Nonlinear Gradient Estimation for Query Efficient Blackbox Attack

Huichen Li [*1] **Linyi Li** [*1] **Xiaojun Xu** [1] **Xiaolu Zhang** [2] **Shuang Yang** [2] **Bo Li** [1]

Abstract

Gradient projection and *gradient estimation* have been studied as two distinct topics. We aim to bridge the gap between the two by investigating how gradient can be effectively estimated from a projected low-dimensional space. We provide lower and upper bounds for gradient estimation under both linear and non-linear projections. Moreover, we analyze the query complexity for the projection-based gradient estimation. Built upon our theoretic analysis, we propose a novel query-efficient **Nonlinear** Gradient Projection-based **B**oundary-based Blackbox **A**ttack (**NonLinear-BA**). We show that the boundary blackbox attack with projection-based gradient estimation is able to achieve a much smaller magnitude of perturbation with the same number of queries and a 100% attack success rate. We also evaluate NonLinear-BA against commercial online API MEGVII Face++ and demonstrate high attack performance.

1. Introduction

Gradient estimation and gradient projection have both been extensively studied in machine learning, but largely for different purposes. Gradient estimation is used when gradient-based optimization such as back-propagation is employed but the exact gradients are not directly accessible, for example, in the case of blackbox adversarial attack (Chen et al., 2020; Li et al., 2020). Gradient projection (or sparsification), on the other hand, has also been used to speedup training, for instance, by reducing the complexity of communication and/or storage when performing model update in distributed training (Wangni et al., 2018). In this paper, we aim to bridge the gap between the two and ask the following questions: *Can we estimate gradients from a projected low-dimensional subspace? How do different projections affect the gradient estimation quality?*

Our investigation is motivated in particular by the challenging problem of blackbox adversarial attacks (Bhagoji et al., 2017; Ilyas et al., 2018). While extensive progresses have been made in white-box attacks (Carlini & Wagner, 2016; Xu et al., 2018; Evtimov et al., 2017), where attackers have complete knowledge about the target model, the more realistic scenario of blackbox attacks, where the attacker only has query access to the target model, remains a challenging topic. One major challenge is the excessive query complexity. For example, boundary-based blackbox attacks (BA) (Brendel et al., 2017) have shown promising attack effectiveness, but the required query number is too large to be practically feasible (e.g., many approaches require 10^5 or more queries per attack, which could take hours or even days given the rate limit of public APIs). This inefficiency stems partially from the high-dimensionality of the gradient since the Monte Carlo gradient estimation relies on sampling perturbations from the gradient space.

In this work, we study the properties of a general projection $\mathbf{f}$, which transforms vectors from low-dimensional subspace $\mathbb{R}^n$ to the original gradient space $\mathbb{R}^m$ for gradient estimation. We theoretically prove the general cosine similarity lower and upper bounds between the estimated and true gradients based on sampling distribution analysis and Taylor expansion. We then propose NonLinear-BA, which applies deep generative models such as AEs, VAEs, and GANs as the nonlinear projections to perform blackbox attack and therefore demonstrates the power of the projection-based gradient estimator empirically. We experimentally evaluate NonLinear-BA with three proposed nonlinear projections on four image datasets: ImageNet (Deng et al., 2009), CelebA (Liu et al., 2015), CIFAR-10 (Krizhevsky et al., 2009) and MNIST (LeCun et al., 1998). We show that NonLinear-BA can achieve 100% attack success rate with much smaller magnitude of perturbation efficiently. We also evaluate the NonLinear-BA against a commercial online API MEGVII Face++ (fac). Both quantitative and qualitative results are shown to demonstrate its superiority in terms of attack effectiveness.

Contributions: (1) We propose a novel nonlinear gradient projection-based BA (NonLinear-BA) which exploits the power of nonlinear-projection based gradient estimator and achieves the state-of-the-art performance. (2) We provide the first general theoretical analysis framework for analyzing the cosine similarity between estimated and true gradi-

[*]Equal contribution [1]University of Illinois at Urbana-Champaign [2]Ant Financial. Correspondence to: Huichen Li <huichen3@illinois.edu>, Linyi Li <linyi2@illinois.edu>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.

ents via different projections. (3) We prove and compare the lower bounds of gradient cosine similarity for linear and nonlinear projections. We also analyze the query complexity of the projection based gradient estimators. (4) We conduct extensive experiments on both offline ML models and commercial online APIs with high dimensional image datasets to demonstrate the high attack success of NonLinear-BA. The empirical results verify our theoretical findings that efficient projection-based gradient estimation via sampling in low dimension is possible and some projections would be more effective under certain conditions.

2. Problem Definition

In this section, we will first introduce the framework of *boundary-based blackbox attack*, and then focus on tackling the challenge of query-based gradient estimation.

Boundary-based blackbox Attack (BA). Given an instance x drawn from certain distribution $\mathcal{D}$: $x \sim \mathcal{D}$, where $x \in \mathbb{R}^m$, a C-way classification model $G : \mathbb{R}^m \mapsto \mathbb{R}^C$ is trained to output the confidence score for each class. The final prediction of the model is obtained by selecting the class with the highest confidence score $y = \text{argmax}_{i\in[C]}\, G(x)_i$ ($[C] = \{1, \ldots, C\}$). The model G is referred to as 'target model' throughout our discussion as it is the target of the adversarial attack. In this work we focus on the scenario where the adversaries do not have access to the details of model G (i.e. blackbox attack), and can only query the model to obtain the final prediction label y instead of the confidence scores.

The general framework of a BA is as follows: given a target-image $x_{tgt} \in \mathbb{R}^m$ whose true label is $y_{ben} \in [C]$, the attacker's goal is to craft an adversarial image x_{adv} that is predicted as a maliciously chosen label $y_{mal} \in [C]$, while the distance $D(x_{tgt}, x_{adv})$ between the two images is as small as possible. Here D is a L_p-norm based distance function which aims to restrict the perturbation added to the target-image in order to make it less noticeable. In this paper we only consider targeted attack with an intentionally chosen y_{mal} since untargeted attack is a trivial extension of the targeted case (by randomly sampling a y_{mal}).

Definition 1 ((G, y_{mal}))-Difference Function)**.** Given a model G, and malicious target y_{mal}, the *difference function* $S : \mathbb{R}^m \to \mathbb{R}$ is defined as $S(x) = G(x)_{y_{mal}} - G(x)_{y_{ben}}$, where y_{ben} denotes the ground truth label.

The difference function S is an important indicator of whether the image is successfully perturbed from being predicted as y_{ben} to y_{mal}. A *boundary-image* is an image x that lies on the decision boundary between y_{ben} and y_{mal} , i.e., $S(x) = 0$.

Projected gradient estimation. There are three main steps to perform the BA: (1) gradient estimation at G's decision boundary, (2) move the boundary-image along the estimated gradient direction, and (3) map the image back to the decision boundary. Typically, the first step requires to estimate the gradient based on the sign of *difference function* given multiple queries. It's computationally expensive as the high-dimensional gradient estimation requires a large number of queries (Chen et al., 2020).

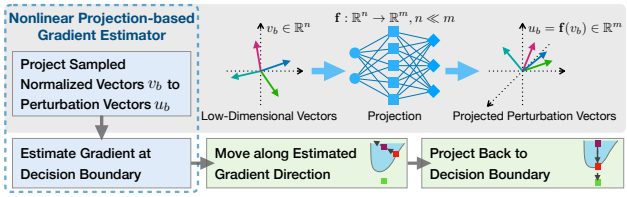

Figure 1. Algorithm overview for NonLinear-BA.

3. NonLinear-BA: Nonlinear Gradient Projection-based Boundary Blackbox Attack

In this section we introduce the proposed nonlinear gradient projection-based boundary blackbox attack NonLinear-BA as illustraed in Figure 1, followed by the detailed theoretical analysis and guarantees in Section 4.

In standard BA, the way to estimate the gradient given the query results is done by Monte Carlo sampling method (Chen et al., 2020):

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

where $x_{adv}^{(t)}$ is the boundary-image at iteration t obtained by binary search. The u_b's are B normalized perturbation vectors sampled from the whole gradient space. The size of random perturbation δ is chosen as a function of image size and the binary search threshold (Chen et al., 2020) to control the error of gradient estimation due to the fact the boundary-image never exactly lies on the decision boundary. The function $\text{sgn}\,(S(\cdot))$ denotes the sign of the difference-function (Definition 1). Its value is acquired by querying the victim model and compare the output label with y_{mal}. It is clear that the query cost is very high when the input dimension m is large. (Li et al., 2020) propose to search for a *representative subspace* with orthonormal mappings $\mathbf{W} = [w_1, \ldots, w_n] \in \mathbb{R}^{m\times n}, n \ll m$ and $\mathbf{W}^\intercal\mathbf{W} = I$. The perturbation vectors are generated by first sampling n dimensional unit vectors v_b and project them with $u_b = \mathbf{W}v_b$.

