Towards Lightweight Black-Box Attacks against Deep Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Black-box attacks can generate adversarial examples without accessing the parameters of target model, largely exacerbating the threats of deployed deep neural networks (DNNs). However, previous works state that black-box attacks fail to mislead target models when their training data and outputs are inaccessible. In this work, we argue that black-box attacks can pose practical attacks in this extremely restrictive scenario where only several test samples are available. Specifically, we find that attacking the shallow layers of DNNs trained on a few test samples can generate powerful adversarial examples. As only a few samples are required, we refer to these attacks as *lightweight* black-box attacks. The main challenge to promoting lightweight attacks is to mitigate the adverse impact caused by the approximation error of shallow layers. As it is hard to mitigate the approximation error with few available samples, we propose Error TransFormer (ETF) for lightweight attacks. Namely, ETF transforms the approximation error in the parameter space into a perturbation in the feature space and alleviates the error by disturbing features. In experiments, lightweight black-box attacks with the proposed ETF achieve surprising results. For example, even if only 1 sample per category available, the success attack rate in lightweight black-box attacks is only about 3% lower than that of the black-box attacks with complete training data.

1 Introduction

Black-box attacks [7, 36, 6] can mount successful attacks without accessing the parameters of deep neural networks (DNNs), posing great challenges to deep learning in safety-critical situations. Existing black-box attack methods assume that the training data and/or the outputs of target models are available to adversaries. For scenarios where training data are available, adversarial examples can be generated by attacking surrogate models constructed for approximate target models. Using the target models' outputs for estimating gradients is another practical approach to crafting adversarial examples. In general, due to the low dependency on target model information, the black-box attacks raise realistic threats to real-world deep learning systems, including semantic segmentation [3, 22], object detection [9, 28, 27], and automatic driving [35].

However, these assumptions can be violated in many practical scenarios [4, 29], where both the training data and the outputs of target models are inaccessible. This realistic attack scenario is introduced in [4], generally termed as the *no-box setting*. Under such a restrictive setting, Li et al. [20] states that existing black-box attacks can fail to mislead target models, as the information previously considered essential in mounting black-box attacks is now inaccessible. Therein, the failure of existing black-box attacks is not surprising. The reason is that constructing reliable surrogate models typically relies on target models' training data, which is inaccessible in such a restrictive setting with only a few test samples available.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In this work, we aim to reveal the potential threats of black-box attacks in the no-box setting, as mounting black-box attacks with highly limited accessible information leaves many models exposed to attack. Specifically, we challenge the previous believes by raising the following question: *Can black-box attacks success in the no-box setting?* The doubts about the potential threats of the black-box attack are not groundless. Specifically, the potential threats of black-box attacks stem from two facts: (a) adversarial examples can be generated by perturbing representations at shallow layers of DNNs [18, 17, 14]; (b) regarding the representation of shallow layers, there do not exist critical differences between those models learned from a few data and that of the whole training data. [1].

Building upon the above facts, it is actually possible that an adversary can successfully mount attacks using limited samples, as powerful adversarial examples can be generated by attacking the shallow layers of DNNs trained on few samples. Concretely, the adversary can leverage available samples to construct a surrogate model to approximate the shallow layers of target models within acceptable errors. Consequently, adversarial examples can be generated by perturbing the features obtained from the shallow layers of the constructed surrogate model. As merely a few samples are required for attacking, we refer to black-box attacks in the no-box setting as *lightweight black-box attacks*.

Intuitively, the closer the surrogate model is to the target model, the higher the attack success rate the crafted adversarial examples have [19]. If the approximation error of shallow layers is alleviated, the lightweight black-box attack can be as powerful as black-box attacks with complete training data. Hence, the main challenge to mounting a powerful lightweight attack is to mitigate the adverse impact caused by the approximation error of shallow layers.

However, it is challenging to mitigate the approximation error, especially when the number of available samples is limited. Fortunately, it is straightforward to identify which kind of perturbations are preferred: if perturbations applied to a feature contribute to fooling the surrogate model, the perturbation is preferred. Therefore, bridging the connection between the parameter space and the feature space is the key to mitigating the approximation error. Accordingly, we propose transforming the approximation error in the parameter space as the perturbation in the feature space, dubbed as Error TransFormer (ETF). Namely, ETF transforms the worst-case approximation error to the worst-case feature perturbation, leading to a min-max scheme that generates adversarial examples under the worst feature perturbations. We verify the attack success rate of lightweight black-box attacks 7 models trained on the ImageNet dataset [30], and find that existing attack methods are much powerful than previous claimed. Moreover, the performance of lightweight black-box attacks can be further promoted by the proposed ETF, i.e., the attack success rate is only 3% lower than that achieved by black-box attacks having complete target model's training data.

