f}^\intercal\nabla S}\right\|_2 = \sqrt{B}$. Furthermore, $\left\|\widetilde{\nabla S}\right\| \le \lambda_{\max}(\nabla\mathbf{f})\cdot\left\|\widetilde{\nabla\mathbf{f}^\intercal\nabla S}\right\|_2 \le L_\mathbf{f}\sqrt{B}$, $\left\|\widetilde{\nabla S}\right\| \ge \lambda_{\min}(\nabla\mathbf{f})\cdot\left\|\widetilde{\nabla\mathbf{f}^\intercal\nabla S}\right\|_2 = l_\mathbf{f}\sqrt{B}$, which means that

$$\frac{1}{L_\mathbf{f}} \le \frac{\left\|\widetilde{\nabla\mathbf{f}^\intercal\nabla S}\right\|_2}{\left\|\widetilde{\nabla S}\right\|_2} \le \frac{1}{l_\mathbf{f}}.$$

According to Equation (28), we have

$$\cos\langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\, \nabla\mathbf{f}^\intercal\nabla S\rangle\frac{\|\nabla\mathbf{f}^\intercal\nabla S\|_2}{L_\mathbf{f}\|\nabla S\|_2} \le \cos\langle\widetilde{\nabla S},\, \nabla S\rangle \le \cos\langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\, \nabla\mathbf{f}^\intercal\nabla S\rangle\frac{\|\nabla\mathbf{f}^\intercal\nabla S\|_2}{l_\mathbf{f}\|\nabla S\|_2}. \tag{29}$$

Inject the bound for $\mathbb{E}\cos\langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\, \nabla\mathbf{f}^\intercal\nabla S\rangle$ in Lemma 3 and the simplification from Lemma 4 to Equation (29) yields the desired bound. □

We discuss the implications of the bound in Section 4.2 and Appendix D.

Corollary 1 (restated)**.** *Let $\mathbf{f}(x_0)$ be a boundary-image, i.e., $S(\mathbf{f}(x_0)) = 0$. The projection $\mathbf{f}$ is locally linear around x_0 with radius δ. $L_\mathbf{f} := \lambda_{\max}(\nabla\mathbf{f}(x_0)), l_\mathbf{f} := \lambda_{\min}(\nabla\mathbf{f}(x_0))$. The difference function S satisfies the assumptions in Section 4.1. Over the randomness of the sampling of orthogonal basis subset $u_1, u_2, \ldots, u_B$ for $\mathbb{R}^n$ space, the expectation of cosine similarity between $\widetilde{\nabla S}(\mathbf{f}(x_0))$ ($\widetilde{\nabla S}$ for short) and $\nabla S(\mathbf{f}(x_0))$ (∇S for short) satisfies Equation* (8) *with*

$$\omega := \frac{1}{2}\delta\beta_S L_\mathbf{f}^2. \tag{30}$$

We assume $\omega \le \|\nabla\mathbf{f}^\intercal\nabla S\|_2$. The $c_n \in (2/\pi,\, 1)$ is a constant depended on n.

Remark. This is a direct application of Theorem 1. Since $\mathbf{f}$ is locally linear, we have $\beta_\mathbf{f} = 0$, and the corollary follows. We discuss its implication in Section 4.2.

Corollary 2 (restated)**.** *Given the projection $\mathbf{f}$ and the difference function S, to achieve expected cosine similarity $\mathbb{E}\langle\nabla S(\mathbf{f}(x_0)),\, \widetilde{\nabla S}(\mathbf{f}(x_0))\rangle = s$, the required query number B is in $\Theta(s^2)$.*

Proof of Corollary 2. From Theorem 1, we can observe that

$$\Theta(\sqrt{B}) \le \mathbb{E}\cos\langle\widetilde{\nabla S},\, \nabla S\rangle \le \Theta(\sqrt{B}).$$

Therefore, when $\mathbb{E}\cos\langle\widetilde{\nabla S},\, \nabla S\rangle = s$, the number of queries B is in $\Theta(s^2)$. □

Remark. The above corollary shows the relation between the expected cosine similarity and the query number when the projection $\mathbf{f}$ is fixed. Note that the cosine similarity is bounded, i.e., the cosine similarity between two totally aligned vectors is 1. The $\Theta(s^2)$ order implies that to achieve moderate cosine similarity, a small number of queries is needed, while high cosine similarity needs much more queries. Therefore, to achieve high cosine similarity, it is better to fix the number of queries and reduce the dimension of subspace, n, which is related with cosine similarity with order $\Theta(1/\sqrt{n})$. The reduction on subspace dimension is the shared technique between QEBA and NonLinear-BA.

C Proof of Existence of Better Nonlinear Projection

Theorem 2 (restated)**.** *Let $\mathbf{f}(x_0)$ be a boundary-image, i.e., $S(\mathbf{f}(x_0)) = 0$. The projection $\mathbf{f}$ is locally linear around x_0 with radius δ. $L_{\mathbf{f}} := \lambda_{\max}(\nabla \mathbf{f}(x_0))$, $l_{\mathbf{f}} := \lambda_{\min}(\nabla \mathbf{f}(x_0))$. The difference function S satisfies the assumptions in Section 4.1.*

There exists a nonlinear projection $\mathbf{f}'$ satisfying the assumptions in Section 4.1, with $\mathbf{f}'(x_0) = \mathbf{f}(x_0)$ and $\nabla \mathbf{f}'(x_0) = \nabla \mathbf{f}(x_0)$, such that over the randomness of the sampling of orthogonal basis subset $u_1, u_2, \ldots, u_B$ for $\mathbb{R}^n$ space, the expectation of cosine similarity between $\widetilde{\nabla S}(\mathbf{f}'(x_0))$ ($\widetilde{\nabla S}$ for short) and $\nabla S(\mathbf{f}'(x_0))$ (∇S for short) satisfies Equation (8) with

$$\omega := \frac{1}{2}\delta\beta_S L_{\mathbf{f}}^2 - \frac{1}{5}\beta_{\mathbf{f}}\beta_S\delta^2 L_{\mathbf{f}} < \frac{1}{2}\delta\beta_S L_{\mathbf{f}}^2. \tag{31}$$

We assume $\omega \le \|\nabla \mathbf{f}^\intercal \nabla S\|_2$. The $c_n \in (2/\pi, 1)$ is a constant depended on n.

Proof of Theorem 2. For convenience, in the proof, we define $J := \nabla \mathbf{f}(x_0)$. According to the proof of Theorem 1 (especially the usage of Lemma 2), we only need to show that for arbitrary S, there exists a projection $\mathbf{f}'$ such that $\nabla \mathbf{f}'(x_0) = \nabla \mathbf{f}(x_0) = J$, $\mathbf{f}'(x_0) = \mathbf{f}(x_0)$ and $\mathbf{f}'$ satisfies the smoothness and Lipschitz assumptions, so that for arbitrary vector u with $\|u\|_2 = 1$,

$$\begin{aligned}
\left\langle u, \frac{J^\intercal \nabla S}{\|J^\intercal \nabla S\|_2} \right\rangle > \frac{\omega}{\|J^\intercal \nabla S\|_2} &\implies \operatorname{sgn}(S(\mathbf{f}(x_0 + \delta u))) = 1, \\
\left\langle u, \frac{J^\intercal \nabla S}{\|J^\intercal \nabla S\|_2} \right\rangle < \frac{\omega}{\|J^\intercal \nabla S\|_2} &\implies \operatorname{sgn}(S(\mathbf{f}(x_0 + \delta u))) = -1.
\end{aligned} \tag{32}$$

We prove this by construction: we define $\mathbf{f}' : \mathbb{R}^n \to \mathbb{R}^m$ such that for arbitrary $u \in \mathbb{R}^n$,

$$\mathbf{f}'(x_0 + u) = \mathbf{f}(x_0) + J \cdot u - \frac{1}{2}\alpha\|u\|_2 J u, \tag{33}$$

where $\alpha \in [0,\, 0.8\beta_{\mathbf{f}}/L_{\mathbf{f}}]$ is an adjustable parameter (it is later fixed to $0.8\beta_{\mathbf{f}}/L_{\mathbf{f}}$, but for the generality of the proof, we deem it as an adjustable parameter for now).

Fact 2.1. *The $\mathbf{f}'$ defined as in Equation (33): (1) has gradient J at point x_0, (2) is $L_{\mathbf{f}}$-Lipschitz, and (3) is $\beta_{\mathbf{f}}$-smooth around x_0 with radius δ.*

Proof of Fact 2.1.

Gradient at x_0. Since

$$\lim_{u\to 0} \frac{\left\|\frac{1}{2}\alpha\|u\|_2 J u\right\|_2}{\|u\|_2} = \frac{1}{2}\alpha \lim_{u\to 0}\|Ju\|_2 \le \frac{1}{2}\alpha L_{\mathbf{f}}\|u\|_2 = 0,$$

we have $\mathbf{f}'(x_0 + u) = \mathbf{f}'(x_0) + J \cdot u + o(u)$ so $\nabla \mathbf{f}' := \nabla \mathbf{f}'(x_0) = J$.

