Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks

Thomas Brunner[1,2] Frederik Diehl[1,2] Michael Truong Le[1,2] Alois Knoll[2]
[1] fortiss GmbH [2] Technical University of Munich
{brunner,diehl,truongle}@fortiss.org knoll@in.tum.de

Abstract

We consider adversarial examples for image classification in the black-box decision-based setting. Here, an attacker cannot access confidence scores, but only the final label. Most attacks for this scenario are either unreliable or inefficient. Focusing on the latter, we show that a specific class of attacks, Boundary Attacks, can be reinterpreted as a biased sampling framework that gains efficiency from domain knowledge. We identify three such biases, image frequency, regional masks and surrogate gradients, and evaluate their performance against an ImageNet classifier. We show that the combination of these biases outperforms the state of the art by a wide margin. We also showcase an efficient way to attack the Google Cloud Vision API, where we craft convincing perturbations with just a few hundred queries. Finally, the methods we propose have also been found to work very well against strong defenses: Our targeted attack won second place in the NeurIPS 2018 Adversarial Vision Challenge.

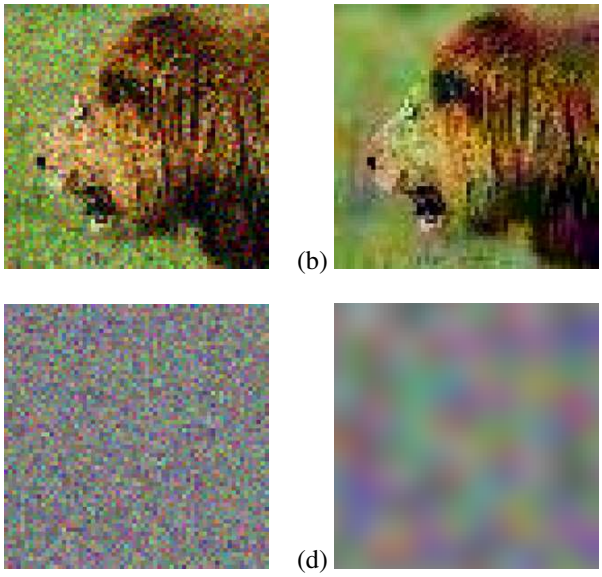


Figure 1. (a) and (b): Black-box adversarial examples obtained by random sampling. (c) and (d): isolated perturbation patterns. (c) is sampled from a normal distribution, and (d) is sampled from a distribution of Perlin noise patterns, which is one of the biases we propose. Both (a) and (b) fool the classifier, but (b) can be obtained with fewer samples.

1. Introduction

Ever since the term was fist coined, adversarial examples have enjoyed much attention from machine learning researchers. The fact that tiny perturbations can lead otherwise robust-seeming models to misclassify an input could pose a major problem for safety and security. But when discussing adversarial examples, it is often unclear how realistic the scenario of a proposed attack truly is. In this work, we consider a threat setting with the following parameters:

Black-box. The black-box setting assumes that an attacker has access only to the input and output of a model. Compared to the white-box setting, where an attacker has complete access to the architecture and parameters of the model, attacks in this setting are significantly harder to conduct: Most state-of-the-art white-box attacks [7, 4, 13] rely on gradients that are directly computed from the model parameters, which are not available in the black-box setting.

Decision-based classification (label-only). Depending on the output format of the model, the problem of missing gradients can be circumvented. In a score-based scenario, the model provides real-valued outputs (for example, softmax activations). By applying tiny modifications to the input, an attacker can estimate gradients by observing changes in the output [5] and then follow this estimate to generate adversarial examples. The decision-based setting, in contrast, provides only a single discrete result (the top-1 label) on which gradient estimation is very inefficient [9]. This form of black-box attack is much more difficult, but also extends the range of possible targets in the real world [3].

Limited queries. Black-box attacks might not be feasible if they need millions of queries to the model, and possibly multiple hours' time, to be successful. We therefore consider a scenario where the attacker must find a convinc-

ing adversarial example in less than 15000 queries.

Targeted. An untargeted attack is considered successful when the classification result is any label other than the original. Depending on the number and semantics of the classes, it can be easy to find a label that requires little change, but is considered adversarial (*e.g.* Egyptian cat vs Persian cat). A targeted attack, in contrast, needs to produce exactly the specified label. This task is strictly harder than the untargeted attack, further decreasing the probability of success.