2 Related works

2.1 Adversarial attack

According to the amount of accessible information, existing adversarial attacks can be roughly divided into two categories: white-box attacks and black-box attacks. White-box attacks [10, 25, 24, 2] assume that the target model is transparent to adversaries, i.e., adversaries can access all information about the target model. Nevertheless, the assumption that all information is available to adversaries can be violated in many practical scenarios [6]. Hence, black-box attacks [24, 25, 36, 7] are proposed and applied to the scenario where relatively limited information about the target model is accessible, i.e., only a certain number of model queries or the training data of target tasks are available. Utilizing the training data or queries, adversaries generate adversarial examples on the surrogate model constructed for approximate target models.

Among these black-box attacks, intermediate-level attacks [31, 18, 14] are widely explored to improve the adversarial transferability. The core idea of this attack is based on the empirical observation that well-trained models' intermediate features are transferable [37]. Hence, Inkawhich et al. [18] propose to perturb the feature space of neural networks to create more transferable adversarial examples. Besides, these attacks provides an interesting empirical conclusion that disturbing the shallow layers of DNNs can also generate adversarial examples.

Existing black-box attacks assume that the large-scale training data of target models and the feedback from querying target models are accessible, but these assumptions can be violated in many scenarios, such as no-box setting [4]. Li et al. [20] state that existing black-box attacks cannot be successfully

mounted because existing DNNs require large-scale training data for generalization. Given only small-size data available, obtaining a surrogate model with strong generalization is challenging. Hence, Li et al. propose replacing black-box attacks with their proposed no-box attack [20], where they train a classical auto-encoder model instead of the supervised classification model due to the constraint of a small-scale dataset. Concretely, they train 20 auto-encoders for each category and generate adversarial examples by attacking these auto-encoders. Instead of proposing a new attack, which is time- and computational-consuming.

2.2 Approximation of Shallow Networks

A recent study [38] confirms the intuition that shallow layers in DNNs can be seen as low-level feature extractors. Motivated by this observation, Asano et al. [1] design a method to explore the information in every layer of DNNs. Asano et al. [1] train a supervision model on ImageNet LSVRC-12, and a self-supervision model on a small-scale dataset. Then, they apply linear probes [40] to all intermediate layers of networks, where a linear classifier is trained on the top of pre-trained and fixed feature representations. In this way, they evaluate the quality of the representation learned at different depths of the networks. The results show that, given heavily synthetic transformations, the shallow layers of DNNs learned from a few images can approximate that of DNNs trained on millions of images. According to the intriguing empirical observation, it is possible to construct a surrogate model with limited samples, where its shallow layers are similar to that of the target models.

3 Preliminaries

This section defines the no-box threat model and the lightweight black-box attack.

3.1 No-Box Threat Model

The no-box setting [4] denotes a threat model where available information about the target model is extremely limited, making it challenging to generate adversarial examples. According to the description in [4], the no-box threat model can be defined as follows.

Definition 1 (No-Box Threat Model) *A threat model is called the no-box threat model if the adversary in this scenario is not allowed to access the target model's training data and outputs, so it has and only has some samples for attacking that can be correctly predicted with high probability by the target model.*

The definition is consistent with the no-box setting introduced in [4], where the target model's training data and outputs are inaccessible to the adversary. Moreover, the available samples used for generating adversarial examples are expected to be limited. This is because if the number of samples is similar to or even larger than the number of samples used for training target models, the adversary can leverage these samples to construct a surrogate model similar to target models, making the condition of inaccessible training data meaningless. It is intuitive that target models can correctly predict the labels of these available samples with high probability. For example, it is meaningless to perform attacks using samples misclassified by the target model.

3.2 Lightweight Black-Box Attack

Definition 2 (Lightweight Black-Box Attack) *An attack is called the lightweight black-box attack if an adversary aims to perform an attack in the context of a no-box threat model.*

The definition of the lightweight black-box attack shows that lightweight attack is a special kind of black-box attacks. The extremely limited accessible information distinguishes it from other black-box attacks. For example, query-efficient black-box attacks require the feedback information of target models [15, 16], while lightweight attacks do not. To our best knowledge, how to perform a lightweight black-box attack is still lacking in the literature. For example, Li et al. [20] claim that black-box attacks fail to fool target models when their training data and outputs are inaccessible, i.e., in the context of the no-box threat model.