Nonlinear projection based gradient estimation. To search for the gradient representative subspaces more efficiently, we propose to perform the nonlinear projection-based gradient estimation. In particular, we propose to leverage generative models given their expressive power. Here we mainly consider AE, VAE and GAN as examples. There are two phases in NonLinear-BA: training and attacking. The detailed model structure and the training phase are described in Section H.1. We denote both the 'de-

coder' of AE and VAE and the 'generator' part of GAN as 'projection-based gradient estimator' in our following discussion. The gradient estimator is then used as the projection $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ in the attacking phase. We randomly sample unit latent vectors v_b's. Then the perturbation vectors generated as $u_b = \mathbf{f}(v_b)$ are used in the gradient estimation, yielding our gradient estimator as

$$\widetilde{\nabla S}(x_{adv}^{(t)}) = \frac{1}{B}\sum_{b=1}^{B} \operatorname{sgn}\left(S\left(x_{adv}^{(t)} + \delta \mathbf{f}(v_b)\right)\right)\mathbf{f}(v_b). \quad (2)$$

Move the image along estimated gradient direction and map back to decision boundary. After getting the estimated gradient $\widetilde{\nabla S}$, the boundary-image $x_{adv}^{(t)}$ is moved along that direction and mapped back to the decision boundary similar to (Chen et al., 2020).

4. Projected Gradient Estimation Analysis

To study the effectiveness of our projection-based gradient estimator in Equation (2) to improve the estimation accuracy and reduce the number of queries, in this section, we theoretically analyze the expected cosine similarity between the estimated gradient $\widetilde{\nabla S}(x_{adv}^{(t)})$ and the true gradient $\nabla S(x_{adv}^{(t)})$ for the boundary-image $x_{adv}^{(t)}$ at step t.

4.1. Generalized Gradient Estimator

We first formally define the gradient projection function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, which maps from the low-dimensional representative space $\mathbb{R}^n$ to the original high-dimensional space $\mathbb{R}^m$, where $n \leq m$.

Definition 2 (Generalized Projection-based Gradient Estimator)**.** Suppose $\mathbf{f}(x_0)$ is a boundary image, i.e., $S(\mathbf{f}(x_0)) = 0$, let $u_1, u_2, \ldots, u_B$ be a subset of orthonormal basis of space $\mathbb{R}^n$ sampled uniformly ($B \leq n$), we define

$$\widetilde{\nabla \mathbf{f}^\intercal \nabla S} := \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(\mathbf{f}(x_0 + \delta u_i)\right)\right) u_i. \quad (3)$$

Then, the *generalized gradient estimator* for $\nabla S(\mathbf{f}(x_0))$ is defined as

$$\widetilde{\nabla S}(\mathbf{f}(x_0)) := \nabla \mathbf{f}(x_0)\widetilde{\nabla \mathbf{f}^\intercal \nabla S}. \quad (4)$$

We sometimes abbreviate $\widetilde{\nabla S}(\mathbf{f}(x_0))$ as $\widetilde{\nabla S}$ for brevity.

All the aforementioned gradient estimators are concretization of this generalized gradient estimator with different projection $\mathbf{f}$, including HSJA (Chen et al., 2020), QEBA (Li et al., 2020) and our proposed NonLinear-BA. We defer the instantiations to Appendix B.

We now impose local Lipschitz and local smoothness conditions on the projection $\mathbf{f}$ and the difference function S.

Definition 3 (Local L-Lipschitz)**.** A (scalar or vector) function f is called local L-Lipschitz around x_0 with radius r, if for any two inputs $x, x' \in \{x_0 + \delta : \|\delta\|_2 \leq r\}$, $\frac{\|f(x)-f(x')\|_2}{\|x-x'\|_2} \leq L$.

Definition 4 (Local β-moothness)**.** A (scalar or vector) function f is called local β-smooth around x_0 with radius r, if (1) f is differentiable everywhere in region $\{x_0 + \delta : \|\delta\|_2 \leq r\}$; and (2) for any two inputs $x, x' \in \{x_0 + \delta : \|\delta\|_2 \leq r\}$, $\frac{\lambda_{\max}\left(\nabla f(x) - \nabla f(x')\right)}{\|x-x'\|_2} \leq \beta$, where $\lambda_{\max}(\mathbf{M})$ denotes the maximum eigenvalue of the matrix $\mathbf{M}$. Specifically, if M is a vector, $\lambda_{\max}(M) = \|M\|_2$.

Throughout the section, we assume the projection $\mathbf{f}$ is $L_{\mathbf{f}}$-Lipschitz and $\beta_{\mathbf{f}}$-smooth around x_0 with radius δ, and the difference function S is L_S-Lipschitz and β_S-smooth around $\mathbf{f}(x_0)$ with radius $L_{\mathbf{f}}\delta$.

For the convenience of our analysis, we define the constant ω as such:

Definition 5 (Gradient cosine similarity indicator ω)**.**

$$\omega := \delta\left(\frac{1}{2}\beta_{\mathbf{f}} L_S + \frac{1}{2}\beta_S L_{\mathbf{f}}^2 + \frac{1}{2}\delta\beta_{\mathbf{f}}\beta_S L_{\mathbf{f}} + \frac{1}{8}\delta^2\beta_{\mathbf{f}}^2\beta_S\right). \quad (5)$$

The Gradient cosine similarity indicator ω is an important quantity appearing in the cosine similarity lower bound. δ denotes the step size used in gradient estimation.

Theorem 1 (General Bound for Gradient Estimator)**.** *Let $\mathbf{f}(x_0)$ be a boundary image, i.e., $S(\mathbf{f}(x_0)) = 0$. 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 $u_1, u_2, \ldots, u_B$ in $\mathbb{R}^n$ 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*

$$\left(2\left(1 - \frac{\omega^2}{\|\nabla \mathbf{f}^\intercal \nabla S\|_2^2}\right)^{(n-1)/2} - 1\right)\frac{\|\nabla \mathbf{f}^\intercal \nabla S\|_2}{L_{\mathbf{f}}\|\nabla S\|_2}\sqrt{\frac{B}{n}}c_n \leq \mathbb{E}\cos\langle\widetilde{\nabla S}, \nabla S\rangle \leq \frac{\|\nabla \mathbf{f}^\intercal \nabla S\|_2}{l_{\mathbf{f}}\|\nabla S\|_2}\sqrt{\frac{B}{n}}c_n, \quad (6)$$

where ω is 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; $l_{\mathbf{f}} := \lambda_{\min}(\nabla \mathbf{f}(x_0))$.

We defer the detailed proof to Appendix C.

Remark. The theorem provides the cosine similarity bounds between our generalized gradient estimator $\widetilde{\nabla S}$ and the true gradient of the difference function ∇S. We remark that smaller ω implies larger cosine similarity lower bound. Detail discussions of the bound are presented in the following.

4.2. Gradient Estimation with Different Projections

We also show that for any linear projection $\mathbf{f}$, there exists a nonlinear projection $\mathbf{f}'$ which has the same local gradient

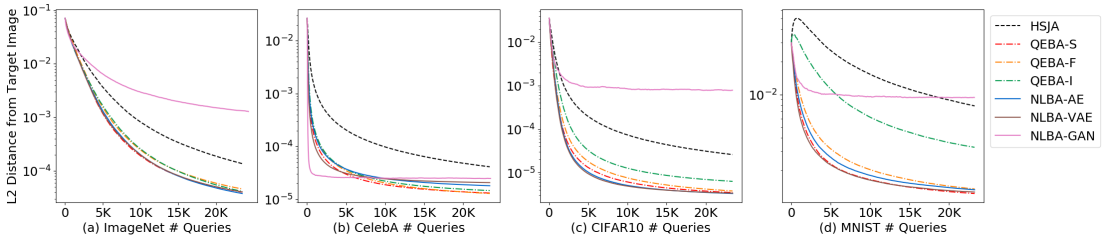

Figure 2. The perturbation magnitude based on different queries for attacks on diverse datasets.

$\nabla \mathbf{f}'(x_0)$ as $\nabla \mathbf{f}(x_0)$ but yields higher cosine similarity lower bound. The formal statement (Theorem 2) and proof are deferred to Appendix D.

Implications Based on the above results, we aim to further analyze two research questions.

Can we estimate gradients from a projected low-dimension subspace?
The answer is yes. According to Theorems 1 and 2, the cosine similarity between true gradient and estimated gradient depends on the ratio B/n, rather than only the subspace dimension n. Now we assume the number of queries B is equal to the subspace dimensionality n. We can observe a *sufficient condition* for good cosine similarity: $\|\nabla \mathbf{f}^\intercal \nabla S\|_2 / \|\nabla S\|_2$ is large, i.e., *the gradient* $\nabla \mathbf{f}$ *aligns with the gradient* ∇S *well.* We remark that the condition is independent with subspace dimensionality n.

On the other hand, we inspect the relation between cosine similarity bound and the number of queries B. As shown in Equation (6), both the lower and upper bound are in $\Theta(\sqrt{B})$ with respect to number of queries — to achieve a cosine similarity s, one need to perform $\Theta(s^2)$ number of queries. We formalize the query complexity analysis as below and defer the proof and discussion to Appendix C.