In this setting, current state-of-the-art attacks are either unreliable or inefficient. Our contribution is as follows:

- We show how a recently proposed method, the Boundary Attack, can be re-framed as a biased sampling framework that gains efficiency from prior beliefs about the target domain.
- We discuss three such biases: low-frequency patterns, regional masks and gradients from a surrogate model.
- We evaluate the effectiveness of each bias and show that their combination drastically outperforms the previous state of the art in label-only black box attacks. Our source code is publicly available.[1]

2. Related Work

There currently exist two major schools of attacks in the threat setting we consider:

2.1. Transfer-based

It is known that adversarial examples display a high degree of transferability, even between different model architectures [19]. Transfer attacks seek to exploit this by training substitute models that are reasonably similar to the model under attack, and then apply regular white-box attacks to them.

Typically, this is performed by iterative applications of fast gradient-based methods such as the Fast Gradient Sign Method (FGSM) [7] and, more generally, Projected Gradient Descent (PGD) [13]. The black-box model is used in the forward pass, while the backward pass is performed with the surrogate model [1]. In order to maximize the chance of a successful transfer, newer methods use large ensembles of substitute models, and applying adversarial training to the substitute models has been found to increase the probability of finding strong adversarial examples even further [18].

Although these methods currently form the state of the art in label-only black-box attacks [11], they have one major weakness: as soon as a defender manages to reduce transferability, direct transfer attacks often run a risk of complete failure, delivering no result even after thousands of iterations. As a result, conducting transfer attacks is a cat-and-mouse game between attacker and defender, where the attacker must go to great lengths to train models that are just as robust as the defender's. Therefore, transfer-based attacks can be very efficient, but also somewhat unreliable.

2.2. Sampling-based

Circumventing this problem, sampling-based attacks do not rely on direct transfer and instead try to find adversarial examples by randomly sampling perturbations from the input space.

Perhaps the simplest attack consists of sampling a hypersphere around the original image, and drawing more and more samples until an adversarial example is found. Owing to the high dimensionality of the input space, this method is very inefficient and has been dismissed as completely unviable [17]. While this is not our main focus, we show in Appendix C that even this crude attack can be accelerated and made competitive in certain scenarios.

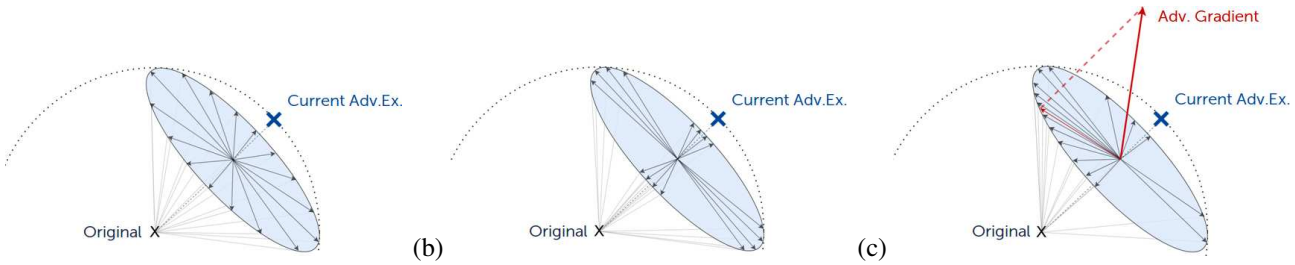
Recently, a more efficient attack has been proposed: the Boundary Attack (BA) [3]. This attack is initialized with an input of the desired class, and then takes small steps along the decision boundary to reduce the distance to the original input. Previous works have established that regions which contain adversarial examples often have the shape of a "cone" [18], which can be traversed from start to finish. At each step, the BA employs random sampling to find a sideways direction that leads deeper into this cone. From there, it can then take the next step towards the target.

The BA has been shown to be very powerful, producing adversarial examples that are competitive with even the results of state-of-the-art white-box attacks [3]. However, its weakness is query efficiency: to achieve these results, the attack typically needs to query a model hundreds of thousands of times.

It should be noted that recent research on black-box attacks has largely focused on classifiers that provide confidence scores, which is an easier setting. Nevertheless, many of these methods also use random sampling [5, 20, 9], and the biases we propose could also benefit their approaches. As an aside, Ilyas et al. [9] propose a variation of their attack that manages to apply gradient estimation to discrete labels. Although this does fit our setting, we find it to be much less efficient than BA variants (see Section 4).

