A. Evaluation images

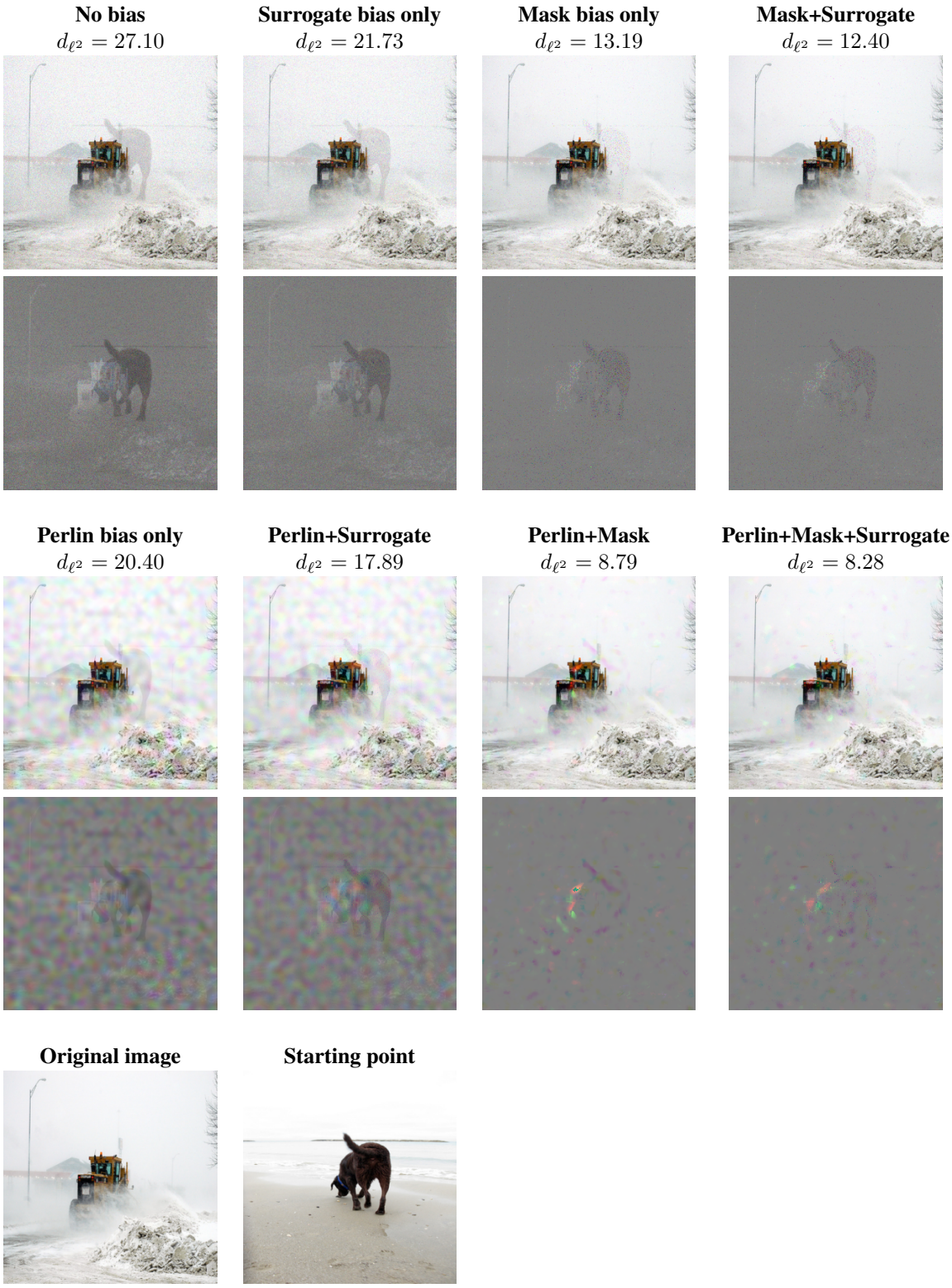

Figure 1. Targeted adversarial examples on ImageNet, obtained with different biases after 15000 iterations. The original class is "snowplow" – all images are classified as the target "Chesapeake Bay retriever". The mask bias is especially effective, as start and original image have similar backgrounds. See https://github.com/ttbrunner/biased_boundary_attack for an animated version.