Lipschitz. Firstly, let us derive the gradient of $\mathbf{f}'$ at an arbitrary point. Because

$$\begin{aligned}
\frac{\partial \mathbf{f}'(x_0 + u)_i}{\partial u_j} &= J_{ij} - \frac{1}{2}\alpha \frac{\partial \left(\|u\|_2 J u\right)_i}{\partial u_j} = J_{ij} - \frac{1}{2}\alpha\left(\frac{u_j}{\|u\|_2}\sum_{k=1}^{n} J_{ik}u_k + \|u\|_2 J_{ij}\right) \\
&= \left(1 - \frac{1}{2}\alpha\|u\|_2\right) J_{ij} - \frac{\alpha}{2\|u\|_2}(J u u^\intercal)_{ij},
\end{aligned}$$

we have

$$\nabla \mathbf{f}'(x_0 + u) = \left(1 - \frac{1}{2}\alpha\|u\|_2\right) J - \frac{\alpha}{2\|u\|_2} J u u^\intercal. \tag{34}$$

We bound its maximum eigenvalue:

$$\lambda_{\max}\left(\nabla \mathbf{f}'(x_0 + u)\right) \le \left(1 - \frac{1}{2}\alpha\|u\|_2\right)\lambda_{\max}(J) + \frac{\alpha}{2\|u\|_2}\lambda_{\max}(J)\|u\|_2^2 = \lambda_{\max}(J) = L_{\mathbf{f}}.$$

Therefore, $\mathbf{f}'$ is $L_{\mathbf{f}}$-Lipschitz.

Smoothness. The smoothness part is more involved.

To show $\mathbf{f}'$ is $\beta_{\mathbf{f}}$-smooth, we need to consider arbitrary $u_1, u_2 \in \mathbb{R}^n$, and prove that

$$\frac{\lambda_{\max}\left(\nabla \mathbf{f}'(x_0+u_1) - \nabla \mathbf{f}'(x_0+u_2)\right)}{\|u_1-u_2\|_2} \le \beta_{\mathbf{f}}$$

always holds. From Equation (34),

$$\nabla \mathbf{f}'(x_0+u_1) - \nabla \mathbf{f}'(x_0+u_2) = \frac{\alpha}{2}(\|u_2\|_2 - \|u_1\|_2)J - \frac{\alpha}{2}J\left(\frac{u_1u_1^\top}{\|u_1\|_2} - \frac{u_2u_2^\top}{\|u_2\|_2}\right).$$

Thus,

$$\frac{\lambda_{\max}\left(\nabla \mathbf{f}'(x_0+u_1) - \nabla \mathbf{f}'(x_0+u_2)\right)}{\|u_1-u_2\|_2} \le \frac{\lambda_{\max}\left(\frac{\alpha}{2}(\|u_2\|_2 - \|u_1\|_2)J\right)}{\|u_1-u_2\|_2} + \frac{\alpha L_{\mathbf{f}}}{2} \cdot \underbrace{\frac{\lambda_{\max}\left(\frac{u_1u_1^\top}{\|u_1\|_2} - \frac{u_2u_2^\top}{\|u_2\|_2}\right)}{\|u_1-u_2\|_2}}_{(*)}.$$

Consider the first term: from $\left|\|u_2\|_2 - \|u_1\|_2\right| \le \|u_1-u_2\|_2$,

$$\frac{\lambda_{\max}\left(\frac{\alpha}{2}(\|u_2\|_2 - \|u_1\|_2)J\right)}{\|u_1-u_2\|_2} \le \frac{1}{2}\alpha L_{\mathbf{f}}.$$

Fact 2.2. *For arbitrary $u, v \in \mathbb{R}^n$,*

$$\lambda_{\max}\left(\frac{uu^\top}{\|u\|_2} - \frac{vv^\top}{\|v\|_2}\right) \le 1.5\|u-v\|_2.$$

From Fact 2.2, the second term $(*)$ is bounded by 1.5. By summing them up, we have

$$\frac{\lambda_{\max}\left(\nabla \mathbf{f}'(x_0+u_1) - \nabla \mathbf{f}'(x_0+u_2)\right)}{\|u_1-u_2\|_2} \le 1.25\alpha L_{\mathbf{f}} \le \beta_{\mathbf{f}}/L_{\mathbf{f}} \cdot L_{\mathbf{f}} = \beta_{\mathbf{f}},$$

i.e., $\mathbf{f}'$ is β-smooth.

Proof of Fact 2.2.

$$\begin{aligned}\lambda_{\max}\left(\frac{uu^\top}{\|u\|_2} - \frac{vv^\top}{\|v\|_2}\right) &= \max_{\|w\|_2=1} w^\top\left(\frac{uu^\top}{\|u\|_2} - \frac{vv^\top}{\|v\|_2}\right)w = \max_{\|w\|_2=1} \frac{\|u^\top w\|_2^2}{\|u\|_2} - \frac{\|v^\top w\|_2^2}{\|v\|_2} \\ &= \max_{\|w\|_2=1} \|u\|\cos^2\langle u, w\rangle - \|v\|\cos^2\langle v, w\rangle.\end{aligned} \tag{35}$$

From geometry, we know that the $\cos\langle u, w\rangle$ of a unit vector w lying outside the place P_{uv} equals to $\|w_{uv}\|_2\cos\langle w_{uv}, u\rangle$, where w_{uv} is its projection onto plane P_{uv}, having length $\|w_{uv}\|_2 \le 1$. Therefore, we only need to consider all vectors with length smaller or equal to 1 lying on the plane P_{uv} (i.e., the projection of any unit vector w onto the plane P_{uv}), i.e.,

$$\text{Equation (35)} = \max_{\substack{\|w\|_2\le 1\\ w\in P_{uv}}} \|w\|^2\left(\|u\|\cos^2\langle u, w\rangle - \|v\|\cos^2\langle v, w\rangle\right) = \max_{\substack{\|w\|_2=1\\ w\in P_{uv}}} \left(\|u\|\cos^2\langle u, w\rangle - \|v\|\cos^2\langle v, w\rangle\right).$$

Let θ be the angle between u and v, β be the angle between u and w, then the angle between v and w is $\beta-\theta$. Written as the optimization over β, we have

$$\begin{aligned}\text{Equation (35)} &= \max_{\beta} \|u\|\cos^2\beta - \|v\|\cos^2(\beta-\theta) \\ &= \max_{\beta} \frac{1}{2}\left(\|u\| - \|v\|\right) + \frac{1}{2}\left(\|u\|\cos 2\beta - \|v\|\cos 2(\beta-\theta)\right) \\ &= \frac{1}{2}\left(\|u\| - \|v\|\right) + \frac{1}{2}\left(\max_{\beta}\|u\|\cos\beta - \|v\|\cos(\beta-2\theta)\right).\end{aligned}$$

From geometry, we know for any β, $\|u\|\cos\beta - \|v\|\cos(\beta-2\theta) \le 2\|u-v\|$. Furthermore, $\|u\| - \|v\| \le \|u-v\|$. Thus, Equation (35) $\le 1.5\|u-v\|$. □

Given Fact 2.2, as shown before, $\mathbf{f}'$ is β-smooth.

To this point, we have proven the three arguments in Fact 2.1 respectively. □

Now we inject $\mathbf{f}$ into the Taylor expansion expression for $S(\mathbf{f}'(x_0+\delta u))$, where u is a unit vector, i.e., $\|u\|_2=1$. Similar as Equations (20) to (22):

$$\begin{aligned}
&S(\mathbf{f}'(x_0+\delta u))\\
=&S\left(\mathbf{f}(x_0)+\delta Ju-\frac{1}{2}\alpha\delta^2 Ju\right)\\
=&S(\mathbf{f}(x_0))+\delta\nabla S^\intercal Ju-\frac{1}{2}\alpha\delta^2\nabla S^\intercal Ju+\frac{1}{2}\theta^2\left(\delta Ju-\frac{1}{2}\alpha\delta^2 Ju\right)^\intercal\mathbf{H}\left(\delta Ju-\frac{1}{2}\alpha\delta^2 Ju\right),
\end{aligned}\tag{36}$$

where $\theta\in[-1,1]$ is depended on S, and $\mathbf{H}$ is the Hessian matrix of S at point x_0. Because $\mathbf{f}(x_0)$ is the boundary-image, we have $S(\mathbf{f}(x_0))=0$. We can also bound the last term from the smoothness assumption on S:

$$\left|\frac{1}{2}\theta^2\left(\delta Ju-\frac{1}{2}\alpha\delta^2 Ju\right)^\intercal\mathbf{H}\left(\delta Ju-\frac{1}{2}\alpha\delta^2 Ju\right)\right|\le\frac{1}{2}\beta_S\delta^2\left\|Ju-\frac{1}{2}\alpha\delta Ju\right\|_2^2\le\frac{1}{2}\beta_S\delta^2\left(1-\frac{1}{2}\alpha\delta\right)^2L_{\mathbf{f}}^2.$$

Define $v:=J^\intercal\nabla S(\mathbf{f}(x_0))/\|J^\intercal\nabla S(\mathbf{f}(x_0))\|_2$. From Equation (36), we get

$$\begin{aligned}
S(\mathbf{f}'(x_0+\delta u))&\ge\delta\left(1-\frac{1}{2}\alpha\delta\right)\langle u,v\rangle\|v\|_2-\frac{1}{2}\beta_S\delta^2\left(1-\frac{1}{2}\alpha\delta\right)^2L_{\mathbf{f}}^2,\\
S(\mathbf{f}'(x_0+\delta u))&\le\delta\left(1-\frac{1}{2}\alpha\delta\right)\langle u,v\rangle\|v\|_2+\frac{1}{2}\beta_S\delta^2\left(1-\frac{1}{2}\alpha\delta\right)^2L_{\mathbf{f}}^2.
\end{aligned}$$

