Table 1: Attack performance of two counterparts following the same setting as main Table 1.

Methods	Inception-V3		ResNet-50		VGG-16		Inception-V4		IncRes-V2	
	ASR	AVG.Q	ASR	AVG.Q	ASR	AVG.Q	ASR	AVG.Q	ASR	AVG.Q
NATTACK Li et al. (2019) ICML'19	98.2%	936.1	99.5%	621.3	99.7%	313.0	97.5%	1826.2	92.8%	2688.5
Square Andriushchenko et al. (2020) ECCV'20	99.4%	351.9	99.8%	401.4	**100.0%**	**142.3**	98.3%	475.6	94.9%	670.3
SimBA++ (Ours)	99.2%	295.7	**99.9%**	187.3	99.9%	166.0	98.3%	420.2	95.8%	555.1
LeBA (Ours)	**99.4%**	**243.8**	**99.9%**	**178.7**	99.9%	145.5	**98.7%**	**347.4**	**96.6%**	**514.2**

To all reviewers: We deeply appreciate all reviewers for the high-quality reviews, especially in the shadow of COVID-19. All discussions in this rebuttal will be included in the revised paper. All typos and writing issues will be refined.

Counterpart Methods: As suggested by reviewers, in Table 1, we additionally report the performance of NATTACK Li et al. (2019) and Square Attack Andriushchenko et al. (2020) (current SOTA without surrogate models). Although under different threat models (we use surrogate models while they do not), the proposed SimBA++ and LeBA outperform both counterparts (except for VGG-16). We will also provide their performance over defensive models in the revised paper.

Writing Structure: We will adjust the writing structure to make the paper self-contained, including simplifying the related work, moving Algorithm A3 (LeBA) into main text, and experiment setting/ablation studies into appendix.

R1: We do appreciate your positive feedback and suggestions. On targeted attack, due to its intensive computational cost, we provide the performance against Inception-V3 under 10,000 queries (same as the untargeted): SimBA (48.3%, 6465.4), SimBA++ (60.1%, 3472.6) and LeBA (66.7%, 4197.7). There is no surprise; both contributions introduced by SimBA++ and LeBA still remain effective. We will report all results under 60,000 queries in the revised appendix.

R2: Thank you for insightful comments. We would like to emphasize that our method is intuitive yet effective; instead of efficient gradient estimation with surrogate models (P-RGF), our method (SimBA++) significantly outperforms previous SOTA, which inspires us to rethink the way to combine transferability-based and query-based attack. Besides, we provide the first method to efficiently update the surrogate model with limited queries.

R3: Thank you for the very detailed suggestions for improving the structure and writing of our paper! We will carefully revise our writing and language issues as per your advice. To respond your concerns, first, we would like to emphasize that though the proposed SimBA++ is straightforward, it outperforms previous SOTA. This simple baseline could make the community rethink the ways to use surrogate models. Second, the improvement of LeBA over SimBA++ seems "marginal", but it is hard to obtain: SimBA++ has already outperformed previous SOTA by large margins, LeBA could further improve the query efficiency consistently in all the experiments. We believe the setting of LeBA is reasonable in practical scenario: we expect higher query attack efficiency with more query feedback obtained from victim models. Please note that LeBA becomes effective just when query procedure begins (not after 1,000 queries totally completed). As depicted in main Table 2, LeBA (*test*) is more efficient than LeBA (*training*). Third, we absolutely agree with you that the threat model without surrogate models is also important, which is the reason why we emphasize the threat model used in our paper (with surrogate models). Both threat models are theoretically and practically important .

As suggested, we report the performance of Square Attack Andriushchenko et al. (2020) on the same images as ours. Please note that the attacked images are critical to the reported performance, therefore it is not suitable to compare them directly. As shown in Table 1, our methods (both SimBA++ and LeBA) outperforms the Square Attack in most cases. The brilliant ideas in Square Attack could also be integrated into ours (e.g., replace SimBA with Square Attack).

R4: Thank you for the golden comments! First, L_2 and L_∞ threat models are both critical in practice. It is important to select perturbation pixels for L_2 while not for L_∞. Therefore, many black-box attack studies focus on only 1 threat model (e.g., L_2 in SimBA). Even if both threat models are addressed in certain studies, the algorithms for L_2 and L_∞ are different (e.g., P-RGF and Square Attack). We focus on L_2 threat model only in our study. Second, PGD (2000 steps) achieves 100%, 100%, 99.9% success for the defensive models in main Table 3, respectively. Note that JPEG is not differentiable, we report the BPDA extension [2] of PGD. As suggested, it will be reported in main Table 3 as a reference for white-box attack. Third, we report the performance of NATTACK Li et al. (2019) in Table 1. As NATTACK focus on the attack success over defensive models, we will also report its performance as per main Table 3.

References

Andriushchenko, M. et al. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.

Li, Y. et al. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *ICML*, 2019.