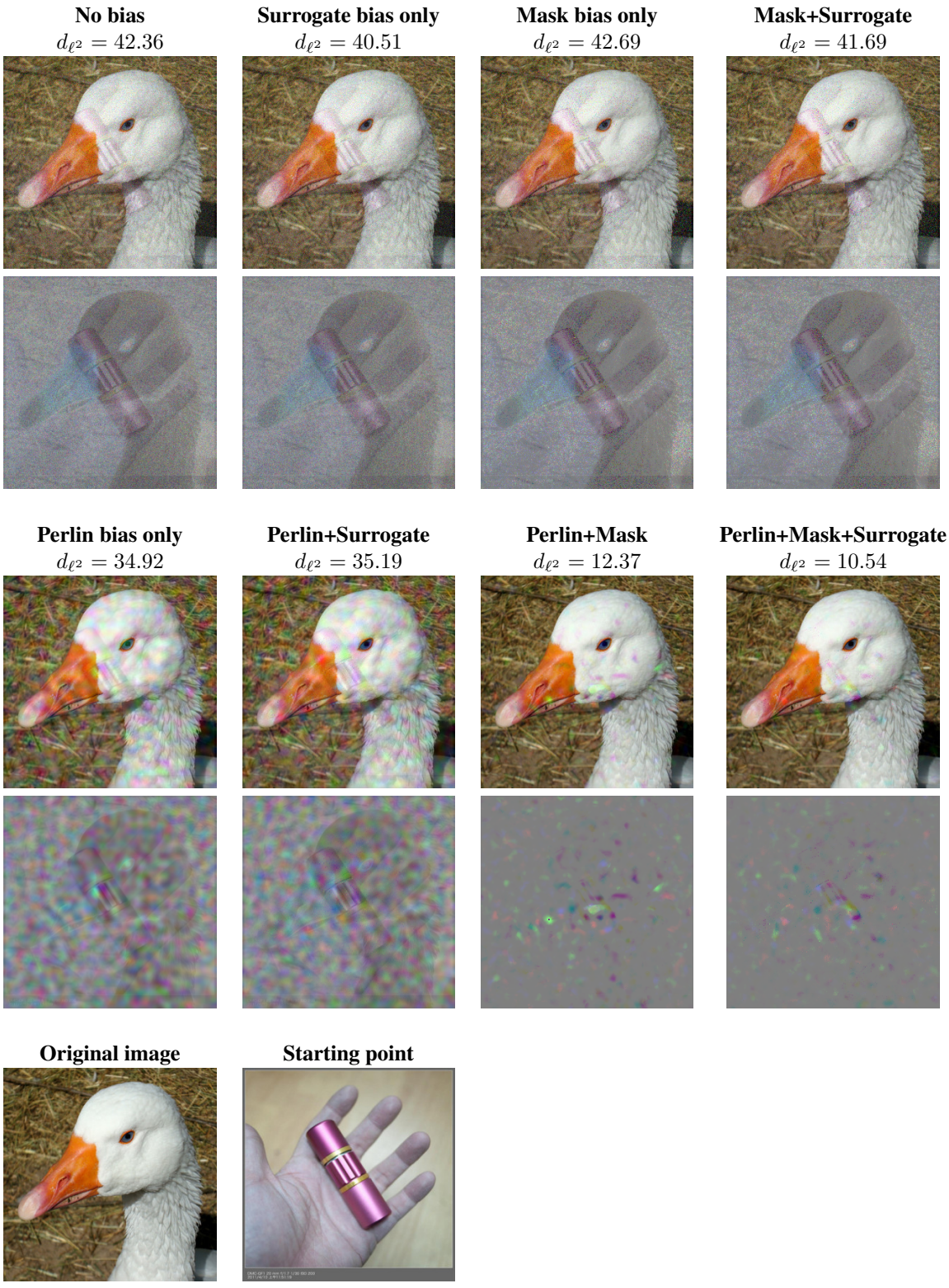

Figure 2. Targeted adversarial examples on ImageNet, obtained with different biases after 15000 iterations. The original class is "goose" – all images are classified as the target "lipstick". In the case of this image, not every bias comes with an improvement: mask+surrogate is even slightly worse than surrogate only. However, when all biases are combined the result is still significantly better. See https://github.com/ttbrunner/biased_boundary_attack for an animated version.