Therefore,

$$|\langle u,v\rangle|\|v\|_2\ge\frac{1}{2}\beta_S\delta\left(1-\frac{1}{2}\alpha\delta\right)L_{\mathbf{f}}^2\implies\operatorname{sgn}(S(\mathbf{f}(x_0+\delta u)))=\operatorname{sgn}(\langle u,v\rangle).$$

Note that $\alpha\in[0,\,0.8\beta_{\mathbf{f}}/L_{\mathbf{f}}]$, and larger α induces smaller RHS. We let $\alpha=0.8\beta_{\mathbf{f}}/L_{\mathbf{f}}$, and get

$$|\langle u,v\rangle|\|v\|_2\ge\frac{1}{2}\delta\beta_SL_{\mathbf{f}}^2-\frac{1}{5}\beta_{\mathbf{f}}\beta_S\delta^2L_{\mathbf{f}}\implies\operatorname{sgn}(S(\mathbf{f}(x_0+\delta u)))=\operatorname{sgn}(\langle u,v\rangle).$$

In other words,

$$\omega:=\frac{1}{2}\delta\beta_SL_{\mathbf{f}}^2-\frac{1}{5}\beta_{\mathbf{f}}\beta_S\delta^2L_{\mathbf{f}}$$

satisfies the condition Equation (32). Following the same proof as in Theorem 1 using ω, we get the desired cosine similarity bound for the projection $\mathbf{f}'$. □

D Implications of Gradient Estimation Analysis

In this section, we provide further discussions on the gradient estimation analysis omitted in Section 4.2 and the supporting theorems.

D.1 Comparison of Different Gradient Estimators

We instantiate the cosine similarity bounds for gradient estimators in HSJA (Chen et al., 2020) and QEBA (Li et al., 2020). Then, we compare these bounds along with the bound for NonLinear-BA. The definitions of these estimators are presented in Appendix A.

HSJA. In HSJA, the projection is an identical function. Therefore, $\|\nabla\mathbf{f}^\intercal\nabla S\|=\|\nabla\mathbf{f}^\intercal\nabla S\|$, and $L_{\mathbf{f}}=1$, $\beta_{\mathbf{f}}=0$. We apply Theorem 1 and yield the following cosine similarity bound.

Corollary 3 (Bound for HSJA Gradient Estimator). *Let x_0 be a boundary-image, i.e., $S(x_0) = 0$. The difference function S satisfies the assumptions in Section 4.1. Using HSJA gradient estimator as in Equation (13), over the randomness of the sampling of orthogonal basis subset $u_1, u_2, \ldots, u_B$ for $\mathbb{R}^m$ space, the expectation of cosine similarity between $\widetilde{\nabla S}(x_0)$ ($\widetilde{\nabla S}$ for short) and $\nabla S(x_0)$ (∇S for short) satisfies*

$$\left(2\left(1-\frac{\omega^2}{\|\nabla S\|_2^2}\right)^{\frac{m-1}{2}}-1\right)\sqrt{\frac{B}{m}}c_m \le \mathbb{E}\cos\langle\widetilde{\nabla S}, \nabla S\rangle \le \sqrt{\frac{B}{m}}c_m,$$

where $\omega = \frac{1}{2}\delta\beta_S$, and the $c_m \in (2/\pi, 1)$ is a constant depended on m.

Remark. In the corollary, we can see that without subspace projection, all terms are directly related to the dimensionality of the input space, m.

QEBA. In QEBA, the projection is a random orthogonal transformation denoted by the matrix $\mathbf{W}$. Similarly, we yield the following bound.

Corollary 4 (Bound for QEBA Gradient Estimator). *Let x_0 be a boundary-image, i.e., $S(x_0) = 0$. The difference function S satisfies the assumptions in Section 4.1. Using QEBA gradient estimator as in Equation (14), over the randomness of the sampling of orthogonal basis subset $u_1, u_2, \ldots, u_B$ for $\mathbb{R}^n$ space, the expectation of cosine similarity between $\widetilde{\nabla S}(x_0)$ ($\widetilde{\nabla S}$ for short) and $\nabla S(x_0)$ (∇S for short) satisfies*

$$\left(2\left(1-\frac{\omega^2}{\|\mathbf{W}^\intercal\nabla S\|_2^2}\right)^{\frac{n-1}{2}}-1\right)\frac{\|\mathbf{W}^\intercal\nabla S\|_2}{\|\nabla S\|_2}\sqrt{\frac{B}{n}}c_n \le \mathbb{E}\cos\langle\widetilde{\nabla S}, \nabla S\rangle \le \frac{\|\mathbf{W}^\intercal\nabla S\|_2}{\|\nabla S\|_2}\sqrt{\frac{B}{n}}c_n,$$

where $\omega = \frac{1}{2}\delta\beta_S$, and the $c_n \in (2/\pi, 1)$ is a constant depended on m.

Li et al. (2020) present a similar but slightly tighter cosine similarity bound which replaces $\|\mathbf{W}^\intercal\nabla S\|_2$ by $\|\nabla S\|_2$ leveraging the fact that the projection $\mathbf{W}$ is random.

Comparison between HSJA and QEBA. In QEBA, when $\mathbf{W}$ contains a base vector which aligns well with ∇S, i.e., there exists $i \in [n]$ such that $|\cos\langle\mathbf{W}_{:,i}, \nabla S\rangle|$ is close to 1, then $\|\mathbf{W}^\intercal\nabla S\|_2 \approx \|\nabla S\|_2$. Heuristics are used in QEBA to increase the alignment between basis and the vector ∇S. When the alignment is good, the bound in Corollary 4 differs from that in Corollary 3 only in that m is replaced by n. Given that n is the dimension of subspace which is usually much smaller than m, we know

$$\left(1-\frac{\omega^2}{\|\mathbf{W}^\intercal\nabla S\|_2^2}\right)^{\frac{n-1}{2}} \gg \left(1-\frac{\omega^2}{\|\nabla S\|_2^2}\right)^{\frac{m-1}{2}} \text{ and } \sqrt{\frac{B}{n}} \gg \sqrt{\frac{B}{m}}.$$

As a result, when B is the same, both the lower bound and upper bound in QEBA outperform those of HSJA significantly; and to achieve the same cosine similarity, QEBA requires much fewer queries than HSJA.

NonLinear-BA. Our proposed NonLinear-BA enables the use of nonlinear projection $\mathbf{f}$. As shown by Theorem 1, due to the nonlinearity, the cosine similarity lower bound of nonlinear projection is worse than the linear counterpart (QEBA) due to the additional terms in ω. However, Theorem 2, when compared with linear projection bound in Section 4.2, implies the existence of better nonlinear projection. The existence is proved by a specific construction of a 'good' nonlinear projection which provides higher cosine similarity. Here, we present another 'good' nonlinear projection, to show that such nonlinear projection is not rare or specific.

Theorem 3 (Existence of Better Nonlinear Projection, Part II). *Let $\mathbf{f}(x_0)$ be a boundary-image, i.e., $S(\mathbf{f}(x_0)) = 0$. The projection $\mathbf{f}$ is locally linear around x_0 with radius δ. $L_\mathbf{f} := \lambda_{\max}(\nabla\mathbf{f}(x_0))$, $l_\mathbf{f} := \lambda_{\min}(\nabla\mathbf{f}(x_0))$. The difference function S satisfies the assumptions in Section 4.1.*

There exists a nonlinear projection $\mathbf{f}'$ satisfying the assumptions in Section 4.1, with $\mathbf{f}'(x_0) = \mathbf{f}(x_0)$ and $\nabla\mathbf{f}'(x_0) = \nabla\mathbf{f}(x_0)$, such that over the randomness of the sampling of orthogonal basis subset $u_1, u_2, \ldots, u_B$ for $\mathbb{R}^n$ space, the expectation of cosine similarity between $\widetilde{\nabla S}(\mathbf{f}'(x_0))$ ($\widetilde{\nabla S}$ for short) and $\nabla S(\mathbf{f}'(x_0))$ (∇S for short) satisfies Equation (8) with

$$\omega < \frac{1}{2}\delta\beta_S L_\mathbf{f}^2. \tag{37}$$

We assume $\omega \le \|\nabla\mathbf{f}^\intercal\nabla S\|_2$, and $\delta < L_S/(\beta_S L_\mathbf{f})$. The $c_n \in (2/\pi, 1)$ is a constant depended on n.

Proof of Theorem 3. Let $J := \nabla \mathbf{f}(x_0)$, and $v := J^\intercal \nabla S(\mathbf{f}(x_0))/\|J^\intercal \nabla S(\mathbf{f}(x_0))\|_2$. For arbitrary $u \in \mathbb{R}^n$, we define $\mathbf{f}'(x_0 + u)$ as such:

$$\mathbf{f}'(x_0 + u) = \mathbf{f}(x_0) + J \cdot u + \frac{1}{2}\text{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2 k \nabla S, \tag{38}$$

where $k \in [0,\, \beta_{\mathbf{f}}/L_S]$ is an adjustable parameter.

