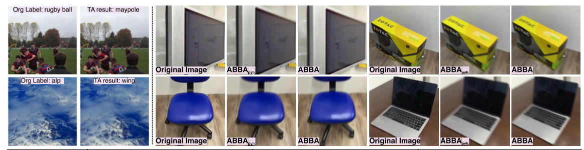
Common response: We re-emphasize our key contribution and novelty: ❶ **Difference from adversarial noise and other blurs.** The motion blur and additive noise are 2 levels/scopes of perturbations that are orthogonal from the perspective of camera perception. Specifically, additive noise mainly investigates the influence of additive distortions on the received image to DNNs, while motion blur considers the perception system's front-end, *i.e.*, the motion of object or camera. Motion blur often occurs in the physical process of practical image perception and can potentially post serious effects on safety and security, making it of great importance. Compared with other image blurs (*e.g.*, defocus blur), motion blur, as an intrinsic phenomenon, directly relates to the motion of object and camera and cannot easily be removed by adjusting camera setting. ❷ **Key contribution.** Although extensive work has been conducted on attacking/defensing for adversarial noise, up to present, limited studies have been performed on how motion blur affects DNN-based prediction. This work initiates the first step to comprehensively investigate motion blur effects of camera perception from the perspective of adversarial attack and proposes the motion-based adversarial blur attack (ABBA). ❸ **Benefits to blur-robustness enhancement (R4).** We conduct an experiment that trains the IncResv2 on the clean imagenet (1.1M) and a modified imagenet (1.3M) containing ABBA-blurred images (0.2M), and evaluate the accuracy on the motion-blur subsets of ImageNetC. We see that the Top1 error of IncResv2 decreases from 73.0% to 53.2% with our blurred images, which strongly demonstrates the impact of ABBA to enhance the blur-robustness of DNNs.



	Adv. from Inc-v3			DeblurGANv2			Re-DeblurGANv2		
	Inc-v3	Inc-v4	IncRes-v2	Inc-v3	Inc-v4	IncRes-v2	Inc-v3	Inc-v4	IncRes-v2
ABBA	65.3	31.1	30.0	31.4	24.2	18.4	22.9	22.5	16.8
NormalBlur	36.4	20.8	18.5	15.2	10.0	4.7	26.4	18.1	13.7

Figure R-1: (Top-L) subfigure shows two examples of the targeted attack via ABBA. (Top-R) subfigure shows four examples of our ABBA and $\text{ABBA}_{\text{Pyhsical}}$ that performs attack in the real-world with the estimated translation parameters of ABBA. (Bottom) Succ. Rate of ABBA and NormalBlur before (Adv. from Inc-v3) and after deblurring via existing and retrained DeblurGANv2s.

Q1 (R1): Targeted attack (TA) and more real-world examples. We can intuitively achieve the **TA** that is to fool a classifier to predict a specified category by replacing the $\max$ objective function (Eq. (5)) with a $\min$ objective function towards the specified category. We give two TA examples and our real-world examples in Fig. R-1(T).

Q2 (R2): Explanation of the NormalBlur in Sec. 3.5. NormalBlur generates motion-blurred image by optimizing Eq. (5) while fixing all kernel elements as $\frac{1}{N}$, which is equivalent to averaging neighbouring video frames where object and background move uniformly. In contrast, ABBA effectively tunes kernel elements to fool DNNs. Actually, the intention of Sec 3.5 is to study the effectiveness of existing deblurring method (*i.e.*, the 'already-deployed' deblurring modules) in defending the attack of ABBA with the tunable kernels. We thank the reviewer's suggestion in using the deblurring modules trained from ABBA. More detailed response: ❶ **NormalBlur** utilizes Eq. (5) to generate motion blur and **has considered background motion** via optimizing θ_b. ❷ In practice, our assumption is that **we cannot get real motion information in the scene** and there is only one given static image. Hence, our attack is conducted under this assumption, *i.e.*, the object and background move uniformly (*i.e.*, at fixed speed) in a short time, which is a common phenomenon in the real world (*e.g.*, walking). Our attack could be easily extended to other cases where more motion information is available (*e.g.*, video). ❸ With the DeblurGANv2 trained on normal motion blur dataset (*e.g.*, GOPRO [22]), the decrease in Succ. Rate before and after deblurring in Fig. R-1(B) have shown that the NormBlur can be defended more easily than ABBA, which demonstrates ABBA's tunable kernels facilitate achieving high attack success rate and anti-deblurring capability. **As suggested by the reviewer, when we further retrained the DeblurGANv2 with blurred images from ABBA.** ABBA can be defended more easily, which further indicates a promising direction of combining ABBA and the existing deblurring method for effective defense.

Q3 (R4): ABBA does not generate the motion blur on the whole image. As defined in Eq. (4) and (5), ABBA jointly (but differently) tunes the object and background's translation parameters (*i.e.*, θ_o and θ_b) to generate motion-blurred adversarial images. The visualization results in Fig. 3 in the submission, Fig. III and Fig. VII in the supplementary material all already demonstrate that the motion blur of object and background can be different.

Q4 (R3): Explanation of ABBA's performance. ❶ $\textbf{ABBA}_{\textbf{pixel}}$ **vs. ABBA.** $\text{ABBA}_{\text{pixel}}$ achieves strong attack capability since we perform fine-grained tuning for the kernel of each pixel independently (Eq. (3)). However, this can make $\text{ABBA}_{\text{pixel}}$ generate perceptible noise-like images (Fig. 2). To generate more realistic blur, we further propose the ABBA that uses the saliency regularization to constraint kernels to be the same in both object and background regions, which trades off the attack success rate a bit. ❷ **Advantages over averaging neighbouring video frames and SOTA noise-based attacks.** We have already compared ABBA with the 'averaging neighbouring video frames', *i.e.*, NormalBlur in Sec. 3.5 and the column 'Adv. from Inc-v3' of Fig. R-1(B). Obviously, ABBA achieves much higher success rate and transferability than NormalBlur. Moreover, compared with SOTA noise-based attacks (Tab. 1 in submission), $\text{ABBA}_{\text{pixel}}$ and ABBA obtain the best and second best results across all defense methods.

Q5 (R4): *c.f.* [14]. We have made our best efforts to cite and compare with [14]. The failure for experimental comparison is due to missing of key data/model components and fundamentally technical differences: ❶ We have made private communication with the authors [14] for pre-trained models and training data. However, both are unavailable due to commercial reasons. ❷ Technically, [14] needs two neighboring frames as the input while we focus on generating visually natural motion blur with only one static image as inputs. Moreover, [14] relies on an offline-trained UNet for realistic motion blur instead of and without focusing on conducting the attack.