4 Approach

Generating adversarial examples in the lightweight black-box attacks requires constructing surrogate models. As the training procedure requires limited samples, we refer to these models as lightweight surrogate models. After constructing the surrogate model, we can generate adversarial examples by attacking the model in the feature space, i.e., feature space perturbation. The reason is that using limited samples can merely approximate the shallow layers within acceptable errors. To further promote the performance of lightweight attacks, we propose the Error TransFormer (ETF) to alleviate to adverse impact caused by approximation error.

4.1 Lightweight Surrogate Model

Surrogate models are widely used in black-box attack, as adversarial examples are usually generated by attack surrogate models when querying target models is forbidden. Since the lightweight black-box attack is a special kind of black-box attacks, performing attack requires constructing surrogate models. We call surrogate models employed in lightweight black-box attacks lightweight surrogate models, considering that the number of samples are usually limited.

Similar to existing black-box attacks, we assume that the label information of samples used for generating adversarial examples are accessible. In this scenario, the lightweight surrogate model is trained for a classification task. Specifically, the lightweight model is realized by training a classification DNN equipped via conventional supervised learning, using cross-entropy loss $\ell(\cdot,\cdot)$:

$$\min_{w} \mathbb{E}_{(x,y)\sim\hat{D}}\ \ell(f(x;w),y), \tag{1}$$

where w stands for the parameters of surrogate models f, (x,y) sampled from the dataset $\hat{D}$ denotes the pair of natural sample and its label y.

In black-box attacks, adversary only leverages the shallow layers of DNNs, leading to a straightforward approach to constructing lightweight surrogate models. Specifically, we can train the lightweight surrogate model in a contrastive manner, where the supervised information is no longer necessary:

$$\min_{w}\ \mathbb{E}_{x\sim\hat{D}} - \log \frac{\sigma(f(x;w), f(\mathcal{T}(x);w))}{\mathbb{E}_{x'\sim\hat{D}\backslash x}\ \sigma(f(x;w), f(\mathcal{T}(x');w))}, \tag{2}$$

where σ is a specific similarity metric, e.g., cosine similarity, $\mathcal{T}$ denotes the data transformation operation, and $\hat{D}\backslash x$ means that x is removed from the dataset $\hat{D}$.

4.2 Feature Space Perturbation

Built upon the previous empirical observation, using limited samples can approximate the shallow layers of DNNs within acceptable errors. This suggests that layers other than shallow layers can cause large approximation error. Accordingly, merely the shallow layers of the learned lightweight surrogate models are used for generating adversarial examples, which is similar to the feature space attacks. For brevity, we define the shallow layers as φ and denote the feature of x obtained by φ as $\varphi(x;w)$.

In feature space attacks, guide images are usually required for generating adversarial examples. A natural image used for generating adversarial example x_{adv} is usually called source image. Given a source image x_s, we generate an adversarial example x_{adv} by perturbing x_s such that its feature is similar to that of a guide image x_g, where the labels of these two images are different. Specifically, the adversarial examples are generated as follows:

$$x_{adv} = \arg \min_{\|x'-x\|_p \le \epsilon} d(\varphi(x_g;w), \varphi(x';w)), \tag{3}$$

where d stands for a specific distance and ϵ controls the strength of adversarial perturbation.

4.3 Error Transformer

Although applying feature space attacks to lightweight surrogate models can generate transferable adversarial examples, the approximation error will cause adverse impact to the mounting attacks.

The reason is that large dissimilarity between the surrogate models and the target models reduces the attack success rate. Therefore, the main challenge to mounting a powerful lightweight attack is to mitigate the adverse impact caused by the approximation error. However, available samples are usually limited in the no-box threat model, making it hard to alleviate the approximation error.

To address the challenge, we propose to transform the approximation error in the parameter space as the perturbation in the feature space. The inspiration is borrowed from the feature space attack. Specifically, it is clear to identify the preferred feature perturbations, i.e., we prefer the perturbations than can contribute to fooling the surrogate model. Similarly, it is also straightforward to define the "bad" perturbations, which hinder the generation of adversarial examples. Therefore, connecting the parameter space and feature space makes it possible to mitigate the adverse impact caused by approximation error. To bridge the connection, we introduce the following key identity, which is also used in [26]:

$$\varphi(x; \{w^1 + w^1 A\} \cup \{w \backslash w^1\}) = g((w^1 + w^1 A)x; w \backslash w^1) = g(w^1(x + Ax); w \backslash w^1) = \varphi(x + Ax; w), \quad (4)$$

where w^1 stands for the first layer parameters of model φ, A is a transformation matrix used for perturbing w^1, $\{w^1 + w^1 A\} \cup \{w \backslash w^1\}$ means that the first layer's parameters are perturbed and the other layers' parameters keep the same, and g is the function parameterized with $w \backslash w^1$ used for processing the first layer's outputs. According to Eq. 4.3, we can transform a perturbation in the parameter space as the perturbation in the data space, i.e., $\varphi(x; \{w^1 + w^1 A\} \cup \{w \backslash w^1\}) = \varphi(x + Ax; w)$. Similarly, we can apply the transformation operation to the hidden layers, see Appendix for more details.