Fact 3.1. *The $\mathbf{f}'$ defined as Equation (38) has gradient J at point x_0 and is $\beta_{\mathbf{f}}$-smooth.*

Proof of Fact 3.1. Since

$$\lim_{u\to 0} \frac{\left\|\frac{1}{2}\langle u,\, v\rangle^2 k \nabla S\right\|_2}{\|u\|_2} \le \lim_{u\to 0} \frac{1}{2}|\langle u,\, v\rangle| k \|\nabla S\|_2 \le \lim_{u\to 0} \frac{1}{2}\frac{\beta_{\mathbf{f}}}{L_S} L_S \|u\|_2 = 0,$$

we have $\mathbf{f}'(x_0 + u) = \mathbf{f}(x_0) + J \cdot u + o(u)$ so $\nabla \mathbf{f}'(x_0) := \nabla \mathbf{f}(x_0) = J$.

We compute $\nabla \mathbf{f}'$ for arbitrary point, since

$$\frac{\partial \mathbf{f}'(x_0 + u)_i}{\partial u_j} = J_{ij} + \text{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle v_j k \nabla S_i,$$

we know $\nabla \mathbf{f}'(x_0 + u) = J + \text{sgn}(\langle u,\, v\rangle) k \langle u,\, v\rangle \nabla S v^\intercal$. Consider arbitrary u_1, u_2:

- If $\langle u_1,\, v\rangle \cdot \langle u_2,\, v\rangle \ge 0$, $\nabla \mathbf{f}'(x_0 + u_1) - \nabla \mathbf{f}'(x_0 + u_2) = \text{sgn}(\langle u_1,\, v\rangle) k \langle u_1 - u_2,\, v\rangle \nabla S v^\intercal$. Therefore,

$$\frac{\lambda_{\max}(\nabla \mathbf{f}'(x_0 + u_1) - \nabla \mathbf{f}'(x_0 + u_2))}{\|u_1 - u_2\|_2} \le \frac{|\langle u_1 - u_2,\, v\rangle|}{\|u_1 - u_2\|_2} k \lambda_{\max}(\nabla S v^\intercal) \le k L_s \le \beta_{\mathbf{f}}.$$

- If $\langle u_1,\, v\rangle \cdot \langle u_2,\, v\rangle < 0$, without loss of generality, let $\langle u_1,\, v\rangle > 0$ and $\langle u_2,\, v\rangle < 0$. Therefore,

$$\nabla \mathbf{f}'(x_0 + u_1) - \nabla \mathbf{f}'(x_0 + u_2) = k \langle u_1 + u_2,\, v\rangle \nabla S v^\intercal.$$

  Since $\langle u_1,\, v\rangle > 0$ and $\langle u_2,\, v\rangle < 0$, $|\langle u_1 + u_2,\, v\rangle| \le |\langle u_1 - u_2,\, v\rangle|$. Thus,

$$\frac{\lambda_{\max}(\nabla \mathbf{f}'(x_0 + u_1) - \nabla \mathbf{f}'(x_0 + u_2))}{\|u_1 - u_2\|_2} \le \frac{|\langle u_1 + u_2,\, v\rangle|}{\|u_1 - u_2\|_2} k \lambda_{\max}(\nabla S v^\intercal) \le \frac{|\langle u_1 - u_2,\, v\rangle|}{\|u_1 - u_2\|_2} k \lambda_{\max}(\nabla S v^\intercal) \le \beta_{\mathbf{f}}.$$

According to the smoothness definition, $\mathbf{f}'$ is $\beta_{\mathbf{f}}$-smooth. □

Now let us inject $\mathbf{f}'$ into the Taylor expansion expression for $S\left(\mathbf{f}'(x_0 + \delta u)\right)$ in a similar way as Equations (20) to (22), where u is a unit vector, i.e., $\|u\|_2 = 1$:

$$\begin{aligned}
&S(\mathbf{f}'(x_0 + \delta u)) \\
=&S\left(\mathbf{f}(x_0) + \delta J u + \frac{1}{2}\text{sgn}\left(\langle u,\, v\rangle\right)\langle u,\, v\rangle^2 \delta^2 k \nabla S\right) \\
=&S(\mathbf{f}(x_0)) + \delta \nabla S^\intercal J u + \frac{1}{2}\text{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2 \delta^2 k \|\nabla S\|^2 + \\
&\frac{1}{2}\theta^2 \left(\delta J u + \frac{1}{2}\text{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2 \delta^2 k \nabla S\right)^\intercal \mathbf{H} \left(\delta J u + \frac{1}{2}\text{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2 \delta^2 k \nabla S\right),
\end{aligned} \tag{39}$$

where $\theta \in [-1,\, 1]$ is depended on S, and $\mathbf{H}$ is the Hessian matrix of S at point x_0. Because x_0 is the boundary point, we have $S(\mathbf{f}(x_0)) = 0$.

We can bound the last term as such:

$$\begin{aligned}
&\left|\frac{1}{2}\theta^2 \left(\delta J u + \frac{1}{2}\text{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2 \delta^2 k \nabla S\right)^\intercal \mathbf{H} \left(\delta J u + \frac{1}{2}\text{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2 \delta^2 k \nabla S\right)\right| \\
\le&\frac{1}{2}\beta_S \left(\delta L_{\mathbf{f}} + \frac{1}{2}\langle u,\, v\rangle^2 \delta^2 k L_S\right)^2 = \frac{1}{2}\beta_S \delta^2 \left(L_{\mathbf{f}} + \frac{1}{2}\langle u,\, v\rangle^2 \delta k L_S\right)^2.
\end{aligned}$$

When $\langle u, v\rangle > 0$, from Equation (39), we get

$$\begin{aligned} S(\mathbf{f}'(x_0+\delta u)) &\geq \delta \nabla S^\top J u + \frac{1}{2}\langle u, v\rangle^2 \delta^2 k L_S^2 - \frac{1}{2}\beta_S \delta^2 \left(L_{\mathbf{f}} + \frac{1}{2}\langle u, v\rangle^2 \delta k L_S\right)^2 \\ &= \delta \langle u, v\rangle \|v\|_2 + \frac{1}{2}\langle u, v\rangle^2 \delta^2 k L_S^2 - \frac{1}{2}\beta_S \delta^2 \left(L_{\mathbf{f}} + \frac{1}{2}\langle u, v\rangle^2 \delta k L_S\right)^2, \end{aligned}$$

and similarly, when $\langle u, v\rangle < 0$, we get

$$S(\mathbf{f}'(x_0+\delta u)) \leq \delta \langle u, v\rangle \|v\|_2 - \frac{1}{2}\langle u, v\rangle^2 \delta^2 k L_S^2 + \frac{1}{2}\beta_S \delta^2 \left(L_{\mathbf{f}} + \frac{1}{2}\langle u, v\rangle^2 \delta k L_S\right)^2.$$

Therefore,

$$|\langle u, v\rangle| \|v\|_2 \geq -\frac{1}{2}\langle u, v\rangle^2 \delta k L_S^2 + \frac{1}{2}\beta_S \delta \left(L_{\mathbf{f}} + \frac{1}{2}\langle u, v\rangle^2 \delta k L_S\right)^2 \implies \operatorname{sgn}(S(\mathbf{f}(x_0+\delta u))) = \operatorname{sgn}(\langle u, v\rangle). \tag{40}$$

Denote $h\left(k; \langle u, v\rangle\right)$ to the RHS:

$$h\left(k; \langle u, v\rangle\right) := -\frac{1}{2}\langle u, v\rangle^2 \delta k L_S^2 + \frac{1}{2}\beta_S \delta \left(L_{\mathbf{f}} + \frac{1}{2}\langle u, v\rangle^2 \delta k L_S\right)^2.$$

When $k = 0$,

$$h(k; \langle u, v\rangle) = \frac{1}{2}\beta_S \delta L_{\mathbf{f}}^2, \quad \frac{\partial h(k; \langle u, v\rangle)}{\partial k}\Big|_{k=0} = -\frac{1}{2}\langle u, v\rangle^2 \delta L_S^2 + \frac{1}{2}\langle u, v\rangle^2 \delta^2 L_S L_{\mathbf{f}} \beta_S = \frac{1}{2}\langle u, v\rangle^2 \delta L_S (\delta L_{\mathbf{f}} \beta_S - L_S).$$

Therefore, when $|\langle u, v\rangle| \geq \epsilon' > 0$,

$$\frac{\partial h(k; \langle u, v\rangle)}{\partial k}\Big|_{k=0} \leq \frac{1}{2}\epsilon'^2 \delta L_S (\delta L_{\mathbf{f}} \beta_S - L_S) < 0,$$

and thus there exists small $\epsilon > 0, \eta > 0$, when $k = \epsilon$ and $|\langle u, v\rangle| \geq \epsilon'$, $h(k; \langle u, v\rangle) < \frac{1}{2}\beta_S \delta L_{\mathbf{f}}^2 - \eta$.

As a result, from Equation (40), we know that when $|\langle u, v\rangle| \geq \epsilon'$, if $|\langle u, v\rangle| \|v\|_2 \geq \frac{1}{2}\beta_S \delta L_{\mathbf{f}}^2 - \eta$, $\operatorname{sgn}(S(\mathbf{f}(x_0+\delta u))) = \operatorname{sgn}(\langle u, v\rangle)$. In other words, let

$$\omega' := \frac{1}{2}\beta_S \delta L_{\mathbf{f}}^2 - \eta,$$

then this ω' satisfies the condition in Equation (32).