Corollary 1 (Query Complexity)**.** *Given projection* $\mathbf{f}$ *and 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* $\Theta(s^2)$.

How do different projections affect the gradient estimation quality?
The above results allow us to compare projection-based gradient estimators in different boundary attacks directly. We instantiate the general bound in Theorem 1 for HSJA and QEBA respectively, which shows that QEBA is significantly better than HSJA as it achieves the same cosine similarity with much fewer queries. For NonLinear-BA, Theorem 2 points out the possibility and a checkable sufficient condition where NonLinear-BA could be better than corresponding linear projection including HSJA and QEBA, in terms of providing higher lower bound of cosine similarity. *In a nutshell, the nonlinear projection which outperforms linear projection is not rare, however, its efficient search algorithm with theoretical guarantees is still unclear.*

5. Experiments

Blackbox Attack Performance on Offline Models We defer the discussion on experimental setup to Appendix F. Figure 2 shows the attack performance of different approaches in terms of the perturbation magnitude (L_2 distance). The attack success rates are shown in Figure 3 in appendix. The 'NonLinear-BA' is denoted as 'NLBA' in figures to save legend space. The NonLinear-BA with the three projection methods exhibit different patterns for the four datasets. NonLinear-BA-AE and NonLinear-BA-VAE are the most consistent across various datasets. They can achieve significantly better performance compared with baseline HSJA method, and outperform QEBA in many cases. The NonLinear-BA-GAN method, on the other hand, is less stable. We defer the discussion in Appendix J We show qualitative case studies of the attacks in Appendix K.1.

We also verify with quantitative results the theoretical findings on cosine similarity between the estimated and ground truth gradients as well as variable ω. The results and discussion are in Appendix I.2.

Attack Performance against Commercial APIs To demonstrate the practicality of the proposed NonLinear-BA, we also perform the blackbox attack against real-world online commercial APIs. Figure 6 in Appendix J shows the L_2 distance between the adv-image and the target-image with different numbers of queries. The attack success rate is always 100% during the whole process. From the figure it is clear that all the 6 gradient projection-based methods including both linear and non-linear projections are better than the baseline HSJA in terms of the L_2 distance under the same number of queries, and the nonlinear projection converges faster while observes slightly higher perturbation magnitude. The qualitative results of case studies are shown in Appendix K.2.

6. Conclusion

We propose NonLinear-BA, a projection-based gradient estimation approach for query-efficient boundary-based blackbox attack. We theoretically show nontrivial cosine similarity bounds for a group of projection based gradient estimation approaches and analyze the properties of different projections. We evaluate the efficiency of NonLinear-BA with extensive experiments against both offline ML models on four image datasets and commercial online APIs.

References

Face++. https://www.faceplusplus.com/.

Pretrained dcgan weights. https://github.com/csinva/gan-vae-pretrained-pytorch/.

Bhagoji, A. N., He, W., Li, B., and Song, D. Exploring the space of black-box attacks on deep neural networks. *arXiv preprint arXiv:1712.09491*, 2017.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Carlini, N. and Wagner, D. A. Towards evaluating the robustness of neural networks. corr abs/1608.04644 (2016). *arXiv preprint arXiv:1608.04644*, 2016.

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. 668–685, 2020.

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.

Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., and Song, D. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*, 2017.

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.

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

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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.

Marsaglia, G. et al. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43(2): 645–646, 1972.

MEGVII. Facial recognition 'compare' api. https://console.faceplusplus.com/documents/5679308, a.

MEGVII. Facial recognition 'compare' api query url. https://api-us.faceplusplus.com/facepp/v3/compare, b.

Muller, M. E. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20, 1959.

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.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

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

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

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

Wangni, J., Wang, J., Liu, J., and Zhang, T. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pp. 1299–1309, 2018.

Xu, X., Chen, X., Liu, C., Rohrbach, A., Darrell, T., and Song, D. Fooling vision and language models despite localization and attention mechanism. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

A. Related Work

The *blackbox attack* can be divided into two categories: transfer-based and query-based attacks. The transfer-based attacks rely on adversarial transferability (Papernot et al., 2016; Tramèr et al., 2017), where the adversarial examples generated against one ML model can also attack another model. The query-based attacks utilize the zero order information to estimate the gradient of the blackbox model via queries. *Boundary-based blackbox attack (BA)* (Brendel et al., 2017) is another type of query-based attack which only provides the final prediction instead of the prediction confidence scores for each query. Several work have been conducted to improve the query efficiency for BA. For instance, (Chen et al., 2020) applies the Monte-Carlo sampling strategy to perform gradient estimation for BA, and (Li et al., 2020) improves it by sampling from representative low-dimensional orthonormal subspace. Our work, on the other hand, aims to explore more general projection based gradient estimator with a unified theoretical analysis framework, as well as a more effective real-world blackbox attack approach.

B. Concretization 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 concretization 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 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.

HSJA. At 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

$$\widetilde{\nabla S}(\mathbf{f}(x_0)) = I\left(\frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta u_i\right)\right) u_i\right) = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta u_i\right)\right) u_i, \tag{7}$$

which is exactly the HSJA gradient estimator.

QEBA. At 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. The Equation (3) 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

$$\widetilde{\nabla S}(\mathbf{f}(0)) = \mathbf{W}\widetilde{\nabla \mathbf{f}^\intercal \nabla S} = \frac{1}{B}\sum_{i=1}^{B} \operatorname{sgn}\left(S\left(x_0 + \delta \mathbf{W}u_i\right)\right) \mathbf{W}u_i, \tag{8}$$

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 it with the generalized gradient estimator, 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 $\{r : \|r\| \leq 1\}$. 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 $\mathbf{g}(0) = x_0$:

$$\widetilde{\nabla S}(\mathbf{g}(0)) = \nabla\mathbf{g}(0)\left(\frac{1}{B}\sum_{i=1}^{B}\operatorname{sgn}\left(S\left(\mathbf{g}(\delta u_i)\right)\right)u_i\right) = \frac{1}{B}\sum_{i=1}^{B}\operatorname{sgn}\left(S\left(x_0 + \delta\mathbf{f}(u_i)\right)\right)\nabla\mathbf{g}(0)u_i \tag{9}$$

$$\approx \frac{1}{B}\sum_{i=1}^{B}\operatorname{sgn}\left(S\left(x_0 + \delta\mathbf{f}(u_i)\right)\right)\mathbf{f}(u_i), \tag{10}$$

where the Equation (10) is the NonLinear-BA gradient estimator in Equation (2). We implement NonLinear-BA gradient estimator by Equation (10) instead of the precise Equation (9) to avoid gradient computation and improve the efficiency.

Notice that in all these methods we perform boundary attack iterations in raw input space but in QEBA and NonLinear-BA we perform boundary gradient estimation in low dimension 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 boundary image 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.

C. 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*

$$p_a(x) := \frac{(1-x^2)^{(n-3)/2}}{\mathcal{B}\left(\frac{n-1}{2}, \frac{1}{2}\right)}, \quad x \in [-1, 1], \tag{11}$$

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 $(n-1)$-sphere. As the result, for any unit vector v, the distribution of $\langle u_i, v\rangle$ should be the same. Consider $e_1 = (1, 0, 0, \ldots, 0)^\intercal$,

$$a_i = \langle u_i, e_1\rangle = u_{i1}. \tag{12}$$

Now consider the distribution of u_{i1}, i.e., the first component of u_i. From (Muller, 1959; Marsaglia et al., 1972), 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. 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

$$f(x) = \int_0^\infty \frac{y^{\frac{n-1}{2}-1}\exp\left(-\frac{y}{2}\right)}{2^{\frac{n-1}{2}}\Gamma\left(\frac{n-1}{2}\right)} \cdot \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2 y}{2(1-x^2)}\right)\frac{\sqrt{y}}{(1-x^2)^{-3/2}}\mathrm{d}y = \frac{(1-x^2)^{\frac{n-3}{2}}}{\mathcal{B}\left(\frac{n-1}{2}, \frac{1}{2}\right)} \tag{13}$$

for $x \in (-1, 1)$. Combining Equation (12) and Equation (13), 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*

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

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 $\langle u,\, v\rangle \operatorname{sgn}\left(\mathbf{f}(x_0+\delta u)\right)$. Such alignment, along with Lemma Lemma 1, provides the bound for this sum of the product.