Built upon the connection in Eq. 4.3, we can transform the first layer parameters' discrepancy between the target model and the surrogate model. Taking the first layer as an example, let w_t be the parameters of the target model and w_t^1 be its first layer parameters, where the other parameters are the same as the surrogate model. Assume that there exists a transformation matrix A applied to the first layer parameters of the surrogate model such that $w_t^1 = w^1 + w^1 A$, we have:

$$\varphi(x; w_t^1 \cup \{w_t \backslash w_t^1\}) = \varphi(x; \{w^1 + w^1 A\} \cup \{w_t \backslash w_t^1\}) = \varphi(x + Ax; \{w^1\} \cup \{w_t \backslash w_t^1\}) = \varphi(x + Ax; w). \quad (5)$$

The identity in Eq. 5 shows that we can find a perturbation in the data space to alleviate the difference in the first layer parameters between these two models. Built upon the transformation, we can transform the worst approximation error to the worst perturbation in the data space. Therefore, we can employ a min-max strategy to mitigate the adverse impact of approximation error through generating adversarial examples under the worst perturbation. Thus, lightweight black-box attacks with ETF generate adversarial examples as follows (equation applying ETF to other layers can be found in Appendix):

$$x_{adv} = \arg \min_{\|x' - x\|_p \leq \epsilon} \max_{\|\Delta_s\|_p \leq \tau,\ \|\Delta_t\|_p \leq \tau} d(\varphi(x_g + \Delta_g; w), \varphi(x' + \Delta_s; w)), \quad (6)$$

where $\|\cdot\|_p$ is the ℓ_p-norm, ϵ and τ control the strength of perturbations, Δ_g and Δ_t are the data space perturbations for mitigating the approximation error in parameter space. In this way, we reduce the error in parameter space to approximate the target model.

5 Experiments

In this section, experimental setups and implementation details are introduced in Section 5.1. In Section 5.2, lightweight black-box attacks through ETF shows consistent effectiveness on both normally and adversarially trained models. Further analysis and ablation studies are provided in Section 5.3

5.1 Experimental Setup

Models Architectures. All surrogate models are based on the ResNet-18 [11]. Following the ordinary settings in the black-box attack, the general surrogate models are trained through the whole training set of ImageNet [30]. On the contrary, the lightweight surrogate models adopt only $1,000$ images sampled from the validation set of ImageNet, considering that images in the training set are probably inaccessible in practice [20]. In addition, the architecture of surrogate models is modified

Table 1: The accuracy of 7 normally trained target models evaluated on 1000 adversarial examples generated by lightweight black-box attacks or existing black-box attacks, under $\epsilon < 0.1$. The Shallow-(PGD, MI, DI, TI) mean applying PGD, MI, DI and TI to the shallow layers of the model. Deep-(PGD, MI, DI and TI) mean applying PGD, MI, DI and TI to the model's output. EFT-(PGD, MI, DI and TI) mean applying ETF combined with PGD, MI, DI or TI to the shallow layers. (The lower, the better)

Model	VGG19	Inception [33]	RN152 v3[34]	DenseNet [11]	SENet [13]	WRN [12]	MobileNet [39]	Average v3[32]
Clean	67.43%	64.36%	74.21%	73.34%	51.28%	73.22%	65.06%	66.99%
Deep-PGD	49.01%	52.26%	60.71%	57.92%	27.94%	60.18%	44.20%	50.31%
Deep-MI	38.92%	42.37%	49.53%	49.06%	19.44%	49.11%	33.46%	40.69%
Deep-DI	43.34%	43.13%	53.78%	55.41%	23.53%	51.77%	38.14%	44.15%
Deep-TI	49.46%	49.64%	58.89%	58.75%	26.19%	56.31%	44.02%	49.03%
Shallow-MI	22.62%	30.83%	34.05%	35.74%	12.31%	29.98%	17.72%	26.17%
Shallow-DI	22.14%	29.78%	35.51%	35.79%	8.99%	30.61%	16.88%	25.67%
Shallow-TI	21.82%	28.54%	34.78%	34.71%	7.96%	30.14%	15.77%	24.81%
Shallow-PGD	22.93%	31.07%	34.71%	36.20%	13.08%	32.16%	16.65%	26.69%
ETF-PGD	14.11%	20.22%	24.20%	24.74%	6.96%	20.73%	10.66%	17.37%
ETF-MI	15.32%	19.97%	26.25%	28.10%	7.02%	22.21%	12.23%	18.72%
ETF-DI	14.77%	20.63%	23.71%	25.70%	7.23%	20.22%	11.53%	17.68%
ETF-TI	15.45%	18.03%	22.63%	24.20%	6.94%	21.53%	12.88%	17.38%
Deep*-PGD	12.43%	28.15%	16.54%	12.61%	7.09%	13.33%	9.64%	14.25%
Deep*-MI	11.77%	25.14%	18.10%	13.72%	4.26%	14.61%	8.30%	13.70%
Deep*-DI	7.61%	18.17%	8.23%	9.90%	6.66%	9.72%	7.91%	9.74%
Deep*-TI	9.55%	23.48%	13.51%	10.63%	6.46%	10.92%	9.55%	12.01%