Following the same proof as in Theorem 1 using ω', we get the desired lower bound. □

Theorems 2 and 3 present two constructions of nonlinear projection $\mathbf{f}'$ which is better than the corresponding linear projection, and they also provide a checkable condition to examine whether the given nonlinear projection is 'good' in terms of outperforming corresponding linear projection. Since the two constructed projections are quite different from each other, we conjecture that such nonlinear projection is not rare or specific. Even though there is no theoretically guaranteed approach for searching such 'good' nonlinear projection, in experiments, we show that AE, VAE, or GAN are possible choices that usually work well in practice.

D.2 Improve The Gradient Estimation

In Theorems 1 and 2, we relate the cosine similarity bound to variables characterizing the projection $\mathbf{f}$ such as $\nabla \mathbf{f}$, $L_{\mathbf{f}}$, $\beta_{\mathbf{f}}$. By examining the change tendency of the bound with respect to these variables, we learn ways for improving the gradient estimation in terms of improving its cosine similarity with the true gradient.

- Increase the alignment between ∇S and $\nabla \mathbf{f}$:
  The term $\|\nabla \mathbf{f}^\intercal \nabla S\|_2 / \|\nabla S\|_2$ reveals that, we should increase the alignment between ∇S and $\nabla \mathbf{f}$ to improve the cosine similarity. When L_S and $L_{\mathbf{f}}$ are fixed, if they are more aligned, $\|\nabla S^\intercal \nabla \mathbf{f}\|_2^2$ is larger so that the lower bound becomes larger. It implies that the mapping $\mathbf{f}$ should reflect the main components of ∇S as much as possible. Similar conclusion is shown for QEBA in Appendix D.1.

- Reduce the subspace dimension n and increase number of queries B:
  When ∇S and $\nabla \mathbf{f}$ can be aligned, it is better to keep the subspace dimension of $\mathbf{f}$, n, be small. The reason is analyzed in Appendix D.1 when comparing HSJA and QEBA. At the same time, increasing number of queries B is also helpful, according to the query complexity analysis in Section 4.2.

- If we can find good nonlinear projection, decrease the smoothness; otherwise, increase the smoothness and decrease step size δ:
  If the a good nonlinear projection can be found, we consider the bound in Theorem 2, which shows the outcome of a good nonlinear projection. Learned from its ω in Equation (10), increasing $\beta_{\mathbf{f}}$, i.e., decreasing the smoothness, could reduce ω and hence improve cosine similarity bound. If the good nonlienar projection cannot be found, we consider the bound in Theorem 1, which bounds the general projections. To reduce ω in this case which is defined by Definition 5, we need to reduce $\beta_{\mathbf{f}}$, i.e., increase the smoothness, and reduce the step size δ. We remark that the choice of step size δ needs to consider many other factors as Chen et al. (2020) outlined.

E Target Models

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

E.1 Implementation Details

Offline Models. Following Li et al. (2020), we use models based on a pretrained ResNet-18 model as the target models. 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 trained exactly on ImageNet. The model is loaded with PyTorch command '`torchvision.models.resnet18(pretrained=True)`' following the documentation (PyTorch, 2021).

For CelebA, the target model is finetuned to do binary classification on image attributes. Among the 40 binary attributes associated with each image, we sort the attributes according to how balance the numbers of positive and negative samples are. The more balanced the dataset is, it is better for the classification model training. The top-5 balanced attributes are 'Attractive', 'Mouth_Slightly_Open', 'Smiling', 'Wearing_Lipstick', 'High_Cheekbones'. Though the 'Attractive' attribute is the most balanced one, it is more objective than subjective, thus we instead use the second attribute 'Mouth_Slightly_Open'.

For MNIST and Cifar10 datasets, we first do linear interpolation and get 224×224 images, then the target model is finetuned to do 10-way classification. One reason for doing interpolation is that our proposed method reduces query complexity when the original data dimension is high so it is more illustrative after upsampling. The linear interpolation step also makes image sizes consistent among all the tasks and experiments.

We report the benign target model performance for the four datasets in Table 1.

Commercial Online API. Among all the APIs provided by the Face++ platform (MEGVII, 2021c), we use the 'Compare' API (MEGVII, 2021a) which takes two images as input and returns a confidence score of whether they are the same person if there are faces in the two images. This is also consistent with the same experiment in QEBA (Li et al., 2020). 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, 2021b). We set the similarity threshold as 50% in the experiments following QEBA (Li et al., 2020): when the confidence score is equal to or larger than 50%, we consider the two faces to belong to the 'same person', vice versa.

Table 1: The benign model accuracies of the target model (ResNet-18).

Dataset	CelebA	CIFAR10	MNIST
Benign Accuracy	0.9417	0.8796	0.9938

For source-target images that are from two different persons, the goal of the attack is to get an adv-image that looks like the target-image (has low mean squared error distance between the adv-image and target-image), but is predicted as 'same person' with the source-image. We randomly sample source-target image pairs from the CelebA dataset that are predicted as different persons by the 'Compare' API. Then we apply the NonLinear-BA pipeline with various nonlinear projection models for comparison.

E.2 Model Performance of Target Models

The benign accuracies of the target model ResNet-18 on the datasets are shown in Table 1.

F Nonlinear Projection Based Gradient Estimator

In this section, we introduce the details of nonlinear projection models including the model structure, training procedure. We also introduce how the projection models are used in the NonLinear-BA process including the gradient estimation and attack implementation details.

F.1 Generative Model Structure

AE and VAE. We borrow the idea from U-Net (Ronneberger et al., 2015) which has the structure of an information contraction path and an expanding path, with a small latent representation in the middle.

Define 2D convolution layer Conv2d(in_channels, out_channels, kernel_size, padding_size).

Define the DoubleConv(in_channels, out_channels) layer as composed of 6 layers: a 2D convolution layer Conv2d(in_channels, out_channels) with kernel size 3 and padding size 1; a 2D batch normalization layer BatchNorm2d(out_channels); a ReLU layer; another 2D convolution layer Conv2d(out_channels, out_channels) with kernel size 3 and padding size 1; a 2D batch normalization layer BatchNorm2d(out_channels); and a ReLU layer.

Define the Down(in_channels, out_channels) layer with two components: a max-pooling layer MaxPool2d with kernel size 2; a DoubleConv(in_channels, out_channels) as defined above.

Likewise, the Up(in_channels, out_channels) is defined with two components: a up-scaling layer and a DoubleConv(in_channels, out_channels) as defined above.

The AE and VAE models have similar structures except for the fact that the encoder part of VAE has two output layers to produce the mean and standard deviation vectors, and the AE only has one. The detailed network structures are shown in Table 2. The n_channels is the number of image channels determined by the image dataset. For the grey-scale images in MNIST, there is only 1 channel; for the other three colored datasets (ImageNet, CelebA and CIFAR10), there are RGB channels so n_channels is 3. The latent dimension of the two models is $48 \times 14 \times 14 = 9408$.

GAN. Define ConvBlock(in_channels, out_channels, n_kernel, n_stride, n_pad, transpose, leaky) with three layers: a 2D convolution layer; a batch normalization layer; and a nonlinear ReLU layer.

For ImageNet and CelebA, the detailed model network structures for the generator and discriminator are listed in Table 3 and Table 4.

For CIFAR10 and MNIST, we use DCGAN (Radford et al., 2015) structure with pretrained weights from `https://github.com/csinva/gan-vae-pretrained-pytorch/` and add a linear interpolation layer to resize the generated images to size 224×224.

Table 2: The detailed network structure for AE and VAE models.

Layer Name	AE	Layer Name	VAE
InConv	DoubleConv(n_channels, 24)	InConv	DoubleConv(n_channels, 24)
Down1	Down(24, 24)	Down1	Down(24, 24)
Down2	Down(24, 48)	Down2	Down(24, 48)
Down3	Down(48, 48)	Down3	Down(48, 48)
Down4	Down(48, 48)	DownMu	Down(48, 48)
-	-	DownStd	Down(48, 48)
Up1	Up(48, 48)	Up1	Up(48, 48)
Up2	Up(48, 48)	Up2	Up(48, 48)
Up3	Up(48, 24)	Up3	Up(48, 24)
Up4	Up(24, 24)	Up4	Up(24, 24)
OutConv	Conv2d(24, n_channels, 1, 0)	OutConv	Conv2d(24, n_channels, 1, 0)

Table 3: The detailed model structure for generator in GAN.

Generator
ConvBlock(z_latent, 128, 4, 1, 0, transpose=True, leaky=True)
ConvBlock(128, 64, 3, 2, 1, transpose=True, leaky=False)
ConvBlock(64, 64, 4, 2, 1, transpose=True, leaky=False)
ConvBlock(64, 32, 4, 2, 1, transpose=True, leaky=False)
ConvBlock(32, 32, 4, 2, 1, transpose=True, leaky=False)
ConvBlock(32, 16, 4, 2, 1, transpose=True, leaky=False)
nn.ConvTranspose2d(16, n_channels, 4, 2, 1, bias=False)
nn.Tanh()

F.2 Estimator Training Procedure

The attacker first trains a set of reference models that are generally assumed to have different structures compared to 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, there are five reference models with different backbones compared with the target model, while the implementation and training details are similar to the target model in Section E.1. The benign test accuracy results for CelebA, Cifar10 and MNIST datasets are shown in Table 5, Table 6 and Table 7 respectively. 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, since the number of images is large, the gradient dataset generated by reference models is also too large to be handled in our GPU memory especially when we evaluate the baseline method QEBA-I (Li et al., 2020) since it requires approximate PCA. Thus, 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 CIFAR10 and MNIST, there are fewer images and the machine can handle them properly, so we use the whole gradient dataset generated with $250,000$ gradient images for CIFAR10 ($50,000$ per reference model) and $300,000$ ($60,000$ per reference model) gradient images for MNIST.