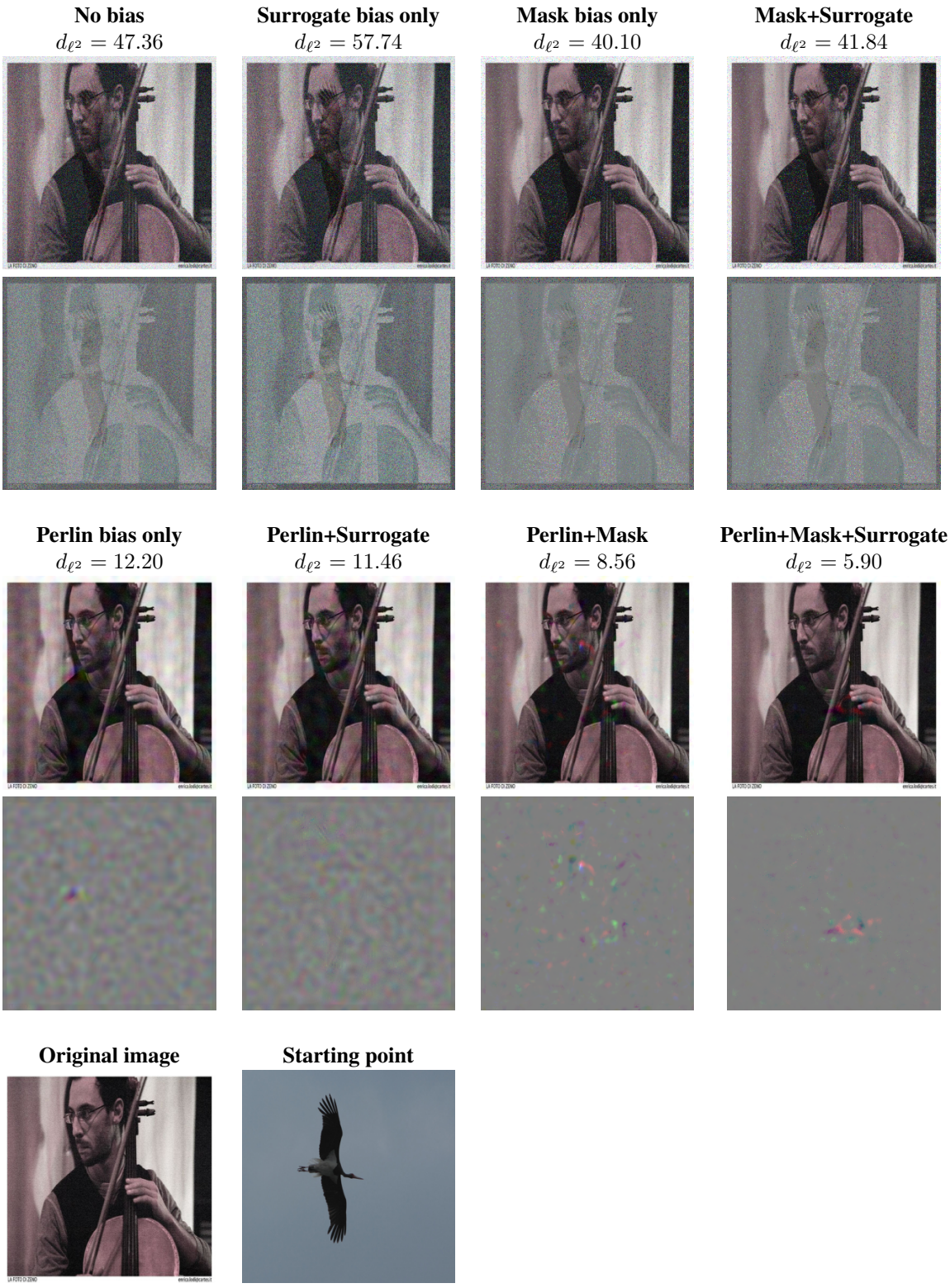

Figure 3. Targeted adversarial examples on ImageNet, obtained with different biases after 15000 iterations. The original class is "cello" – all images are classified as the target "black stork". When used on its own, the surrogate bias seems to be detrimental for this particular image. Still, the final result is impressive: when comparing no biases with all biases, the perturbation norm is reduced by 88%. See https://github.com/ttbrunner/biased_boundary_attack for an animated version.

B. Evaluation Hyperparameters

B.1. Boundary Attack

Step sizes. In the source code of their original implementation, Brendel et al. [1] suggest setting both the orthogonal step to $\eta = 0.01$ and the source step to $\epsilon = 0.01$. Both step sizes are relative to the current distance from the source image. We instead set $\eta = 0.05$ and $\epsilon = 0.002$, as this makes the attack take smaller steps towards the source image, while at the same time allowing for more extreme perturbations in the orthogonal step. We have found this to increase the success chance of perturbation candidates, and the attack gets stuck less often.

Step size adaptation. We do not use the original step size adjustment scheme as proposed by Brendel et al. [1]. They collect statistics about the success of the orthogonal step before performing the step towards the source, and based on this they either reduce or increase the individual step sizes.

This method seems to be geared towards reaching near-zero perturbations and less towards query efficiency, which is our primary goal – we are interested in making as much progress as possible in the early stages of an attack. When testing the Boundary Attack with query counts below 15000, we found the success statistics to be very noisy and the adaptation scheme ended up being detrimental. Therefore, we opt for a different approach:

- At every iteration, we count the number of consecutive previously unsuccessful candidates.
- As this number increases, we dynamically reduce both step sizes towards zero.