Deep* refers to the attacks mounted in the model trained on the large-scale training data.

to avoid overfitting[1]. Please refer to Table 5 in the Appendix A for the detailed information on the lightweight surrogate model and the general surrogate model.

Regarding the target models, various model architectures are selected for full comparison, including, VGG-19 [33], Inception v3 [34], ResNet-152 [11], DenseNet [13], SENet [12], wide ResNet (WRN) [39] and MobileNet v2 [32]. All these target models are well-trained on the training set of ImageNet.

Implement Details. The lightweight surrogate models are randomly initialized. The batch size is 128, the epoch is 500, and the initial learning rate is 0.4, linearly decreasing to 0.008. To better approximate the low-level feature of shallow layers, we refer to the self-supervision work [5] and apply various random augmentations to the data in each epoch for training. For mounting attacks, classic methods, e.g., PGD [24], MI [6], DI [36] and TI [7], are applied to all surrogate models. Unless otherwise specified, our attack is mounted on the first layer of ResNet-18 for lightweight surrogate models. The adversarial examples are generated by increasing the similarity between the source images and the guide images regarding their output features maps of shallow layers. Following previous works on intermediate-level attacks, the mean square error is used as the similarity metric, and apply different attacking methods mentioned before are adopted. The Top-1 prediction accuracy is adopted as the evaluation metric. The lower the classification accuracy, the better the attack success rate. All the experiments are run for 5 individual trials with different random seeds. Since the standard deviation is lower than 1%, we report only their average accuracy.

5.2 Lightweight Black-box Attacks for ImageNet classification

Results on normally trained models. The results of classification accuracy are shown in Table 1, which exhibits the evaluations for both lightweight and general surrogate models. The last four

[1]Considering the limited amount of data, we employ a network with a small model capacity to instantiate the feature extractor. In particular, ResNet-18 is simplified by reducing the number of blocks in each layer, i.e., only one block is used in each layer of ResNet-18.

rows show the ordinary results where the general surrogate models are trained through all data in training set. The black-box attacks are mounted based on the whole model, termed "Deep/Deep*", with average accuracies of around 10%. As the size of the training data is reduced to $1,000$, this conventional method obviously loses its aggressiveness. In specific, the averaged accuracies of 2-5 rows are among 40%-50%, revealing limited attacking capability compared to 66.99% of the clean accuracy. It demonstrates the necessity of large amount of data for classic black-box attacks. In the 6-9 rows, shallow layers without ETF are used to generate adversarial examples. The averaged accuracies of around 25% illustrates the less dependence of shallow networks on the amount of data. In the 10-13 rows, ETF is further combined based on shallow layers, and the averaged accuracies decrease to around 17%. The narrow gap between the proposed ETF, i.e., the lightweight surrogate model, and the general surrogate model indicates the potential capability of the black-box attack in the no-box settings.

Results on Adversarially Trained Models. Therein, models are trained through adversarial training [24], which is more persuasive to evaluate the effectiveness of attacking methods. Following [24], the target models are trained on the whole training set of ImageNet. During adversarial training, the adversarial examples are generated under $\epsilon = 0, 4/255, 8/255$, where ϵ means the constraint of L_∞ distance of the adversarial examples used for adversarial training relative to the original sample. Table 2 exhibits the attacking results mounted by ETF, Black-box attacks (general surrogate models with all training data avaliable), and No-box attacks. For simplicity, only the results of PGD attacks are reported. When $\epsilon = 0/255$, the target models are actually normal trained models. As analysed above, the black-box attacks shows the best attacking results, while ETF has a narrow gap with it. However, when $\epsilon = 4/255$ or $8/255$, ETF gradually exceeds the black-box attacks (29.13%26.14% vs. 48.11%/38.24%). It is interesting that black-box attacks are even worse than no-box attacks. Since adversarial training is adopted, the pattern of adversarial noise have a large probability to be seen during the training procedure. On the contrary, the ETF and no-box attacks generate the adversarial examples in different perspectives, and the superiority of ETF further indicates the effectiveness of the proposed method.