The generative models are trained on the gradient images of the corresponding dataset generated as above.

F.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 performances in terms of prediction accuracy for CelebA, Cifar10 and MNIST datasets are shown in Table 5, Table 6, and Table 7. The model performances are comparable to those of the target models.

Table 4: The detailed model structure for discriminator in GAN.

Discriminator
nn.Conv2d(n_channels, 16, 4, 2, 1, bias=False)
nn.LeakyReLU(0.2, inplace=True)
ConvBlock(16, 32, 4, 2, 1, transpose=False, leaky=True)
ConvBlock(32, 32, 4, 2, 1, transpose=False, leaky=True)
ConvBlock(32, 64, 4, 2, 1, transpose=False, leaky=True)
ConvBlock(64, 64, 4, 2, 1, transpose=False, leaky=True)
ConvBlock(64, 128, 3, 2, 1, transpose=False, leaky=True)
nn.Conv2d(128, 1, 4, 1, 0, transpose=False, leaky=True)

Table 5: The benign model accuracies of the reference models for CelebA dataset (attribute: 'mouth_slightly_open').

CelebA	DenseNet-121	ResNet-50	VGG16	GoogleNet	WideResNet
Benign Accuracy	0.9415	0.9410	0.9417	0.9315	0.9416

Table 6: The benign model accuracies of the reference models for Cifar10 dataset (linearly interpolated to size $3 \times 224 \times 224$).

Cifar10	DenseNet-121	ResNet-50	VGG16	GoogleNet	WideResNet
Benign Accuracy	0.9079	0.8722	0.9230	0.9114	0.8568

Table 7: The benign model accuracies of the reference models for MNIST dataset (linearly interpolated to size 224×224).

MNIST	DenseNet-121	ResNet-50	VGG16	GoogleNet	WideResNet
Benign Accuracy	0.9919	0.9916	0.9948	0.9943	0.9938

F.4 Nonlinear Projection Based Gradient Estimation

We provide the pseudo code for the gradient estimation process with the nonlinear projection functions in Algorithm 1.

F.5 Attack Implementation

The goal is to generate an attack image that looks similar as the target-image but is predicted as the label of the source-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.

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 face image dataset CelebA.

Algorithm 1 Nonlinear Projection Based Gradient Estimation

Input: a data point on the decision boundary $\mathbf{x} \in \mathbb{R}^m$, nonlinear projection function $\mathbf{f}$, number of random sampling B, access to query the decision of target model $\phi(\cdot) = \text{sgn}(S(\cdot))$.

Output: the approximated gradient $\widetilde{\nabla S}(x_{adv}^{(t)})$.

1: sample B random Gaussian vectors of the lower dimension: $v_b \in \mathbb{R}^n$.

2: use nonlinear projection function to project the random vectors to the gradient space: $u_b = \mathbf{f}(v_b) \in \mathbb{R}^m$.

3: get query points by adding perturbation vectors to the original point on the decision boundary $x_{adv}^{(t)} + \delta \mathbf{f} v_b$.

4: Monte Carlo approximation for the gradient:

$$\widetilde{\nabla S}(x_{adv}^{(t)}) = \frac{1}{B}\sum_{b=1}^{B} \phi\left(x_{adv}^{(t)} + \delta \mathbf{f}(v_b)\right)\mathbf{f}(v_b) = \frac{1}{B}\sum_{b=1}^{B} \text{sgn}\left(S\left(x_{adv}^{(t)} + \delta \mathbf{f}(v_b)\right)\right)\mathbf{f}(v_b).$$

5: **return** $\widetilde{\nabla S}(x_{adv}^{(t)})$.

Table 8: The mean squared error (MSE) distance thresholds used for four datasets that determine whether the attack is successful.

Dataset	ImageNet	CelebA	MNSIT	CIFAR10
MSE Threshold	1^{-3}	1^{-4}	5^{-3}	1^{-4}

G Quantitative Results

G.1 Attack Success Rate for Offline Models

The 'successful attack' is defined as the adv-image reaching some predefined mean squared error (MSE) distance threshold. Note that because of the varying complexity of tasks and images among different datasets, we set different MSE distance thresholds for different datasets. For example, ImageNet images are the most complicated and the task is most difficult. Thus, we set larger (looser) threshold for it. Specifically, the thresholds are shown in Table 8. The attack success rates on the four datasets are shown in Table 7.

G.2 Proxy for the ω Value

According to the analysis in Section 4.1, smaller ω leads to better gradient estimation. The exact computation of ω requires computing the tight Lipschitz and smoothness constant for both the projection $\mathbf{f}$ and the difference function S, which is challenging. Therefore, we provide a proxy of the ω variable during the training. When estimating the gradient at each boundary-image $x_{adv}^{(t)}$ point with Equation (2), there are some perturbations that contribute negatively in the Monte-Carlo estimation. More formally, a perturbation vector $\mathbf{f}(v_b)$ has a negative contribution to the gradient estimation if

$$\text{sgn}\left(S\left(x_{adv}^{(t)} + \delta \mathbf{f}(v_b)\right)\right) \neq \text{sgn}\left(\cos\left\langle(\widetilde{\nabla S}(x_{adv}^{(t)}), \mathbf{f}(v_b)\right\rangle\right). \tag{41}$$

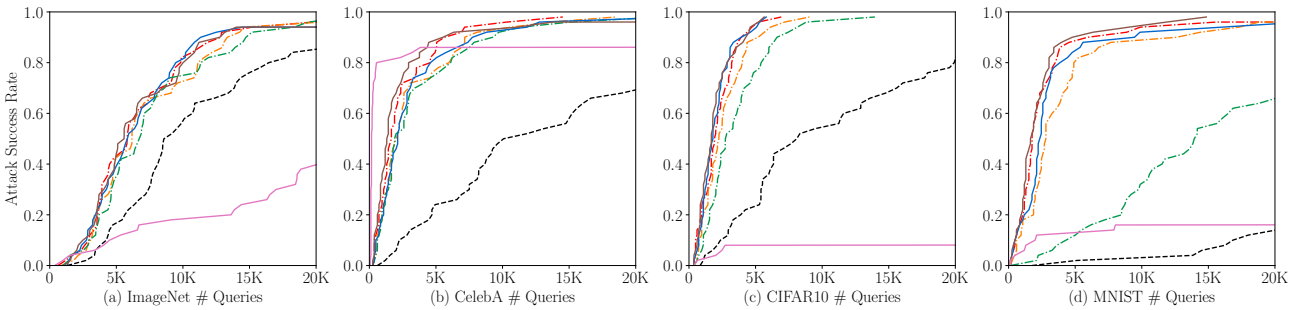

Figure 7: The attack success rate vs query number for four different datasets.

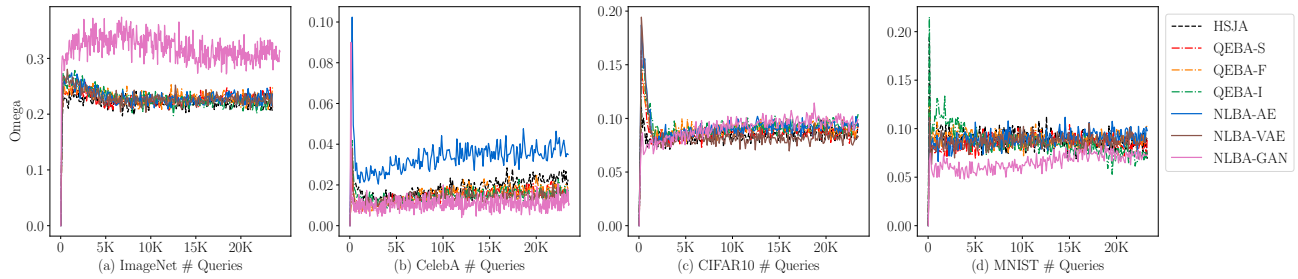

Figure 8: The ω value at different queries for attacks on diverse datasets.

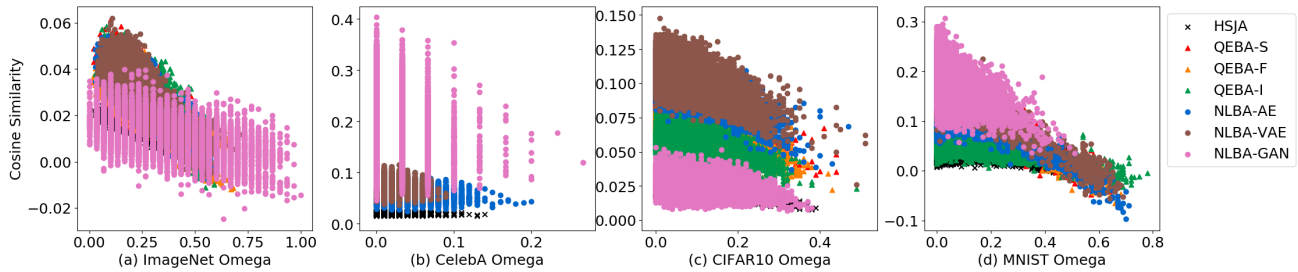

Figure 9: The cosine similarity values at different ω values for attacks on diverse datasets.