Clearly, sampling-based attacks are very flexible but often too inefficient for practical use. Barring pure random guessing, the BA is the simplest attack for our setting. We therefore choose to focus on this method, and show how it can benefit from the biases we propose.

[1] https://github.com/ttbrunner/biased_boundary_attack



(a) (b) (c)

Figure 2. Sampling directions for the orthogonal step. (a) Boundary Attack: uniformly distributed along the surface of the hypersphere. (b) BA with Perlin bias: higher sample density in the direction of low-frequency perturbations. (c) BA with surrogate gradient bias: samples further concentrate towards the direction of the projected gradient.

3. Biased Boundary Attacks

The Boundary Attack, like most sampling-based attacks, draws perturbation candidates from a multidimensional normal distribution. This means that it performs unbiased sampling, perturbing each input feature independently of the others. While this is very flexible, it is also extremely inefficient when used against a robust model.

Consider the distribution of natural images: adjacent pixels are typically not independent of each other, but often have similar colors. This alone is a strong indicator that drawing perturbations from i.i.d random variables will lead to adversarial examples that are clearly out of distribution for natural image datasets. This, of course, renders them vulnerable to detection and filtering - robust models have become increasingly resilient against such patterns [11].

Therefore, it seems only logical to constrain the search space to perturbations that we believe to have a higher chance of success, or to bias the distribution so that the probability of sampling them increases.

We outline three such biases for the domain of image classification, discuss their motivation and show how to integrate them into the sampling procedure of the Boundary Attack.

3.1. Low-frequency perturbations

When one looks at typical adversarial examples, it quickly becomes apparent that most existing methods yield perturbations with high image frequency. But high-frequency patterns have a significant problem: they are easily identified and separated from the original image signal, and are often dampened by spatial transforms. Indeed, most of the winning defenses in the NeurIPS 2017 Adversarial Attacks and Defences Competition were based on denoising [12], simple median filters [11], and random transforms [21]. In other words: state-of-the art defenses are designed to filter high-frequency noise.

At the same time, we know that it is possible to synthesize "robust" adversarial examples which are not easily filtered in this way. Compare Athalye et al. [2]: Their robust perturbations are largely invariant to filters and transforms, and – interestingly enough – at first glance seem to contain very little high-frequency noise.

Inspired by this observation, we hypothesize that image frequency alone could be a key factor in robustness of adversarial perturbations. If true, then simply limiting perturbations to the low-frequency domain should increase the success chance of an attack, while incurring no extra cost.

Perlin Noise patterns. A straightforward way to generate parametrized, low-frequency patterns, is to use Perlin Noise [14]. Originally intended as a procedural texture generator for computer graphics, this function creates low-frequency noise patterns with a reasonably "natural" look. One such pattern can be seen in Figure 1d. But how can we use it to create a prior for the Boundary Attack?

Let k be the dimensionality of the input space. The original Boundary Attack (Figure 2a) works by applying an orthogonal perturbation η^k along the surface of a hypersphere around the original image, in the hope of moving deeper into an adversarial region. From there, a step is taken towards the original image. In the default configuration, candidates for η^k are generated from samples $s \sim \mathcal{N}(0,1)^k$, which are projected orthogonally to the source direction and normalized to the desired step size. This leads to the directions being uniformly distributed along the hypersphere.

To introduce a low-frequency prior into the Boundary Attack, we instead sample from a distribution of Perlin noise patterns (Figure 2b). Perlin noise is typically parametrized with a permutation vector v of size 256, which we randomly shuffle on every call. Effectively, this allows us to sample two-dimensional noise patterns $s \sim Perlin_{h,w}(v)$, where h and w are the image dimensions (and $h \cdot w = k$). As a result, the samples are now strongly concentrated in low-frequency directions.

Our experiments in Section 4 show that this greatly improves the efficiency of the attack. Therefore, we reason that the distribution of Perlin noise patterns contains a higher concentration of adversarial directions than the normal distribution.