Table 2: The attack effect of different attack methods on the adversarial trained ResNet-50 [8]. Therein, ϵ refers to the constraint L_∞ in adversarial examples for adversarial training. The accuracy is evaluated on 1000 adversarial examples by different attacks.(the lower the better)

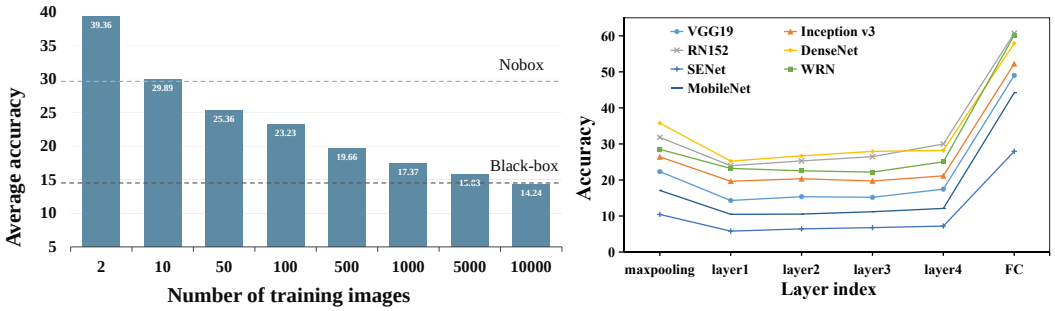
Adv_model	Clean	ETF(ours)	Black-box	No-box
$\epsilon = 0/255$	69.43%	16.97%	**8.20%**	24.53%
$\epsilon = 4/255$	55.62%	**29.13%**	48.11%	39.62%
$\epsilon = 8/255$	41.68%	**26.14%**	38.24%	35.87%

Table 3: The comparison between no-box attacks and lightweight black-box attacks mounted by different intermediate-level attacks. These methods use only 1000 samples to mount attacks and the pre-trained model is unavailable for adversaries. The best results are in **bold**. (The lower, the better)

Model	VGG19 [33]	Inception v3[34]	RN152 [11]	DenseNet [13]	SENet [12]	WRN [39]	MobileNet v3[32]	Average
Clean	67.43%	64.36%	74.21%	73.34%	51.28%	73.22%	65.06%	66.99%
No-box[20]	18.74%	33.68%	34.72%	26.06%	42.36%	33.16%	16.34%	29.29%
ILA[14]	20.13%	28.01%	35.72%	35.14%	9.16%	29.97%	14.31%	24.63%
AA[18]	22.76%	31.21%	36.67%	37.94%	7.72%	33.16%	18.17%	26.81%
ETF	**14.11%**	**20.22%**	**24.20%**	**24.74%**	**6.96%**	**20.73%**	**10.66%**	**17.37%**

5.3 Further Analysis and Ablation Study

Intermediate-level Perturbations. Since the lightweight black-box attacks are implemented mainly on intermediate features, two classic attacks mounted on intermediate features, i.e. ILA [14] and AA [18], are selected for experiments. Comparing the ILA and AA in Table 3 and Shallow-MI/DI/TI/PGD in Table 1, there is little difference between their average accuracies. It indicates those intermediate-level feature perturbation has little benefit to lightweight black-box attack. Because, the intermediate-level attacks rely more on the low-level features extracted by the lightweight model to



(a) (b)

Figure 1: (a) How the lightweight attack performance of our approach varies with the number of images used for training the surrogate model. (b) The influence of low-level feature extraction at different layers of ResNet-18 on lightweight black-box attack performance. (The lower, the better)

generate adversarial examples and weaken the role of high-level information in the attacks. However, he proposed ETF achieves better performance. On the one hand, ETF is not relied on the features from high-level layers, and on the other hand, the lightweight surrogate model are pushed to approximate a well-trained surrogate model.

Comparison with No-box Attacks [20]. No-box attack is the first work to explore how to mount attacks in the no-box setting. Considering the black-box attacks do not work in this setting, they do not attempt to design a surrogate model that approximates the target model as we do. Instead, they design the pretext task to train an auto-encoder per category to generate adversarial examples. However, the difference between the pretext tasks and the target tasks may significantly reduce the transferability of the crafted adversarial examples. In contrast, the proposed method train the lightweight model for the target task, which avoids the impact of different tasks on adversarial transferability. Hence, though one sample per category is available, we achieve better attack performance than no-box attacks, which is shown in Table 3.