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

$$\mathbf{f}(x_0+\delta u) = \mathbf{f}(x_0) + J\cdot \delta u + \frac{1}{2}\sum_{i=1}^{n} (\theta\delta u)^\intercal \mathbf{T}(x_0)_i (\theta\delta u) = \mathbf{f}(x_0) + \delta J u + \frac{1}{2}\beta_{\mathbf{f}}\delta^2\epsilon, \tag{14}$$

$$S\left(\mathbf{f}(x_0+\delta u)\right) = S\left(\mathbf{f}(x_0)\right) + \nabla S^\intercal \left(\delta J u + \frac{1}{2}\beta_{\mathbf{f}}\delta^2\epsilon\right) + \frac{1}{2}\beta_S\left(\delta L_{\mathbf{f}} + \frac{1}{2}\beta_{\mathbf{f}}\delta^2\right)^2\theta_1 \tag{15}$$

$$= \delta \nabla S^\intercal J u + \delta^2\left(\frac{1}{2}\beta_{\mathbf{f}} L_S + \frac{1}{2}\beta_S L_{\mathbf{f}}^2 + \frac{1}{2}\delta\beta_{\mathbf{f}}\beta_S L_{\mathbf{f}} + \frac{1}{8}\delta^2\beta_S\beta_{\mathbf{f}}^2\right)\theta_2. \tag{16}$$

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 (14), 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 (15), 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 (15). In Equation (16), we use $S(\mathbf{f}(x_0)) = 0$ 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* (16), 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 the result, we rewrite Equation (16) 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:

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

As the result, 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., $S(\mathbf{f}(x_0)) = 0$. 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 (3)) and $\nabla \mathbf{f}(x_0)^\intercal \nabla S(\mathbf{f}(x_0))$ ($\nabla \mathbf{f}^\intercal \nabla S$ for short) satisfies*

$$\left(2\left(1-\frac{\omega^2}{\|\nabla \mathbf{f}^\intercal \nabla S\|_2^2}\right)^{(n-1)/2}-1\right)\cdot\frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\frac{1}{2}\right)\cdot(n-1)} \le \mathbb{E}\cos\langle\widetilde{\nabla \mathbf{f}^\intercal \nabla S},\ \nabla \mathbf{f}^\intercal \nabla S\rangle \le \frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\frac{1}{2}\right)\cdot(n-1)}. \tag{17}$$

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 $\lim_{\delta\to 0}\omega/\delta$ is a constant.

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

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

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

$$\cos\langle\widetilde{\nabla \mathbf{f}^\intercal \nabla S},\ \nabla \mathbf{f}^\intercal \nabla S\rangle = \frac{\langle\widetilde{\nabla \mathbf{f}^\intercal \nabla S},\ \nabla \mathbf{f}^\intercal \nabla S\rangle}{\|\widetilde{\nabla \mathbf{f}^\intercal \nabla S}\|_2\|\nabla \mathbf{f}^\intercal \nabla S\|_2} = \frac{1}{\sqrt{B}}\sum_{i=1}^{B}\operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)\langle v,\ u_i\rangle. \tag{18}$$

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 $\operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)\langle v,\ u_i\rangle$, in the worst case, only when $\|\langle v,\ u_i\rangle\| > \omega/\|J^\intercal \nabla S\|_2$, by Lemma 2, the $\operatorname{sgn}\left(S\left(\mathbf{f}(x_0+\delta u_i)\right)\right)$ is aligned with $\langle v,\ u_i\rangle$, 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 (18):

$$\mathbb{E}\cos\langle\widetilde{\nabla \mathbf{f}^\intercal \nabla S},\ \nabla \mathbf{f}^\intercal \nabla S\rangle \ge \frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\frac{1}{2}\right)\cdot(n-1)}\left(2\left(1-\frac{\omega^2}{\|\nabla \mathbf{f}^\intercal \nabla S\|_2^2}\right)^{(n-1)/2}-1\right). \tag{19}$$

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 (18):

$$\mathbb{E}\cos\langle\widetilde{\nabla\mathbf{f}^\intercal\nabla S},\, \nabla\mathbf{f}^\intercal\nabla S\rangle \le \frac{2\sqrt{B}}{\mathcal{B}\left(\frac{n-1}{2},\,\frac{1}{2}\right)\cdot(n-1)}. \tag{20}$$

□

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 $c_n \in (2/\pi,\, 1)$. Furthermore, $c_{n+2} < c_n$.

Remark. Using Lemma 4, we can simplify the frequent 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 $n \ge 5$, $d_n = \dfrac{\Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{n-1}{2}\right)} = \dfrac{n-2}{n-3}\cdot\dfrac{\Gamma\left(\frac{n-2}{2}\right)}{\Gamma\left(\frac{n-3}{2}\right)} = \dfrac{n-2}{n-3}d_{n-2}$. Notice that

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

and

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

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

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

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

(II.) Similarly, notice that

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

and

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

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

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

(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 proof 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., $S(\mathbf{f}(x_0)) = 0$. 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 $\widetilde{\nabla S}(\mathbf{f}(x_0))$ ($\widetilde{\nabla S}$ for short) and $\nabla S(\mathbf{f}(x_0))$ (∇S for short) satisfies*

$$\left(2\left(1 - \frac{\omega^2}{\|\nabla \mathbf{f}^\intercal \nabla S\|_2^2}\right)^{(n-1)/2} - 1\right) \frac{\|\nabla \mathbf{f}^\intercal \nabla S\|_2}{L_{\mathbf{f}} \|\nabla S\|_2} \sqrt{\frac{B}{n}} c_n \leq \mathbb{E} \cos\langle \widetilde{\nabla S}, \nabla S\rangle \leq \frac{\|\nabla \mathbf{f}^\intercal \nabla S\|_2}{l_{\mathbf{f}} \|\nabla S\|_2} \sqrt{\frac{B}{n}} c_n, \tag{21}$$

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 (4), we know $\widetilde{\nabla S} = \nabla \mathbf{f} \widetilde{\nabla \mathbf{f}^\intercal \nabla S}$, where $\nabla \mathbf{f}$ is the short 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,

$$\cos\langle \widetilde{\nabla S}, \nabla S\rangle = \cos\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S}, \nabla \mathbf{f}^\intercal \nabla S\rangle \frac{\left\|\widetilde{\nabla \mathbf{f}^\intercal \nabla S}\right\|_2 \|\nabla \mathbf{f}^\intercal \nabla S\|_2}{\left\|\widetilde{\nabla S}\right\|_2 \|\nabla S\|_2}. \tag{22}$$

According to the estimation formula of $\widetilde{\nabla \mathbf{f}^\intercal \nabla S}$ (Equation (3)), $\left\|\widetilde{\nabla \mathbf{f}^\intercal \nabla S}\right\|_2 = \sqrt{B}$. Furthermore, $\left\|\widetilde{\nabla S}\right\| \leq \lambda_{\max}(\nabla \mathbf{f}) \cdot \left\|\widetilde{\nabla \mathbf{f}^\intercal \nabla S}\right\|_2 \leq L_{\mathbf{f}}\sqrt{B}$, $\left\|\widetilde{\nabla S}\right\| \geq \lambda_{\min}(\nabla \mathbf{f}) \cdot \left\|\widetilde{\nabla \mathbf{f}^\intercal \nabla S}\right\|_2 = l_{\mathbf{f}}\sqrt{B}$, which means that

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

According to Equation (22), we have

$$\cos\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S}, \nabla \mathbf{f}^\intercal \nabla S\rangle \frac{\|\nabla \mathbf{f}^\intercal \nabla S\|_2}{L_{\mathbf{f}} \|\nabla S\|_2} \leq \cos\langle \widetilde{\nabla S}, \nabla S\rangle \leq \cos\langle \widetilde{\nabla \mathbf{f}^\intercal \nabla S}, \nabla \mathbf{f}^\intercal \nabla S\rangle \frac{\|\nabla \mathbf{f}^\intercal \nabla S\|_2}{l_{\mathbf{f}} \|\nabla S\|_2}. \tag{23}$$

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 (23) yields the desired bound. □

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

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 (6) with*

$$\omega := \frac{1}{2}\delta \beta_S L_{\mathbf{f}}^2. \tag{24}$$

We assume $\omega \leq \|\nabla \mathbf{f}^\intercal \nabla S\|_2$. 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 Appendix D.

Corollary 2 (restated)**.** *Given projection* $\mathbf{f}$ *and 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 1. From Theorem 1, we can observe that

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

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 expected cosine similarity and required query number when the projection $\mathbf{f}$ is fixed. Note that cosine similarity is bounded, i.e., even the totally aligned $\widetilde{\nabla S}$ and ∇S only have cosine similarity 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 in both QEBA and NonLinear-BA.

D. Proof of Existence of Better Nonlinear Projection

From Theorem 1, one may think that linear projection is better than nonlinear projection because when the $\nabla\mathbf{f}$ is the same, linear projection implies $\beta_{\mathbf{f}} = 0$, which leads to smaller ω and higher lower bound. However, this lower bound is applied to all models satisfying the Lipschitz and smoothness condition. In fact, there exists nonlinear projection $\mathbf{f}$ leading to higher cosine similarity lower bound. (We will focus on the discussion of lower bound below, since the upper bound is irrelevant with $\beta_{\mathbf{f}}$ from Theorem 1, meaning linear and nonlinear projections would share the same upper bound.)