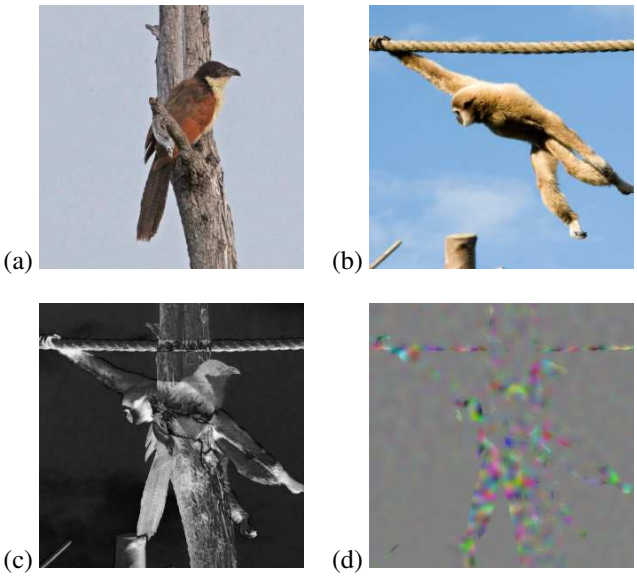


Figure 3. Masking based on per-pixel difference. (a) shows the original image, (b) an image of the target class. (c) is the mask, and (d) is a perturbation to which the mask has been applied. The perturbation concentrates on the central region, as the background is already quite similar between the images.

3.2. Regional masking

Currently, perturbations are evenly applied across the entire image. No matter if low or high frequency, the orthogonal step of the Boundary Attack perturbs all pixels nearly equally (when averaged over a large number of samples). This seems to be a waste – could the attack benefit from limiting the perturbation to specific regions?

The Boundary Attack is an interpolation from an image of the target class towards the image under attack. In some regions, these images might already be quite similar, while being very different in others. Intuitively, we would want an attack to take larger steps in those regions where the difference is high. We are also reluctant to perturb regions that are already similar, as any such distortion will have to be undone in a later step.

It turns out that this is an ideal way to reduce the search space. We can simply create an image mask from the per-pixel difference of the adversarial and original image (see Figure 3):

$$M = |X_{adv} - X_{orig}| \tag{1}$$

At each step, we recalculate this mask based on the current position and apply it element-wise to the previously sampled orthogonal perturbation η^k:

$$\eta^k_{biased} = M \odot \eta^k;\ \eta^k_{biased} = \frac{\eta^k_{biased}}{\|\eta^k_{biased}\|} \tag{2}$$

In this way, the distortion of those pixels that have a high difference is amplified and that of similar pixels dampened, while the magnitude of the perturbation vector stays the same.

This reduces the search space of the attack and therefore increases its efficiency – if one assumes our intuition about regional masking to be correct. We implement this masking strategy as a proof of concept and our evaluation in Section 4 shows that it indeed improves efficiency by a significant amount.

Other masks. An attacker might wish to engineer masks from other knowledge they possess about the image contents. For example, it could be worthwhile to concentrate the perturbation on the most salient features of the target class, reducing the search space to only the most vital dimensions. Such "focused" black-box perturbations could hold much promise, and we aim to investigate them in the future.

3.3. Gradients from surrogate models

What other source of information contains strong hints about directions that are likely to point to an adversarial region? Naturally – gradients from surrogate models. Transfer attacks have been shown to be extremely powerful (albeit brittle) [18], so it should be useful to exploit surrogates whenever they are available.

Arguably, the main weakness of transfer attacks is that they fail when the decision boundary of the surrogate model does not closely match the defender's. However, even when this is the case, the boundary may still be reasonably nearby. Based on this intuition, some approaches extend gradient-based attacks with limited regional sampling [1]. Here, we do exactly the opposite and extend a sampling-based attack with limited gradient information. This has the significant advantage that, in the case of low transferability, our method merely experiences a slowdown where typical transfer attacks completely fail.

Our method works as follows:

- An adversarial gradient from a surrogate model is calculated. Since the current position is already adversarial, it can be helpful to move a small distance towards the original image first, making sure to calculate the gradient from inside a non-adversarial region.
- The gradient usually points away from the original image, therefore we project it orthogonally to the source direction, as shown in Figure 2c.
- This projection is on the same hyperplane as the candidates for the orthogonal step. We can now bias the candidate perturbations toward the projected gradient by any method of our choosing. Provided all vectors are normalized, we opt for simple addition:

$$\eta^k_{biased} = (1 - w) \cdot \eta^k + w \cdot \eta^k_{PG} \tag{3}$$

- w controls the strength of the bias and is a hyperparameter that should be tuned according to the performance of our substitute model. High values for w should be used when transferability is high, and vice versa. Were we to choose the maximum value, $w = 1$, then the orthogonal step would be equivalent to an iteration of the PGD attack. In our experience, $w \leq 0.5$ generally leads to good performance.