In other words, the sign of target model prediction disagrees with the sign of the cosine similarity between the estimated gradient and the perturbation direction. We deem the ratio of samples that satisfy Equation (41) as the proxy of ω. The results are shown in Figure 8.

G.3 Correlation between ω and Cosine Similarity

To verify the correlation between variable ω and the cosine similarity measure as proposed by Equation 11 in Section 4.2, we calculate the two variables during the attack process on different datasets with various projection models and plot them as x and y axis in Figure 9. The cosine similarity values exhibit a descending trend with the increase of the ω values. To further confirm this, we calculate Pearson's correlation score and the results are shown in Figure 10. On ImageNet, CIFAR10, and MNIST datasets, the Pearson's correlation scores are negative with large absolute values, showing the ω and cosine similarity values have a strong negative correlation. On CelebA dataset, the negative correlation between the two variables is less statistically significant.

H Qualitative Results

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

H.1 CelebA Case Study

The whole figure for the case study on CelebA dataset of the attack performance at the early stage of the attack process is shown in Figure 11.

H.2 Offline Models

The goal of the attack is to generate an adv-image that looks like the target-image but has the same label with source-image. We report qualitative results that show how the adv-image changes during the attack process in Figure 12, Figure 13, Figure 14 and Figure 15 for the four datasets respectively. 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 images in the sampled pairs have different ground truth labels (otherwise the attack is trivial). The other five columns each represents the adv-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

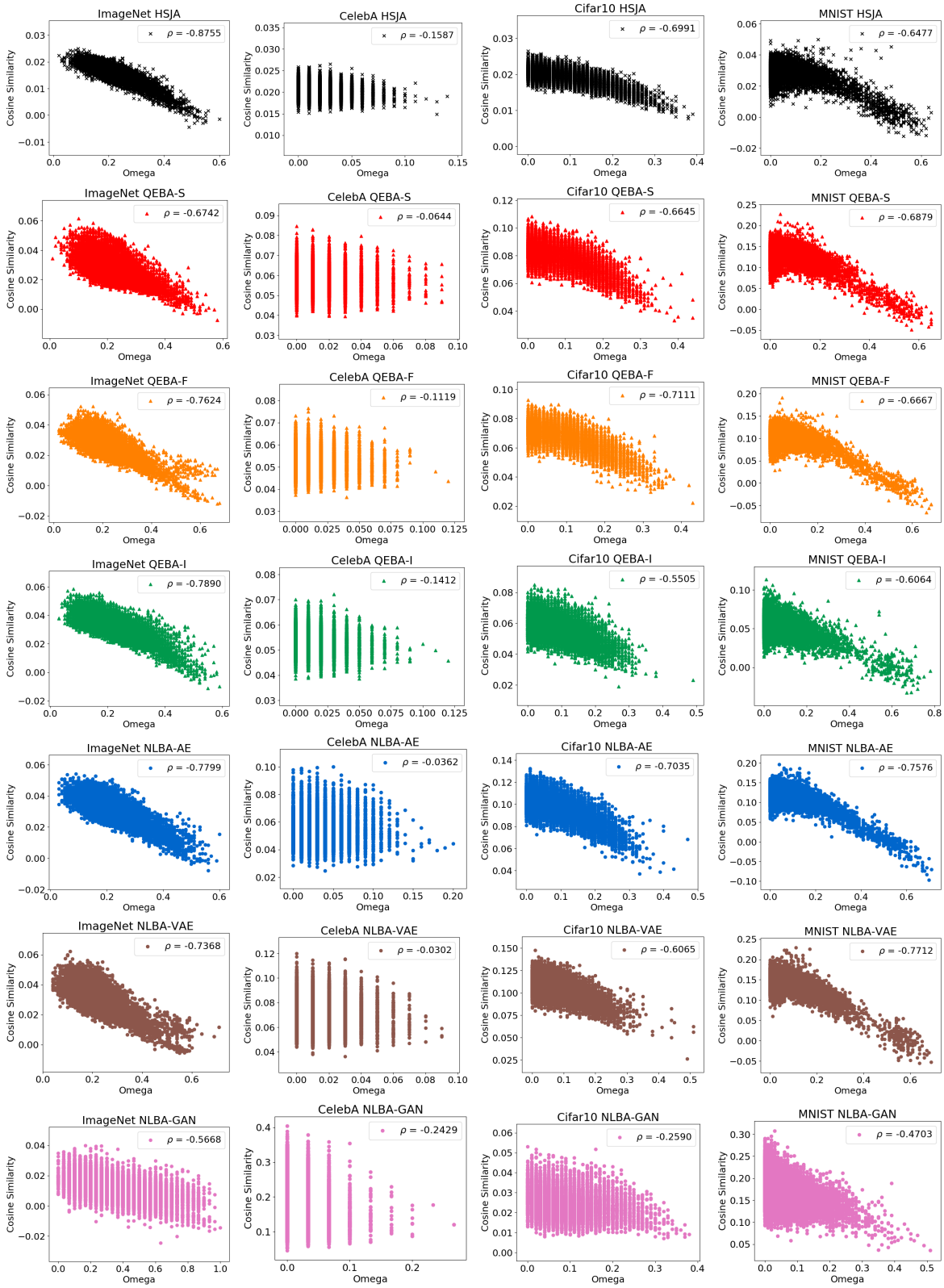

Figure 10: The ω vs cosine similarity values for the 4 datasets and 7 projection methods.

Figure 11: The attack performance of all the NonLinear-BA methods and the baseline methods on one pair of image of the CelebA dataset. The source-image and target-image of this case study are shown in Figure 4. The d in the figure denotes the perturbation magnitude (mean squared error) of the adversarial example with respect to the target-image. The $\#q$ values are the number of queries used at the point for each column.

row represents one method as shown on the right. The d value under each image shows the MSE between the adv-image and the target-image. The smaller d can get, the better the attack is.

H.3 Commercial Online API Attack

As discussed in Section 5, the goal is to generate an adv-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 16. In the figure, the source-image and target-image are shown on the left-most column.

Figure 12: The qualitative case study of attacking ResNet-18 model on ImageNet dataset.

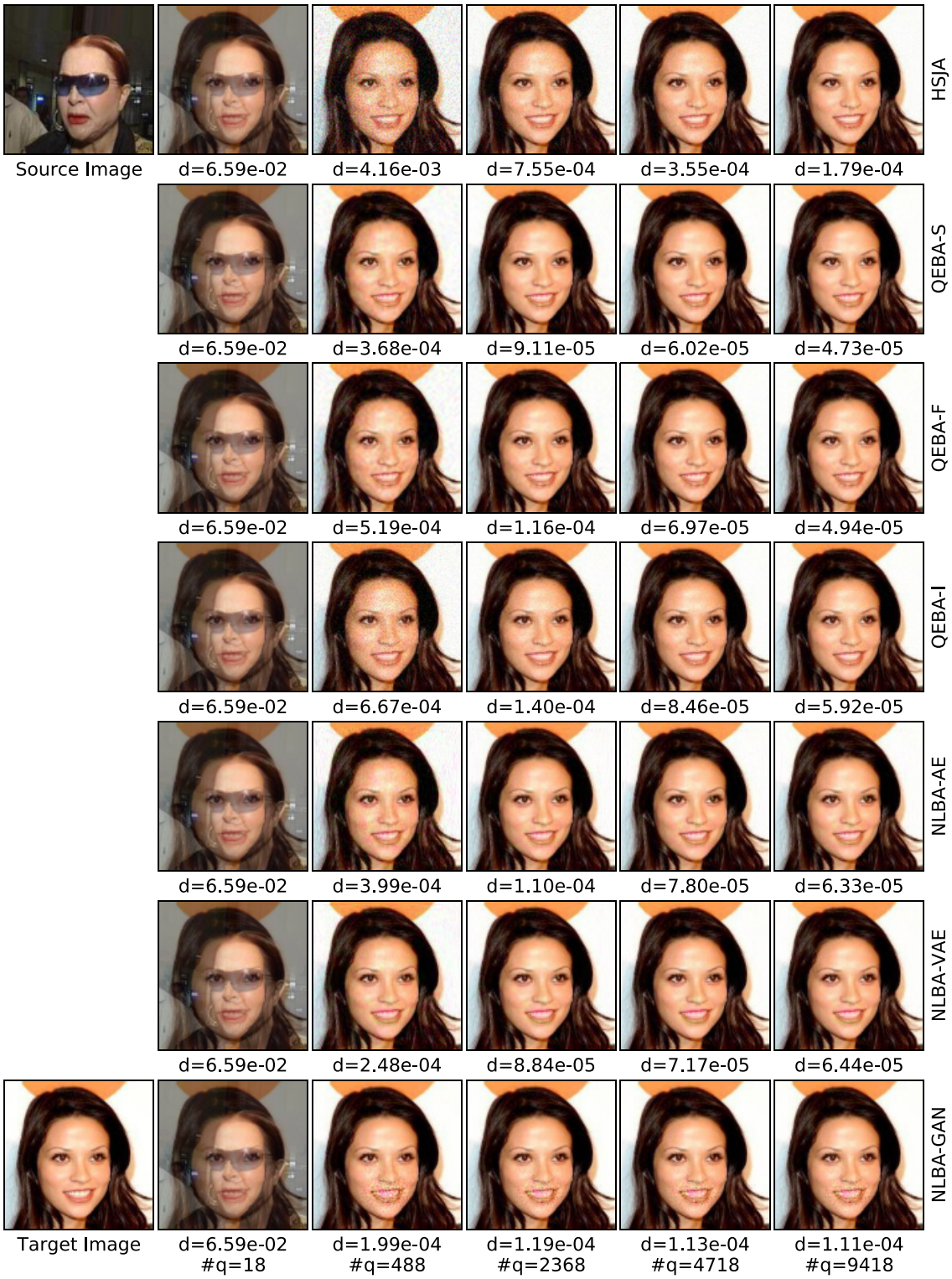

Figure 13: The qualitative case study of attacking ResNet-18 model on CelebA dataset.