Linear Case Firstly, let us consider the linear projection $\mathbf{f}$. Throughout the text, we use $\lambda_{\max}(\mathbf{M})$ to denote the largest eigenvalue of matrix $\mathbf{M}$, and $\lambda_{\min}(\mathbf{M})$ the smallest eigenvalue of matrix $\mathbf{M}$.

Corollary 2 (Linear projection Bound, informal)**.** *Under the same setting of Theorem 1 with additional condition that projection* $\mathbf{f}$ *is locally linear around* x_0 *with radius* δ *and* $L_{\mathbf{f}} := \lambda_{\max}(\nabla\mathbf{f}(x_0))$*, the expectation of cosine similarity satisfies Equation (6) with*

$$\omega := \frac{1}{2}\delta\beta_S L_{\mathbf{f}}^2. \tag{25}$$

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

Remark. We defer the formal statement to Appendix E.1. This is a direct application of Theorem 1 with $\beta_S = 0$ due to linearity. The main difference between the corollary and Theorem 1 is in ω, where the general ω in Equation (5) is altered by Equation (25). Furthermore, if S is also locally linear, then $\beta_S = 0$ and hence $\omega = 0$, which closes the gap between lower bound and upper bound and implies that the gradient estimation is pretty precise (with cosine similarity c_n).

Non-Linear Case For non-linear projection, we have the following theorem.

Theorem 2 (Existence of Better Nonlinear Projection, informal)**.** *Under the same setting of Corollary 2, 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 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 (6) with*

$$\omega := \frac{1}{2}\delta\beta_S L_{\mathbf{f}}^2 - \frac{1}{5}\beta_{\mathbf{f}}\beta_S\delta^2 L_{\mathbf{f}} < \frac{1}{2}\delta\beta_S L_{\mathbf{f}}^2. \tag{26}$$

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

Proof Sketch. We construct the nonlinear projection $\mathbf{f}'$ explicitly from $\mathbf{f}(x_0)$, $\nabla\mathbf{f}(x_0)$ and the difference function S. After showing that $\mathbf{f}'$ satisfies the assumptions in Section 4.1, we derive its cosine similarity bound with corresponding ω. □

Remark. The theorem shows that if the difference function S is nonlinear (i.e., $\beta_S > 0$), for any linear projection $\mathbf{f}$, we can exploit the nonlinearity to define its corresponding nonlinear projection $\mathbf{f}'$, which reduces ω and improves the cosine similarity.

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 $\mathbf{f}'(x_0) = J$, $\mathbf{f}'(x_0) = \mathbf{f}(x_0)$ and $\mathbf{f}'$ satisfies the assumptions, so that for arbitrary vector u with $\|u\|_2 = 1$,

$$\begin{aligned} \left\langle u, \frac{J^\intercal \nabla S}{\|J^\intercal \nabla S\|_2} \right\rangle > \frac{\omega}{\|J^\intercal \nabla S\|_2} &\implies \operatorname{sgn}(S(\mathbf{f}(x_0 + \delta u))) = 1, \\ \left\langle u, \frac{J^\intercal \nabla S}{\|J^\intercal \nabla S\|_2} \right\rangle < \frac{\omega}{\|J^\intercal \nabla S\|_2} &\implies \operatorname{sgn}(S(\mathbf{f}(x_0 + \delta u))) = -1. \end{aligned} \tag{27}$$

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

$$\mathbf{f}'(x_0 + u) = \mathbf{f}(x_0) + J \cdot u - \frac{1}{2}\alpha \|u\|_2 J u, \tag{28}$$

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 (28) (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 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

$$\nabla \mathbf{f}'(x_0 + u) = \left(1 - \frac{1}{2}\alpha \|u\|_2 \right) J - \frac{\alpha}{2\|u\|_2} J u u^\intercal. \tag{29}$$

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 (29),

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

Thus,

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

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 $u, v \in \mathbb{R}^n$,*

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

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.

$$\begin{aligned}\lambda_{\max}\left(\frac{uu^\intercal}{\|u\|_2}-\frac{vv^\intercal}{\|v\|_2}\right) &= \max_{\|w\|_2=1} w^\intercal\left(\frac{uu^\intercal}{\|u\|_2}-\frac{vv^\intercal}{\|v\|_2}\right)w = \max_{\|w\|_2=1}\frac{\|u^\intercal w\|_2^2}{\|u\|_2}-\frac{\|v^\intercal w\|_2^2}{\|v\|_2}\\ &= \max_{\|w\|_2=1}\|u\|\cos^2\langle u,\, w\rangle - \|v\|\cos^2\langle v,\, w\rangle.\end{aligned} \tag{30}$$

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.,

$$\text{Equation (30)} = \max_{\substack{\|w\|_2\le 1\\ w\in P_{uv}}}\|w\|^2\left(\|u\|\cos^2\langle u,\, w\rangle - \|v\|\cos^2\langle v,\, w\rangle\right) = \max_{\substack{\|w\|_2=1\\ w\in P_{uv}}}\left(\|u\|\cos^2\langle u,\, w\rangle - \|v\|\cos^2\langle v,\, w\rangle\right).$$

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

$$\begin{aligned}\text{Equation (30)} &= \max_{\beta}\|u\|\cos^2\beta - \|v\|\cos^2(\beta-\theta)\\ &= \max_{\beta}\frac{1}{2}\left(\|u\|-\|v\|\right)+\frac{1}{2}\left(\|u\|\cos 2\beta - \|v\|\cos 2(\beta-\theta)\right)\\ &= \frac{1}{2}\left(\|u\|-\|v\|\right)+\frac{1}{2}\left(\max_{\beta}\|u\|\cos\beta - \|v\|\cos(\beta-2\theta)\right).\end{aligned}$$

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 (30) $\le 1.5\|u-v\|$. □

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

To this point, we have proven the 3 arguments in Fact 2.1 respectively. □

Now we inject $\mathbf{f}$ into the Taylor expression for $S(\mathbf{f}'(x_0+\delta u))$, where u is a unit vector, i.e., $\|u\|_2 = 1$. Similar as Equations (14) to (16):

$$\begin{aligned} & S(\mathbf{f}'(x_0+\delta u)) \\ =& S\left(\mathbf{f}(x_0) + \delta J u - \frac{1}{2}\alpha\delta^2 J u\right) \\ =& S(\mathbf{f}(x_0)) + \delta \nabla S^\intercal J u - \frac{1}{2}\alpha\delta^2 \nabla S^\intercal J u + \frac{1}{2}\theta^2 \left(\delta J u - \frac{1}{2}\alpha\delta^2 J u\right)^\intercal \mathbf{H} \left(\delta J u - \frac{1}{2}\alpha\delta^2 J u\right), \end{aligned} \tag{31}$$

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:

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

Define $v := J^\intercal \nabla S(\mathbf{f}(x_0)) / \|J^\intercal \nabla S(\mathbf{f}(x_0))\|_2$. From Equation (31), we get

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

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

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

In other words,

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

satisfies the condition Equation (27). Following the same proof as in Theorem 1 using ω, we get the desired cosine similarity bound for the projection $\mathbf{f}'$. □

E. 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.

E.1. Comparison of Different Gradient Estimators

We instantiate the cosine similarity bound for gradient estimators in HSJA (Chen et al., 2020) and QEBA (Li et al., 2020). For our proposed NonLinear-BA, we use the general gradient estimator with nonlinear projection as the proxy. The definitions of these estimators are presented in Appendix B.

HSJA. In HSJA, the projection is just the identical. 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 (7), 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 with the dimentionality of the input space, m.

QEBA. In QEBA, the projection is a linear mapping with orthonormal coefficient 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 (8), 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 (Li et al., 2020) also present the same cosine similarity bound.