As a result, samples concentrate in the vicinity of the projected gradient, but still cover the rest of the search space (albeit with lower resolution). In this way, substitute models are purely optional to our attack instead of forming the central part. It should be noted though that at least *some* measure of transferability should exist. Otherwise, the gradient will point in a bogus direction and using a high value for w would reduce efficiency instead of improving it.

For the time being, this does not pose a major problem. To the best of our knowledge, no strategies exist that successfully eliminate transferability altogether. As we go on to show in Section 4, even surrogate models that are too weak for direct transfer attacks can be used in our framework.

3.4. Concurrent work

Our work is concurrent with Ilyas et al. [10], who introduce a bandit optimization framework that incorporates prior information in order to increase query efficiency. While their approach differs from ours, it is motivated by the same intuition - domain knowledge can be used to speed up optimization. We note that the data-dependent prior they propose is essentially a low-frequency bias not unlike our own. They also introduce a time-dependent prior, which could benefit our work in the future.

Low-frequency perturbations have also recently been described by Guo et al. [8]. They decompose random perturbations with the Discrete Cosine Transform, and then remove high frequencies from the spectrum. We expect these patterns to be very similar to those produced by our Perlin bias.

4. Evaluation

We evaluate our approach against an ImageNet classifier and perform an ablation study to determine the effectiveness of each bias. We also compare our results to a range of recently proposed black-box attacks. Finally, we mount an attack against the Google Cloud Vision API and show that our approach can be efficiently deployed against real-world commercial systems.

Appendix A shows a range of interesting examples produced by our attacks, Appendix B contains a listing of hyperparameters and Appendix C describes our winning submission to the NeurIPS 2018 Adversarial Vision Challenge in detail.

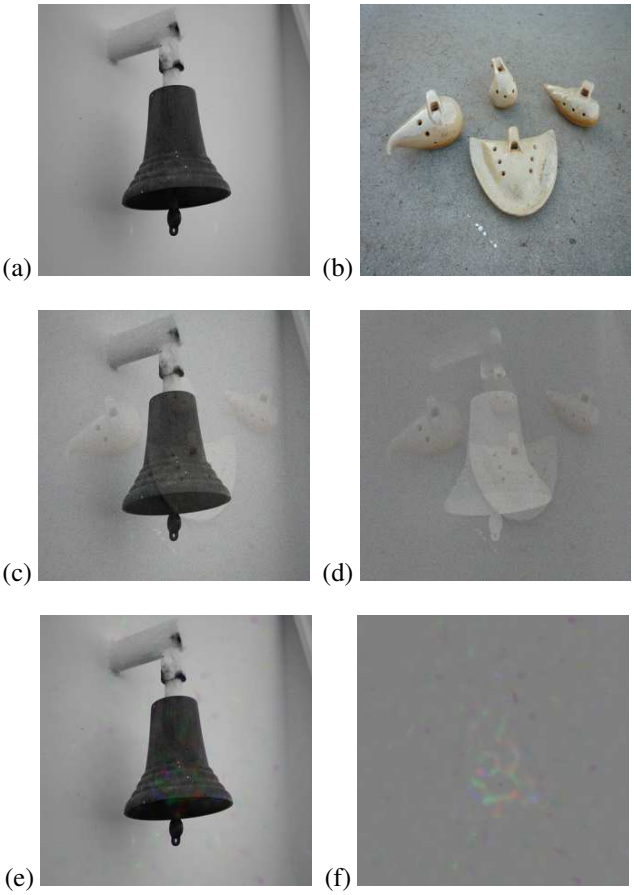


Figure 4. Targeted attack on ImageNet using 15000 queries. (a) shows the original image (bell) and (b) an image of the target class (ocarina). (c) is an adversarial example generated by the original Boundary Attack ($d_{\ell^2} = 18.52$), with (d) showing the difference to the original image. (e) and (f) show the biased Boundary Attack with all biases enabled ($d_{\ell^2} = 4.78$)

4.1. ImageNet

ImageNet consists of images with 299x299 color pixels and has 1000 classes. We run our attacks against a pre-trained InceptionV3 network [16], which achieves 78% top-1 accuracy.