Number of Training Samples The key difference between the lightweight and general black-box attacks mainly lies in the number of samples n used for training models. Therefore, we further explore how lightweight attack performance varies with n. We conduct the experiment on randomly selected n images from ImageNet validation. The experimental results are shown in Figure 1 (a). The average accuracy decreases as more samples are available, and the better attacking results is achieved. More data provides the opportunity for lightweight black-box attack to approximate the shallow layers of the target model more accurately. Especially when the number of samples equals $10,000$, the lightweight black-box attacks achieve the same performance as general black-box attacks which train the surrogate model on 1,200,000 images. More interestingly, the proposed method can still mount attacks even when only 2 or 10 images are available.

Different Layers of the Lightweight Surrogate Model As analyzed above, we utilize the first few layers of the lightweight model to approximate the shallow layers of the target model to mount attacks. Hence, it is interesting to study how the layers selection for lightweight models impacts the attack performance. We adopt different layers of ResNet-18 to generate adversarial examples utilizing ETF and attack the different target models. The results are summarized in Table 1 (b). It can be seen that low-level information in the first block of the model is sufficient for achieving effective attack results. With more block information on the model, the performance of the attack does not improve. In particular, the transferability of adversarial examples significantly decreases when we attack the full connection layer, which verifies our analysis that the high-level layer of the lightweight model cannot approximate the target model well.

6 Discussion

Though the no-box setting provides a weak assumption that a few samples and their labels are available, the realistic scenarios may be more complex. For example, adversaries have access to only data without supervision information due to security [21] or privacy [29] issues. In more extreme

Table 4: Mount lightweight black-box attack without supervision information or without any samples from the same domain, named Unsupervised and OOD.

Model	VGG19 [33]	Inception v3[34]	RN152 [11]	DenseNet [13]	SENet [12]	WRN [39]	MobileNet v3[32]	Average
Clean	67.43%	64.36%	74.21%	73.34%	51.28%	73.22%	65.06%	66.99%
Supervised	14.11%	20.22%	**24.20%**	24.74%	6.96%	**20.73%**	10.66%	17.37%
Unsupervised	15.54%	**19.16%**	26.27%	23.75%	7.66%	22.79%	11.43%	18.08%
OOD	**6.13%**	21.72%	25.44%	**21.89%**	**5.02%**	24.33%	**7.16%**	**15.96%**

case, e.g., autonomous vehicle [35], adversaries even have no access to the test samples from the distribution over which the model is trained. Instead, the adversaries have access to samples from a similar (but not exact) distribution (because the attacker knows the deployment environment). Hence, this section considers two more challenging attack scenarios in which we provide solutions to mount black-box attacks.

Insufficient supervision information. Adversaries have only a few unlabeled samples (e.g., $1,000$ unlabeled images). Since the proposed method focuses on low-level representations, we may not require the high-level semantic information captured by categorical labels. Hence, it may be feasible to train the unsupervised model on unlabeled data for attacks. We design an experiment for mounting attacks using $1,000$ unlabeled samples randomly selected from the ImageNet validation set to explore how the proposed ETF performs under more challenging circumstances with unlabeled data. Specifically, we adopt contrastive learning [5] to train a lightweight surrogate model on the small-scale unlabeled dataset and generate adversarial examples by attacking the trained lightweight surrogate model. The performance is shown in Table 4. We can see that without supervision information ETF still achieves similar performance as that under supervision.

Consideration of Out-of-distribution. As shown in Section 5.3, when only two samples are available, the crafted adversarial examples have weak transferability to mount attacks. This is because too little low-level information is available for lightweight models to approximate the target model well, even at shallow layers. However, adversaries can only get some samples from out-of-distribution data in real scenarios. Hence, we consider utilizing the model pre-trained on OOD samples to mount lightweight black-box attacks. The rationality lies in that only low-level information is required to mount lightweight black-box attacks and the acquisition of low-level information does not require the corresponding high-level category information. We load pre-trained ResNet-50 on STL-10 [23] dataset as the surrogate model. Without further fine-tuning, we experiment with two samples again, and the results are recorded in Table 4, named OOD. Interestingly, attacks mounted in the model trained on OOD achieve a higher attack success rate than those mounted in the lightweight surrogate model trained on $1,000$ samples. Plenty of data from STL-10 provide various low-level information, which is the main reason why OOD can also mount powerful attacks.