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 the 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 better cosine similarity. Here, we present another 'good' nonlinear projection, in order to show that such nonlinear projection is not rare and not 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 (6) with*

$$\omega < \frac{1}{2}\delta\beta_S L_{\mathbf{f}}^2. \tag{32}$$

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:

$$\mathbf{f}'(x_0+u) = \mathbf{f}(x_0) + J\cdot u + \frac{1}{2}\mathrm{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2 k\nabla S, \tag{33}$$

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

Fact 3.1. *The* $\mathbf{f}'$ *defined as Equation (33) has gradient* J *at point* x_0 *and is* $\beta_{\mathbf{f}}$ *smooth.*

Proof of Fact 3.1. Since

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

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

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

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

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

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

- 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 $\langle u_2,\, v\rangle<0$, $|\langle u_1+u_2,\, v\rangle|\leq|\langle u_1-u_2,\, v\rangle|$. Thus,

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

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

Now let us inject $\mathbf{f}'$ into the Taylor expression for $S\left(\mathbf{f}'(x_0+\delta u)\right)$ in a similar way as Equations (14) to (16), where u is a unit vector, i.e., $\|u\|_2=1$:

$$\begin{aligned}
&S(\mathbf{f}'(x_0+\delta u))\\
=&S\left(\mathbf{f}(x_0)+\delta Ju+\frac{1}{2}\operatorname{sgn}\left(\langle u,\, v\rangle\right)\langle u,\, v\rangle^2\delta^2 k\nabla S\right)\\
=&S(\mathbf{f}(x_0))+\delta\nabla S^\intercal Ju+\frac{1}{2}\operatorname{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2\delta^2 k\|\nabla S\|^2+\\
&\quad\frac{1}{2}\theta^2\left(\delta Ju+\frac{1}{2}\operatorname{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2\delta^2 k\nabla S\right)^\intercal\mathbf{H}\left(\delta Ju+\frac{1}{2}\operatorname{sgn}(\langle u,\, v\rangle)\langle u,\, v\rangle^2\delta^2 k\nabla S\right),
\end{aligned}\tag{34}$$

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:

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

When $\langle u,\, v\rangle>0$, from Equation (34), we get

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

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

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

Therefore,

$$|\langle u,\, v\rangle|\|v\|_2 \ge -\frac{1}{2}\langle u,\, v\rangle^2\delta k L_S^2 + \frac{1}{2}\beta_S\delta\left(L_{\mathbf{f}} + \frac{1}{2}\langle u,\, v\rangle^2\delta k L_S\right)^2 \implies \operatorname{sgn}(S(\mathbf{f}(x_0+\delta u))) = \operatorname{sgn}(\langle u,\, v\rangle). \tag{35}$$

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 $|\langle u,\, v\rangle| \ge \epsilon' > 0$,

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

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

As the result, from Equation (35), we know that when $|\langle u,\, v\rangle| \ge \epsilon'$, if $|\langle u,\, v\rangle|\|v\|_2 \ge \frac{1}{2}\beta_S\delta L_{\mathbf{f}}^2 - \eta$, $\operatorname{sgn}(S(\mathbf{f}(x_0+\delta u))) = \operatorname{sgn}(\langle u,\, v\rangle)$. In other words, let

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

then this ω' satisfies the condition Equation (27).

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 corresponding linear projection, and they also provide 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 and not 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.

E.2. Improve The Gradient Estimation

In Theorems 1 and 2, we relate the cosine similarity bound to variables characterizing projection $\mathbf{f}$ such as $\nabla\mathbf{f}$, $L_{\mathbf{f}}$, $\beta_{\mathbf{f}}$. By examining the change tendency of the bound 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 $L_{\mathbf{f}}$ is 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 mush as possible. Similar conclusion is shown for QEBA in Appendix E.1.
- Reduce 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 E.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 good nonlinear projection can be obtained, we consider the bound in Theorem 2, which shows the outcome of a good nonlinear projection. Learn from its ω in Equation (26), increasing $\beta_{\mathbf{f}}$, i.e., decrease the smoothness, could reduce ω and hence improve cosine similarity bound. If the good nonlienar projection may not be obtained, we consider the bound in Theorem 1, which bounds general projection. 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 outlined in (Chen et al., 2020).

F. Experimental Setup

Target Models. We use both offline models and commercial online API as target models. For offline models, following (Li et al., 2020), we use a pretrained ResNet-18 on ImageNet, CelebA, and we also evaluate on MNIST, and CIFAR-10 datasets. On CelebA, the target model is finetuned to perform classification on attributes. The most balanced attribute (e.g., 'Mouth_Slightly_Open') is chosen to enhance benign model performance. On MNIST and CIFAR-10, we scale up the input images to 224×224 with linear interpolation to demonstrate the query reduction for high-dimensional input space. The benign target model performance is shown in Appendix G.2. For commercial online APIs, we use the 'Compare' API (MEGVII, a) from MEGVII Face++ which predicts whether two images are of the same person. The implementation details are discussed in Appendix G.1.

Nonlinear Projection. To get the training data for the nonlinear projection, we first train five reference models on each of the image dataset. The benign accuracy for the reference models are shown in Appendix H.3. The ground truth gradients are generated using PyTorch's (Paszke et al., 2019) automatic differentiation functions. The details including model architectures and training parameters are described in Appendix H.1.

Evaluation Metrics. We mainly evaluate NonLinear-BA and compare with the baseline methods based on two standard evaluation metrics: (1) the average magnitude of perturbation at each step, as indicated by the L_2 distance between the optimized adversarial example and target-image; (2) the attack success rate after reaching some predefined L_2 distance threshold.

G. Target Models

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

G.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).

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.

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

Dataset	CelebA	CIFAR10	MNIST
Benign Accuracy	0.9417	0.8796	0.9938

Commercial Online API Among all the APIs provided by the Face++ platform (fac), we use the 'Compare' API (MEGVII, a) 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 (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, b). We set the similarity threshold as 50% in the experiments following (Li et al., 2020): when the confidence score is 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 adv-image that looks like the target-image (has low L_2 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 perturbation vector generators for comparison.

G.2. Model Performance of Target Models

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

H. 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.

H.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 structure except for the fact that the encoder part of VAE has two output layers in order 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.

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()

For CIFAR10 and MNIST, we use DCGAN (Radford et al., 2015) structure with pretrained weights from (pre) and add a linear interpolation layer to resize the generated images to size 224×224.

H.2. Estimator Training Procedure

The attacker first train a set of reference models that are generally 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, there are five reference models with different backbones compared with the target model, while the implementation and training details are similar with the target model in Section G.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 for each dataset are trained on the gradient images of the corresponding dataset generated as above.

H.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

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)

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 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

H.4. Nonlinear Projection-based Gradient Estimation

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

H.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.

I. Quantitative Results

I.1. Attack Success Rate for Offline Models

The 'successful attack' is defined as the adv-image reaching some pre-defined L_2 distance threshold. Note that because the complexity of tasks and images varies between datasets, we set different L_2 distance thresholds for the datasets. For

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 with 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 L_2 distance thresholds used for four datasets that determine whether the attack is successful.

Dataset	ImageNet	CelebA	MNSIT	CIFAR10
L_2 Threshold	1^{-3}	1^{-4}	5^{-3}	1^{-4}

example, ImageNet images are the most complicated so 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 3.

I.2. Cosine Similarity

The Theorems 1 and 2 state that smaller ω leads to higher cosine similarity between the estimated and ground truth gradients. The proof of these theorems reveals that ω is intuitively an indicator of how much percentage of queries are 'contributing negatively' to the gradient estimation, which is a complement of the cosine similarity values. To verify our theoretic findings, we also plot the gradient cosine similarity values corresponding to different queries in Figure 4. It is clear that the attack performance highly correlates with the cosine similarity positively: when the cosine similarity values are large, the attack performance is better and can converge to a smaller L_2 distance faster. We also use an alternative method to evaluate the effects of ω approximately and observe similar trends as shown in Appendix I.3.

I.3. Proxy for the ω Value

According to the analysis in Section 4.1, smaller ω leads to better gradient estimation. 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 negative contribution to the gradient estimation if

$$\text{sgn}\left(S\left(x_{adv}^{(t)} + \delta \mathbf{f}(v_b)\right)\right) \neq \text{sgn}\left(\cos\left\langle(\widetilde{\nabla S}(x_{adv}^{(t)}), \mathbf{f}(v_b)\right\rangle\right). \tag{36}$$

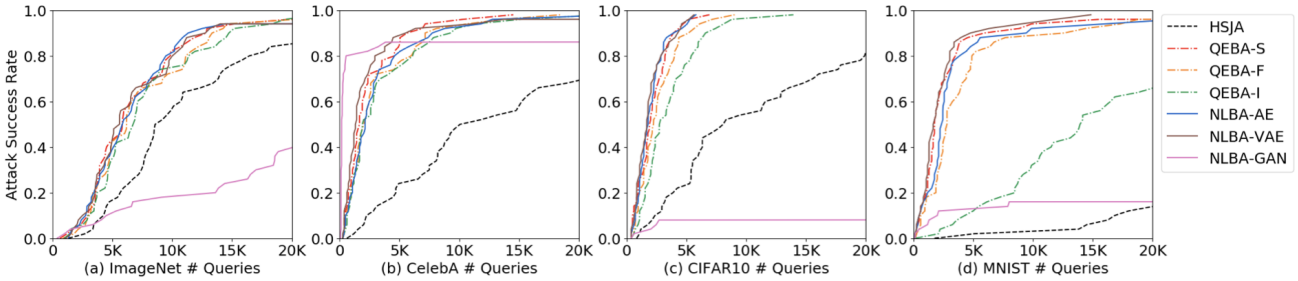

Figure 3. The attack success rate vs query number for four different datasets.