Evaluation. We create an evaluation set by randomly selecting 1000 images from the ImageNet validation set while fixing a random target label for each image. We then proceed to run each attack for up to 15000 queries and measure the *success rate* over all examples.

Success criteria. The Boundary Attack always starts with an image of the adversarial class, therefore success must be characterized by low distance to the original image. For this, we define a threshold on the ℓ^2-norm of the adversarial perturbation and register success when the distance is below. Since some of the methods we compare [9] use ℓ^∞

Active biases			Success rate vs number of queries						Median queries
Perlin	Mask	Surrogate	500	1000	2500	5000	10000	15000	until success
NO	NO	NO	0.01	0.03	0.10	0.17	0.25	0.33	-
YES	NO	NO	0.03	0.08	0.18	0.31	0.47	0.57	10826
NO	YES	NO	0.03	0.04	0.13	0.22	0.35	0.44	-
YES	YES	NO	0.04	0.09	0.25	0.46	**0.72**	0.80	5485
NO	NO	YES	0.02	0.06	0.10	0.19	0.29	0.38	-
YES	NO	YES	**0.05**	0.08	0.17	0.32	0.50	0.60	10277
NO	YES	YES	0.03	0.06	0.15	0.26	0.38	0.49	-
YES	YES	YES	0.04	**0.10**	**0.29**	**0.48**	0.69	**0.85**	**5432**

Table 1. Ablation study of biases on ImageNet (targeted attack). The Perlin bias has the strongest effect, followed by the mask bias and finally the surrogate bias. Each bias improves efficiency on its own, and the combination of all biases delivers the strongest performance. *Note: we only report the median for success rates over 0.5.*

	Success rate vs number of queries					
Method	500	1000	2500	5000	10000	15000
Ilyas et al. [9] (label-only)	0.00	0.00	0.00	0.00	0.00	0.00
Cheng et al. [6]	0.04	0.04	0.04	0.04	0.07	0.07
Madry et al. [13] (PGD transfer attack)	**0.07**	0.08	0.08	0.09	0.09	0.09
Brendel et al. [3] (unbiased Boundary Attack)	0.01	0.03	0.10	0.17	0.25	0.33
Ours (Biased Boundary Attack)	0.04	**0.10**	**0.29**	**0.48**	**0.69**	**0.85**

Table 2. Comparison with recently proposed label-only attacks. We use the original code provided by the respective authors and run all methods with the same data and targets. The biased Boundary Attack (same as in Table 1) outperforms all other methods. Note that in the case of Ilyas et al.[9], the number of required queries is so large that the attack never achieves success in the range we consider.

distance exclusively, we set the ℓ^2-threshold to 25.89. This corresponds to a worst-case ℓ^∞-distortion of 0.05 if one assumes all pixels to be maximally perturbed.

Initialization. We search the ImageNet validation set for an image of the target class, and pick the one that is closest to the one being attacked. From there, we perform a binary line search to find the decision boundary. This is typically done in less than 10 queries. We run all attacks with the same starting points.

Surrogate model. We use a pre-trained Inception-Resnet-v2 model [15] for the gradient bias. This model is not adversarially trained and, as Table 2 shows, performs poorly in a PGD transfer attack. We intentionally use this model to demonstrate that our approach can effectively use pre-trained surrogates without any need for modification.

Hyperparameters. See Appendix B.

4.1.1 Ablation study of biases

We first evaluate all three biases and their combinations. Table 1 shows the result: it is apparent that each of the biases increases the efficiency of the attack. The largest boost is obtained by the Perlin bias, followed by the mask bias, and finally the surrogate gradient bias.

The latter has a rather small effect, which is probably due to the fact that our surrogate model is too weak. But still, we are able to use what little transferability there is instead of slowing down (or failing like a transfer attack would). At this point, it would be interesting to see whether our method could profit even more from better surrogates. We consider this a direction for future work.

It is also noteworthy that all biases can be combined, and that they do not interfere with each other. When all three biases are active, we reach 85% success after only 15000 queries. Figure 4 shows an example of this drastic improvement. This is our strongest attack, which we now compare against other state-of-the-art methods.

4.1.2 Comparison to state of the art

We go on to benchmark our method against a range of recently proposed attacks for our setting. We use publicly available code, together with the hyperparameters recommended by the authors. We modify the implementations to use our evaluation set, therefore all attacks are run on the same 1000 images and use the same target labels as well as starting points (where applicable). Table 2 shows the results.