7 Conclusion

This paper proposes to perturb the shallow layers of a lightweight surrogate model to enable black-box attacks in the no-box setting. We demonstrate that black-box attacks can work well in the no-box setting by leveraging the lightweight model, even if it is trained with a few samples. The proposal is based on the empirical observations that the model trained with limited samples can approximate the shallow layers of the target models within acceptable errors. Therefore, to achieve better attack performance, this work proposes error transformer for lightweight attacks to decrease the approximation error between the lightweight surrogate model and the target model. Specifically, ETF transforms the approximation error in the parameter space into a perturbation in the feature space. We reduce approximation error by applying the worst feature perturbation. The experiment results show that the proposed ETF achieves a comparative attack success rate with only one image per category, which is only 3% lower than black-box attacks with complete training data of target model.

References

[1] Y. M. Asano, C. Rupprecht, and A. Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020.

[2] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pages 39–57. IEEE, 2017.

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell., 40(4):834–848, 2018.

[4] P. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In B. M. Thuraisingham, B. Biggio, D. M. Freeman, B. Miller, and A. Sinha, editors, Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017, pages 15–26. ACM, 2017.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pages 1597–1607. PMLR, 2020.

[6] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 9185–9193. Computer Vision Foundation / IEEE Computer Society, 2018.

[7] Y. Dong, T. Pang, H. Su, and J. Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 4312–4321. Computer Vision Foundation / IEEE, 2019.

[8] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, and D. Tsipras. Robustness (python library), 2019.

[9] R. B. Girshick. Fast R-CNN. In 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 1440–1448. IEEE Computer Society, 2015.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778. IEEE Computer Society, 2016.

[12] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 7132–7141. Computer Vision Foundation / IEEE Computer Society, 2018.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2261–2269. IEEE Computer Society, 2017.

[14] Q. Huang, I. Katsman, Z. Gu, H. He, S. J. Belongie, and S. Lim. Enhancing adversarial example transferability with an intermediate level attack. In 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pages 4732–4741. IEEE, 2019.

[15] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. In International Conference on Machine Learning, pages 2137–2146. PMLR, 2018.

[16] A. Ilyas, L. Engstrom, and A. Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In International Conference on Learning Representations, 2018.

[17] N. Inkawhich, K. Liang, B. Wang, M. Inkawhich, L. Carin, and Y. Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. Advances in Neural Information Processing Systems, 33:20791–20801, 2020.

[18] N. Inkawhich, W. Wen, H. H. Li, and Y. Chen. Feature space perturbations yield more transferable adversarial examples. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 7066–7074. Computer Vision Foundation / IEEE, 2019.

[19] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In International Conference on Learning Representations, 2017.

[20] Q. Li, Y. Guo, and H. Chen. Practical no-box adversarial attacks against dnns. In NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[21] K. J. Liang, G. Heilmann, C. Gregory, S. O. Diallo, D. Carlson, G. P. Spell, J. B. Sigman, K. Roe, and L. Carin. Automatic threat recognition of prohibited items at aviation checkpoint with x-ray imaging: a deep learning approach. In Anomaly Detection and Imaging with X-Rays (ADIX) III, volume 10632, page 1063203. International Society for Optics and Photonics, 2018.

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages 3431–3440. IEEE Computer Society, 2015.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis., 60(2):91–110, 2004.

[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

[25] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2574–2582, 2016.

[26] H. Petzka, M. Kamp, L. Adilova, C. Sminchisescu, and M. Boley. Relative flatness and generalization. Advances in Neural Information Processing Systems, 34, 2021.

[27] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 779–788. IEEE Computer Society, 2016.

[28] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 91–99, 2015.

[29] M. Rigaki and S. Garcia. A survey of privacy attacks in machine learning. arXiv preprint arXiv:2007.07646, 2020.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis., 115(3):211–252, 2015.

[31] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet. Adversarial manipulation of deep representations. In Y. Bengio and Y. LeCun, editors, 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

[32] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 4510–4520. Computer Vision Foundation / IEEE Computer Society, 2018.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 2818–2826. IEEE Computer Society, 2016.

[35] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski. Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 1555–1562. IEEE, 2018.

[36] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille. Improving transferability of adversarial examples with input diversity. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 2730–2739. Computer Vision Foundation / IEEE, 2019.

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? Advances in neural information processing systems, 27, 2014.

[38] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3320–3328, 2014.

[39] S. Zagoruyko and N. Komodakis. Wide residual networks. In R. C. Wilson, E. R. Hancock, and W. A. P. Smith, editors, Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016. BMVA Press, 2016.

[40] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 645–654. IEEE Computer Society, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will provide a private link when rebuttal.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] All the experiments are run for 5 individual trials with different random seeds. Since the standard deviation is lower 1%, we report only their average accuracy.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We use public datasets and give references.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We do not utilize such data.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]