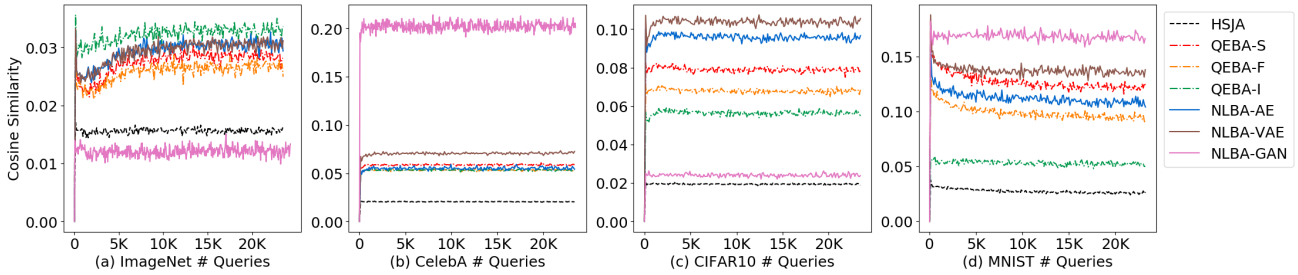

Figure 4. The cosine similarity between the estimated and truth gradients based on different queries for attacks on diverse datasets.

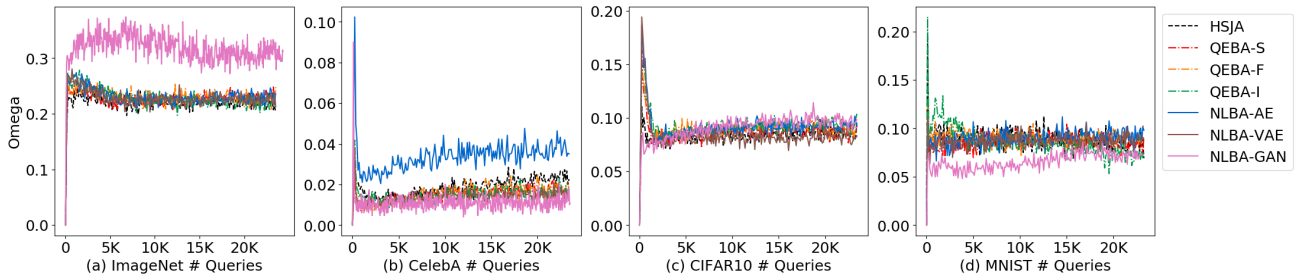

Figure 5. The ω value at different queries for attacks on diverse datasets.

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 (36) as the proxy of ω. The results are shown in Figure 5. The tendency is highly consistent with the cosine similarity between the estimated and truth gradients - smaller ω leads to higher cosine similarity.

J. Quantitative Results

J.1. Discussion on NonLinear-BA-GAN

For attribute classification model on CelebA dataset where the model's ground truth gradients have a simpler pattern, it works significant better than the other methods with very few queries (Fig 2(b)); when the gradient patterns are more complex, the NonLinear-BA-GAN method fails to keep reducing the L_2 distance after some relatively small number of queries and converges to a bad local optima. We conjecture this is because of the instability of GAN training, and it would be interesting future work to develop in-depth understanding about the properties of nonlinear projection GAN.

J.2. Online API

The quantitative results of the API attack is shown in Figure 6. The results are averaged over 40 randomly sampled CelebA face image pairs. The image pairs are the same for each of the 7 methods for fair comparison.

K. Qualitative Results

K.1. 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 7, Figure 8, Figure 9 and Figure 10 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

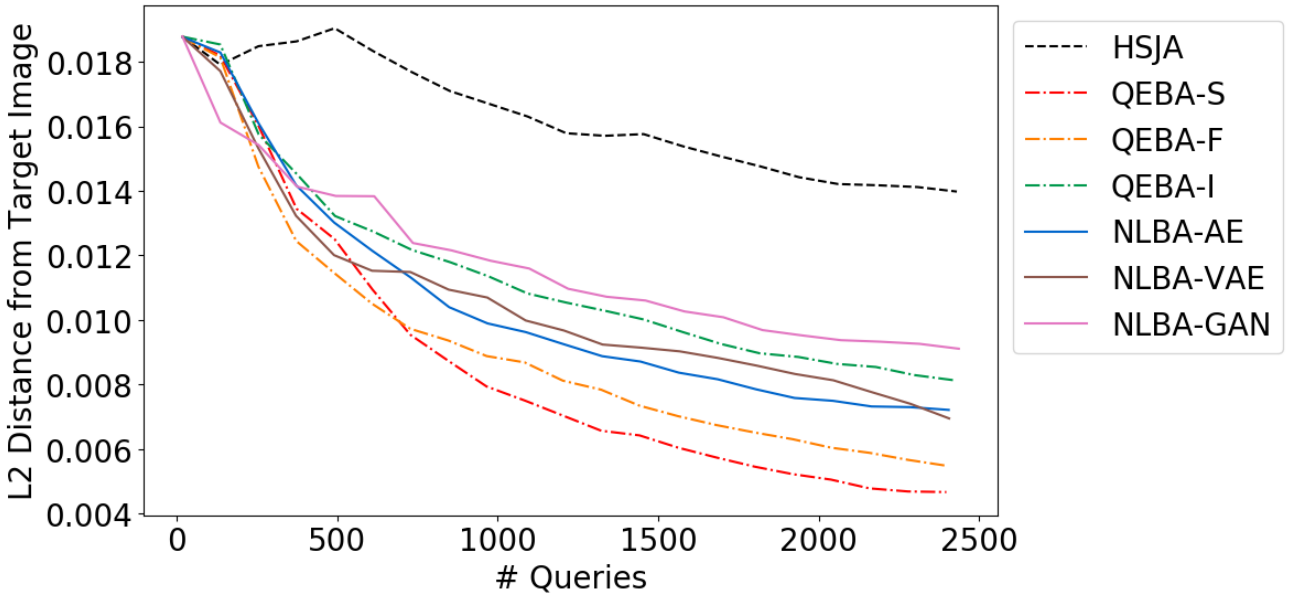

Figure 6. The perturbation magnitude based on different queries against Face++ 'Compare' API.

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 L_2 distance between the adv-image and the target-image. The smaller d can get, the better the attack is.

K.2. 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 11. In the figure, the source-image and target-image are shown on the left-most column.

Figure 7. The qualitative case study of attacking ResNet-18 model on ImageNet dataset.

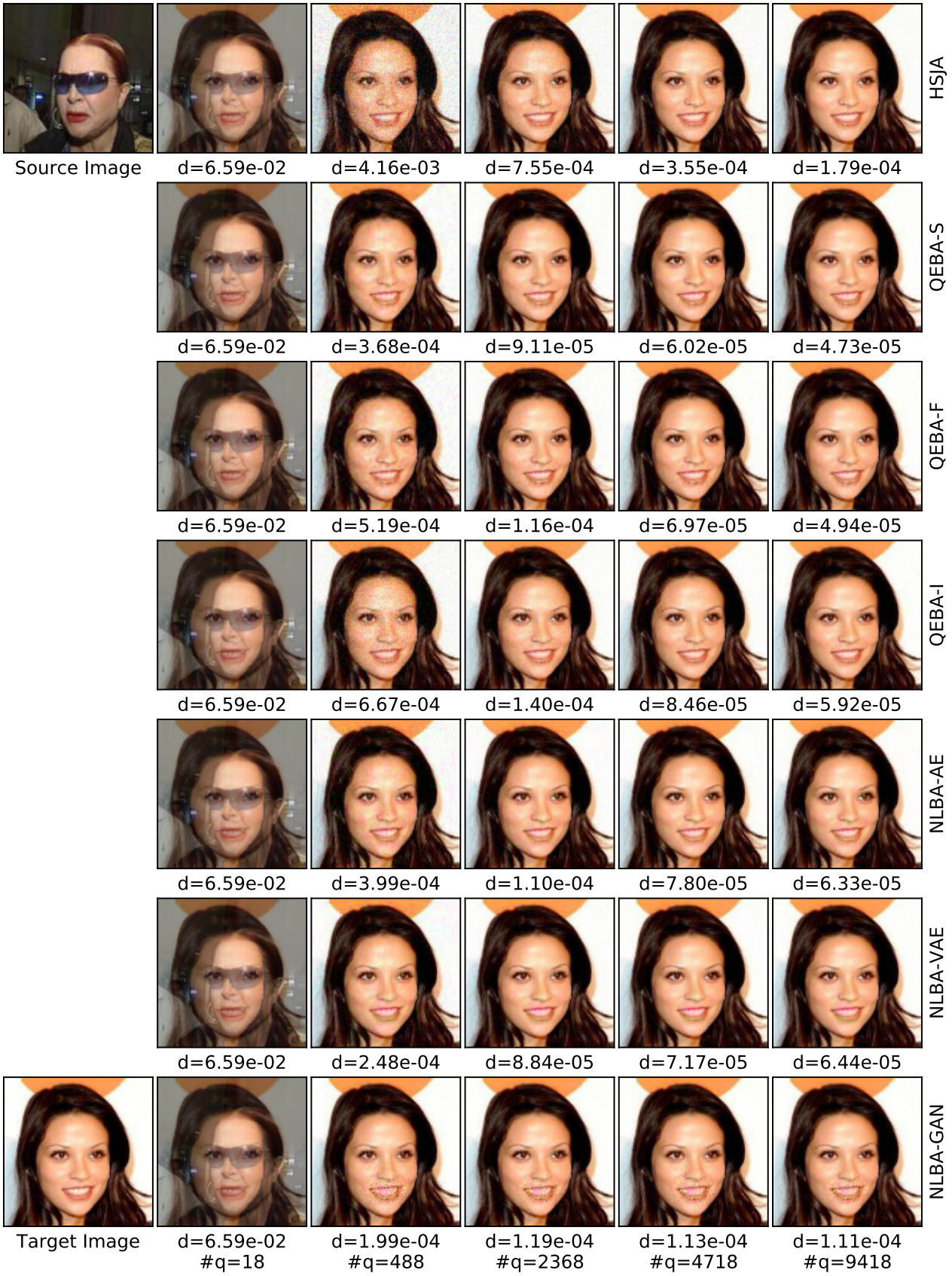

Figure 8. The qualitative case study of attacking ResNet-18 model on CelebA dataset.