- Whenever a perturbation is successful, the step size is reset to its original value.
- As a fail-safe, the step size is also reset after 50 consecutive failures. Typically, we found this to occur often for the unbiased Boundary Attack, but very seldom when using the Perlin bias.

As a result, our strategy is quick to reduce step size, and after success immediately reverts to the original step size. We have found this to be very effective in the early stages of an attack. However, it has the drawback of wasting samples in the later stages (10000+ queries), when it tries to revert to larger step sizes too often. It might be promising to partially reinstate the approach of Brendel et al., or to apply some form of step size annealing.

B.2. Other attacks

For all other attacks, we use the hyperparameters that are provided for ImageNet in the publicly available source code of their implementations.

C. Submission to NeurIPS 2018 Adversarial Vision Challenge

When evaluating adversarial attacks and defenses, it is hard to obtain meaningful results. Very often, attacks are tested against weak defenses and vice versa, and results are cherry-picked. We sidestep this problem by instead presenting our submission to the NeurIPS 2018 Adversarial Vision Challenge (AVC), where our method was pitted against state-of-the-art robust models and defenses and won second place in the targeted attack track.

Evaluation setting. The AVC is an open competition between image classifiers and adversarial attacks in an iterative black-box decision-based setting [2]. Participants can choose between three tracks:

- Robust model: The submitted code is a robust image classifier. The goal is to maximize the ℓ^2 norm of any successful adversarial perturbation.
- Untargeted attack: The submitted code must find a perturbation that changes classifier output, while minimizing the ℓ^2 distance to the original image.
- Targeted attack: Same as above, but the classification must be changed to a specific label.