Ilyas et al. [9] propose a label-only version of their gradient estimation attack. It works in our setting, but at greatly reduced efficiency. We note that it does not produce an adversarial example within 15000 queries and that no run succeeds before 276000 queries, with a median of 2.48 million queries required. This is in line with their published results. Their other attacks use confidence scores, and therefore require a setting that is considerably easier. Even then, our

median number of queries is decisively lower than the one reported by them (5432 versus 11550).

Similarly, **Cheng et al. [6]** re-frame the setting as a real-valued optimization problem. In general, they report higher efficiency than the Boundary Attack, which is not confirmed in the setting of our experiment. We used their publicly available source code with recommended hyperparameters.

Finally, we perform an iterative PGD transfer attack as described by **Madry et al.** [13]. Our results show that its performance is hit-and-miss: when a transfer succeeds it does so very early, but in most cases it never succeeds. Clearly this attack requires a stronger surrogate model.

Interestingly enough, the performance we obtain for the original Boundary Attack (without biases) seems higher than that observed in previous work [3, 6]. This may be due to our initialization method or our choice of hyperparameters, which we list in Appendix B. In any case, our evaluation shows that the biased Boundary Attack decidedly outperforms all other attacks in a label-only setting.

4.2. Google Cloud Vision API

To show that our method is effective even against black boxes with unknown labels, we conduct an attack against the Google Cloud Vision API. This is significantly harder than attacking ImageNet, since the exact classes are unknown. However, we also note that Google Cloud Vision has a very high number of near-redundant class labels. Do we really need to focus on one label alone?

Free-form attacks. We have argued earlier that untargeted attacks with many redundant classes are not truly adversarial. However, the opposite is also true: to achieve an adversarial effect, it is not always necessary to target one specific class. Rather, the same effect could be achieved by targeting a group of classes – if we want to label a dog as a cat, we can take the union of all cat breeds to achieve the desired effect.

To be more precise, we can formulate *any* adversarial criterion, as long as it is a boolean function of the model output. Decision-based attacks make this very simple: if we consider this function to be part of the black box, we can simply treat it like any other model and run our attack on its output.

4.2.1 Turning a person into a bear

Consider, for example, a targeted attack to turn a person into a bear. Instead of using the label ”bear”, we perform a string comparison on the top-1 label and check for the occurrence of ”bear”. This extends the attack to labels like ”grizzly bear”, ”brown bear”, etc. and keeps it from getting stuck whenever one of these labels appears in top-1 position. We also add another condition for good measure: the words

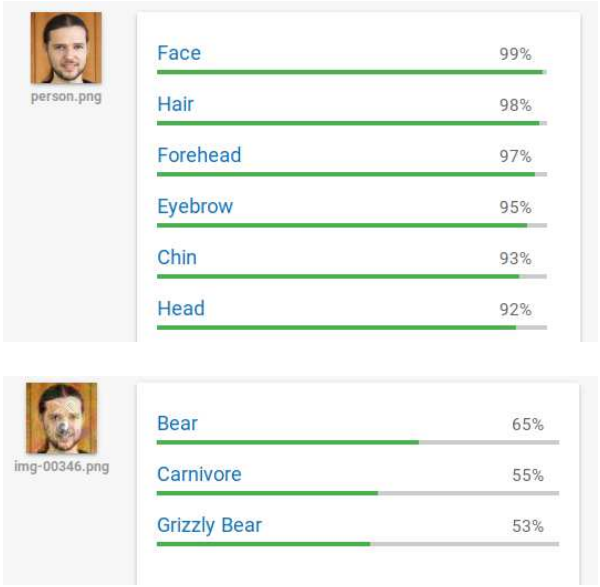


Figure 5. Adversarial image (target ”bear”), classified by Google Cloud Vision after 346 queries. Confidence scores are displayed, but not used by the attack. No label hinting at a person is left (the prediction vector contains only 3 labels).

”face”, ”facial expression”, ”skin”, ”person” must not appear in any of the output labels.

Figure 5 shows the result: after only 346 iterations, our attack produces a perturbation that is still visible, but small enough to fool an unsuspecting person.