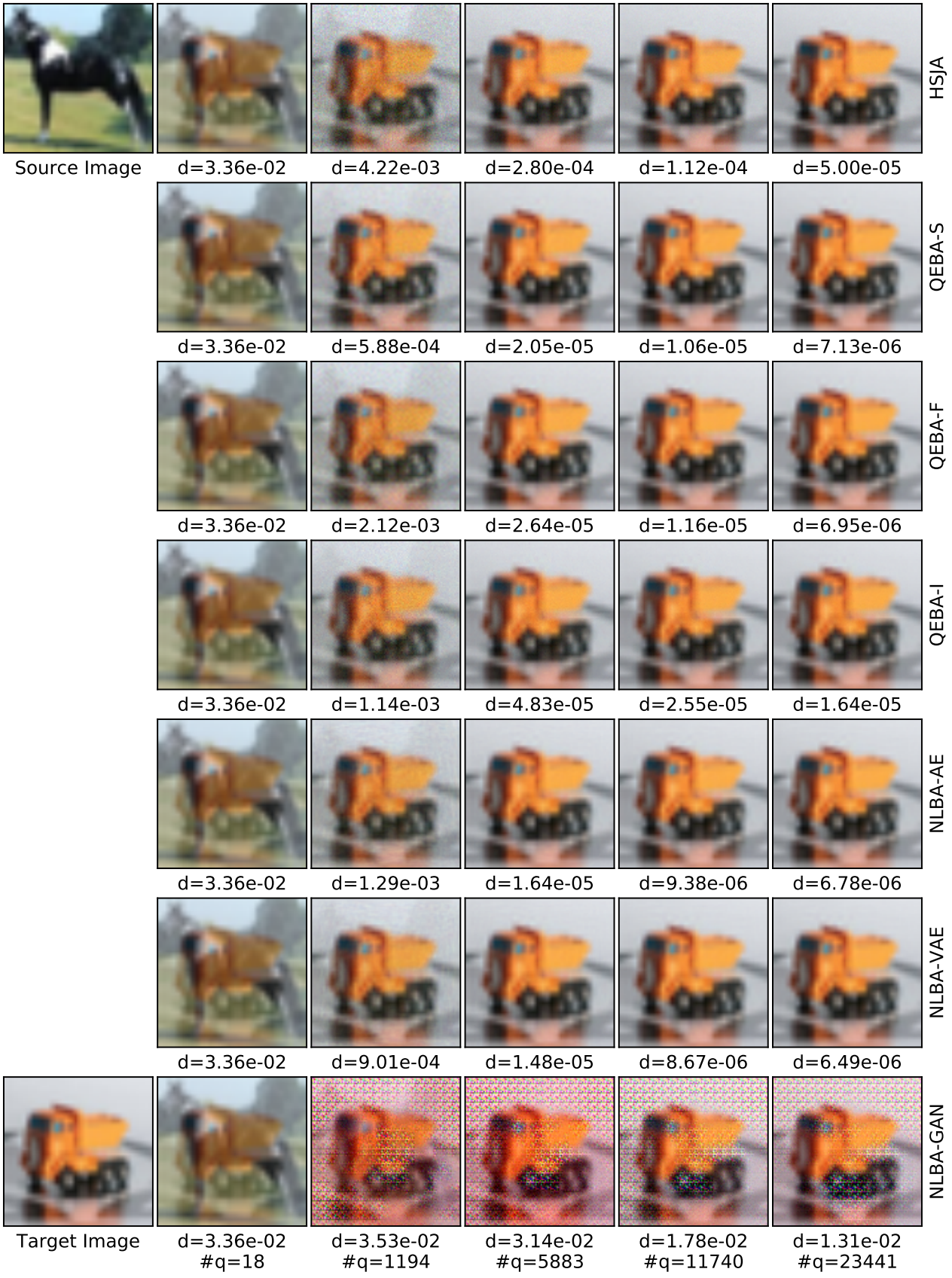

Figure 9. The qualitative case study of attacking ResNet-18 model on CIFAR10 dataset.

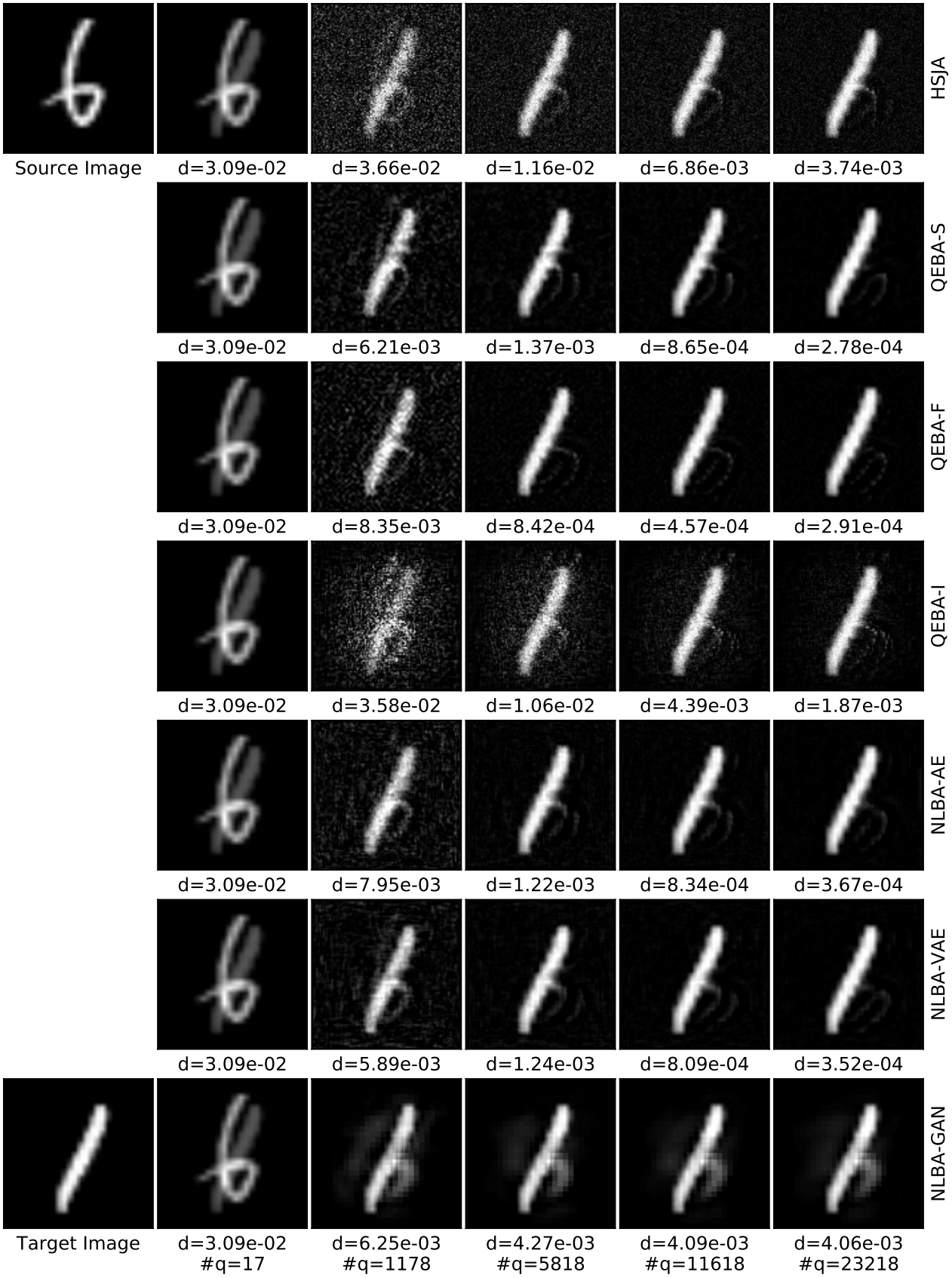

Figure 10. The qualitative case study of attacking ResNet-18 model on MNIST dataset.

Figure 11. A case study of Face++ online API attack process. The source-target image pair is randomly sampled from CelebA dataset (ID: 163922 and 080037).