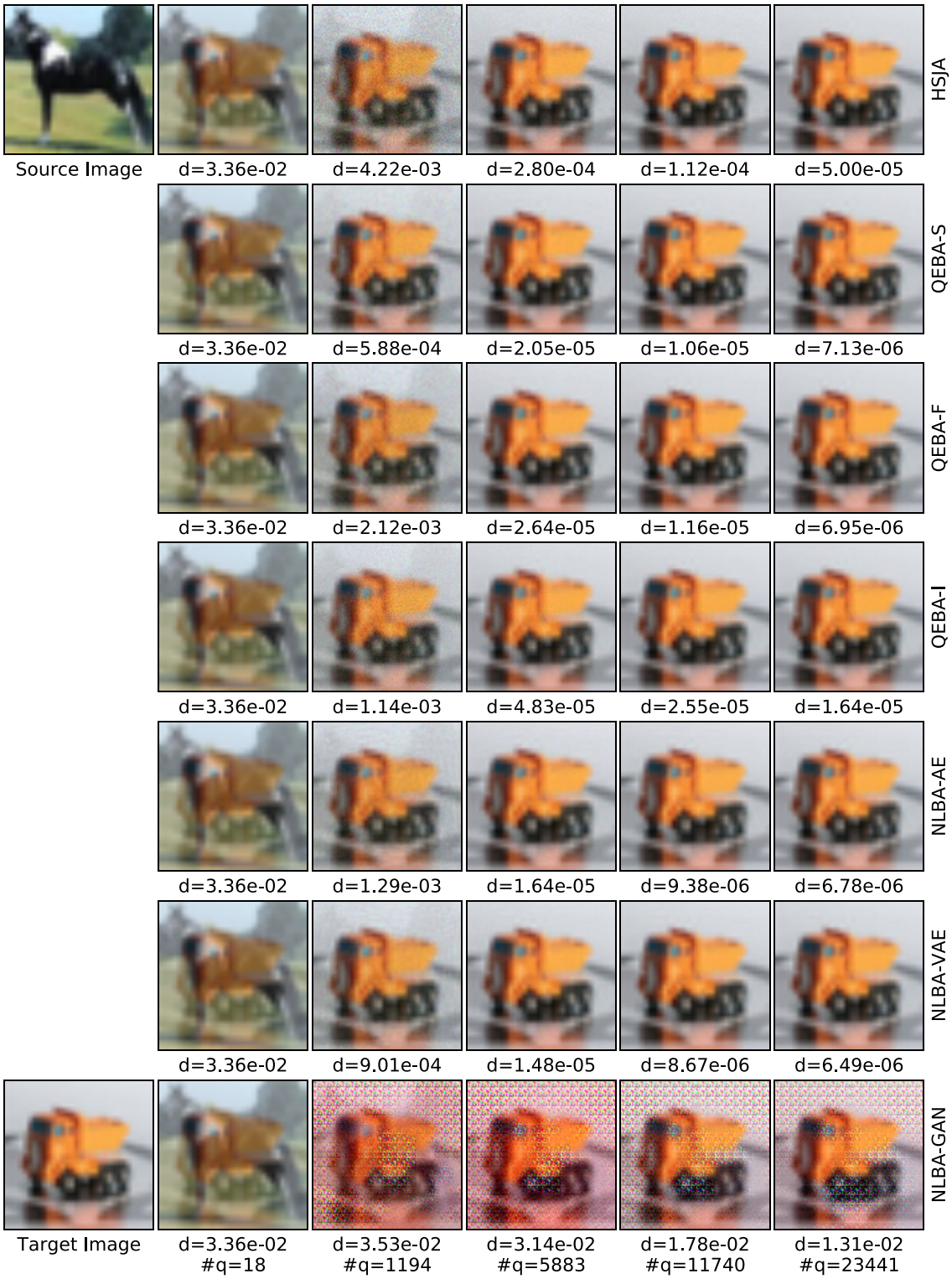

Figure 14: The qualitative case study of attacking ResNet-18 model on CIFAR10 dataset.

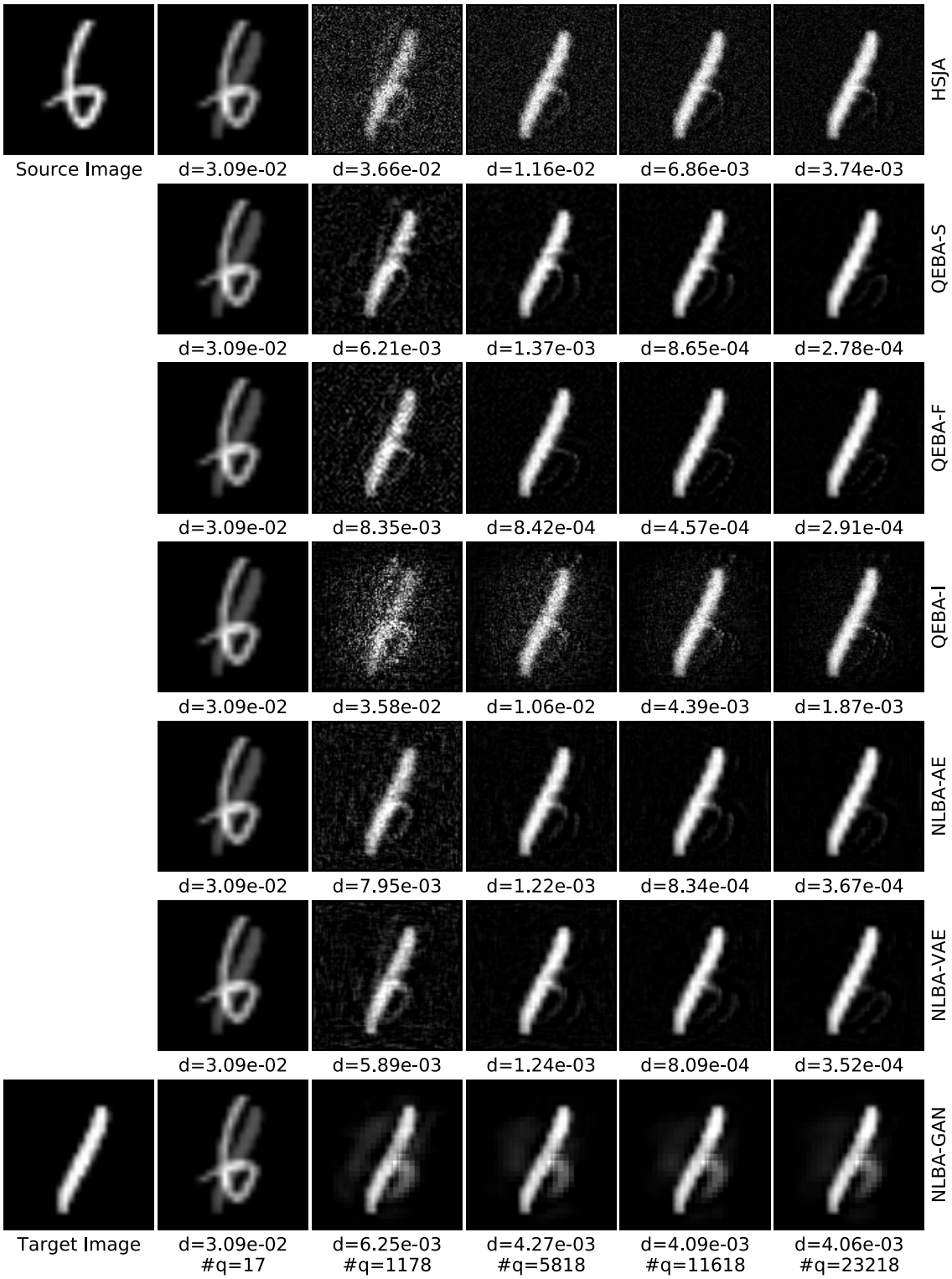

Figure 15: The qualitative case study of attacking ResNet-18 model on MNIST dataset.

Figure 16: A case study of Face++ online API attack process. The source-target image pair is randomly sampled from CelebA dataset (ID: 163922 and 080037).