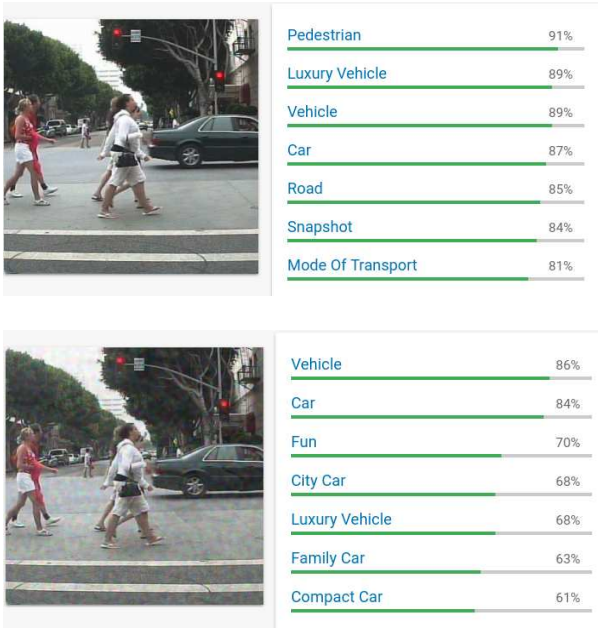


Figure 6. Adversarial image (pedestrians have been removed), classified by Google Cloud Vision after exactly 1000 queries. Confidence scores are displayed, but not used by the attack. Perhaps interestingly, our adversarial pattern is classified as ”Fun”.

4.2.2 Making pedestrians disappear

The adversarial criterion can also be formulated as a top-k untargeted attack on multiple labels. Consider a potentially safety-critical scenario, where the goal is to make the model oblivious to pedestrians. For this, we simply formulate the condition so that the string "pedestrian" does not appear in the prediction vector, and that related labels such as "person, walking, head, clothes" are also absent.

We obtain Figure 6 after exactly 1000 queries. Again, the perturbation is already small enough to fool an unsuspecting observer. An attack with such a low number of queries can be performed on virtually any device – even mobile – in a matter of minutes, and is only limited by the latency of the API under attack.

5. Conclusion

We have shown that decision-based black-box attacks can be greatly sped up with prior knowledge. The Boundary Attack can be interpreted as a biased sampling framework where one merely needs to modify the distribution from which samples are drawn.

Within this framework, we have proposed three priors that are partially motivated by intuitions about the nature of image classification, and partially by a desire to connect research directions in the field of black-box attacks. Consider the surrogate gradient bias: by itself, it does not yield a substantial improvement. However, the observation that we are able to draw even a small benefit from surrogates that otherwise show near-zero transferability seems very promising for future work. We aim to study this effect of partial transferability in more detail and hope to uncover some of its underlying properties.

And it does not end here - we have discussed only three priors for biased sampling, but there is much more domain knowledge that has not yet found its way into adversarial attacks. Other perturbation patterns, spatial transforms, adversarial blending strategies, or even intuitions about semantic features of the target class could all be integrated in a similar fashion.

With the biased Boundary Attack, we have outlined a basic framework into which a broad range of knowledge can be incorporated. Our implementation significantly outperforms the previous state of the art in black-box label-only attacks, which is one of the most difficult settings currently considered. Our methods can be used to craft convincing results after very few iterations, and the threat of black-box adversarial examples becomes more realistic than ever before.

Acknowledgements

This research was funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy, as part of the fortiss project "Dependable AI".

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018. 2, 4

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018. 3

[3] 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. 1, 2, 6, 7

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 1

[5] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pages 15–26, 2017. 1, 2

[6] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018. 6, 7

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2

[8] Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. *arXiv preprint arXiv:1809.08758*, 2018. 5

[9] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018. 1, 2, 5, 6

[10] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018. 5

[11] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, Alan Yuille, Sangxia Huang, Yao Zhao, Yuzhe Zhao, Zhonglin Han, Junjiajia Long, Yerkebulan Berdibekov, Takuya Akiba, Seiya Tokui, and Motoki Abe. Adversarial Attacks and Defences Competition. *arXiv preprint arXiv:1804.00097*, 2018. 2, 3

[12] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. *International Conference on Learning Representations*, 2018. 1, 2, 6, 7

[14] Ken Perlin. An image synthesizer. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, pages 287–296, New York, NY, USA, 1985. ACM. 3

[15] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2016. 6

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015. 5

[17] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 2

[18] 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. 2, 4

[19] Florian Tramèr, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017. 2

[20] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *arXiv preprint arXiv:1805.11770*, 2018. 2

[21] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 3