Attacks are continuously evaluated against the current top-5 robust models and vice versa. Each evaluation run consists of 200 images with a resolution of 64x64, and the attacker is allowed to query the model 1000 times for each image. The final attack score is then determined by the median ℓ^2 norm of the perturbation over all 200 images and top-5 models (lower is better).

Competitors. At the time of writing, the exact methods of most model submissions were not yet published. But seeing as more than 60 teams competed in the challenge, it is reasonable to assume that the top-5 models accurately depicted the state of the art in adversarial robustness. We know from personal correspondence that most winning models used variations of Ensemble Adversarial Training [3], while denoisers were notably absent. On the attack side, most winners used variants of PGD transfer attacks, again in combination with large adversarially-trained ensembles.

Dataset. The models are trained with the Tiny ImageNet dataset, which is a down-scaled version of the ImageNet classification dataset, limited to 200 classes with 500 images each. Model input consists of color images with 64x64 pixels, and the output is one of 200 labels. The evaluation is conducted with a secret hold-out set of images, which is not contained in the original dataset and unknown to participants of the challenge.

C.1. Random guessing with low frequency

Before implementing the biased Boundary Attack, we first conduct a simple experiment to demonstrate the effectiveness of Perlin noise patterns against strong defenses. Specifically, we run a random-guessing attack that samples candidates uniformly from the surface of a ℓ^2-hypersphere with radius ϵ around the original image:

$$s \sim \mathcal{N}(0,1)^k;\ x_{adv} = x_0 + \epsilon \cdot \frac{s}{\|s\|_2} \tag{1}$$

With a total budget of 1000 queries to the model for each image, we use binary search to reduce the sampling distance ϵ whenever an adversarial example is found. First experiments have indicated that the targeted setting may be too difficult for pure random guessing. Therefore we limit this experiment to the untargeted attack track, where the probability of randomly sampling *any of 199* adversarial labels is reasonably high. We then replace the distribution with normalized Perlin noise:

$$s \sim Perlin_{64,64}(v);\ x_{adv} = x_0 + \epsilon \cdot \frac{s}{\|s\|_2} \tag{2}$$

We set the Perlin frequency to 5 for all attacks on Tiny ImageNet. As Table 1 shows, Perlin patterns are more efficient and the attack finds adversarial perturbations with much lower distance (63% reduction). Although intended as a dummy submission to the AVC, this attack was already strong enough for a top-10 placement in the untargeted track. An example obtained in this experiment can be seen in Figure 1.

C.2. Biased Boundary Attack

Next, we evaluate the biased Boundary Attack in our intended setting, the targeted attack track in the AVC. To provide a point of reference, we first implement the original Boundary Attack without biases. This works, but is too slow for our setting. Compare Figure 4, where the starting point is still clearly visible after 1000 iterations (in the unbiased case).

Distribution	Median ℓ^2
Normal	11.15
Perlin noise	**4.28**

Table 1. Random guessing with low frequency (untargeted), evaluated against the top-5 models in the AVC.

Boundary Attack Bias	Median ℓ^2
None	20.2
Perlin	15.1
Perlin + Surrogate gradients	**9.5**

Table 2. Biases for the Boundary Attack (targeted), evaluated against the top-5 models in the AVC.

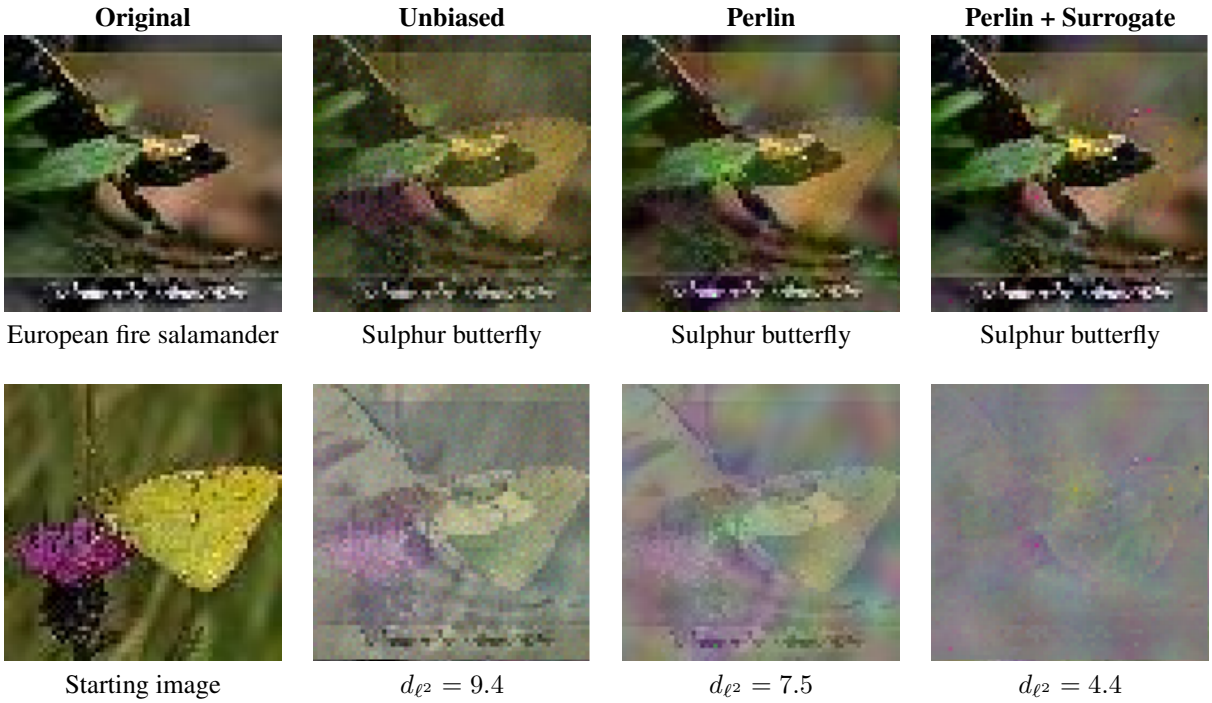

Figure 4. Adversarial examples generated with different biases in our targeted attack submission to the AVC. All images were obtained after 1000 queries. The isolated perturbation is shown below each adversarial example.

Perlin bias. We add our first bias, low-frequency noise. As before, we simply replace the distribution from which the attack samples the orthogonal step with Perlin patterns. See Table 2, where this alone decreases the median ℓ^2 distance by 25%.

Surrogate gradient bias. We also add projected gradients from a surrogate model and set the bias strength w to 0.5. This further reduces the median ℓ^2 distance by another 37%, or a total of 53% when compared with the original Boundary Attack. 1000 iterations are enough to make the butterfly almost invisible to the human eye (see Figure 4).

Here, the efficiency boost is much larger than in our ImageNet evaluation in Section 4. This may be due to our choice of surrogate models: In our submission to the AVC, we simply combined the publicly available baselines (ResNet18 and ResNet50). This ensemble is notably stronger than the simple model we used for the ImageNet evaluation, as the ResNet50 model is adversarially trained. However, it is also significantly weaker than the ones used by other winning AVC attack submissions, most of which were found to use much larger ensembles of carefully-trained models.[1] Nevertheless, our attack outperformed most of them which reinforces our earlier claim: Our method seems to make more efficient use of surrogate models than direct transfer attacks.

Mask Bias. We did not implement the mask bias in our entry to the AVC because of time constraints.

The source code of our submission is publicly available.[2]

[1] https://medium.com/bethgelab/results-of-the-nips-adversarial-vision-challenge-2018-e1e21b690149
[2] https://github.com/ttbrunner/biased_boundary_attack_avc

References

[1] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. 4

[2] Wieland Brendel, Jonas Rauber, Alexey Kurakin, Nicolas Papernot, Behar Veliqi, Marcel Salathé, Sharada P. Mohanty, and Matthias Bethge. Adversarial vision challenge. *arXiv preprint arXiv:1808.01976*, 2018. 5

[3] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 5