Enhancing Adversarial Example Transferability with an Intermediate Level Attack

Qian Huang*
Cornell University
qh53@cornell.edu

Isay Katsman*
Cornell University
isk22@cornell.edu

Horace He*
Cornell University
hh498@cornell.edu

Zeqi Gu*
Cornell University
zg45@cornell.edu

Serge Belongie
Cornell University
sjb344@cornell.edu

Ser-Nam Lim
Facebook AI
sernam@gmail.com

Abstract

Neural networks are vulnerable to adversarial examples, malicious inputs crafted to fool trained models. Adversarial examples often exhibit black-box transfer, meaning that adversarial examples for one model can fool another model. However, adversarial examples are typically overfit to exploit the particular architecture and feature representation of a source model, resulting in sub-optimal black-box transfer attacks to other target models. We introduce the Intermediate Level Attack (ILA), which attempts to fine-tune an existing adversarial example for greater black-box transferability by increasing its perturbation on a pre-specified layer of the source model, improving upon state-of-the-art methods. We show that we can select a layer of the source model to perturb without any knowledge of the target models while achieving high transferability. Additionally, we provide some explanatory insights regarding our method and the effect of optimizing for adversarial examples using intermediate feature maps.

1. Introduction

Adversarial examples are small, imperceptible perturbations of images carefully crafted to fool trained models [30, 8]. Studies such as [14] have shown that Convolutional Neural Networks (CNNs) are particularly vulnerable to such adversarial attacks. The existence of these adversarial attacks suggests that our architectures and training procedures produce fundamental blind spots in our models, and that our models are not learning the same features that humans do.

These adversarial attacks are of interest for more than just the theoretical issues they pose – concerns have also been raised over the vulnerability of CNNs to these perturbations in the real world, where they are used for mission-critical applications such as online content filtration systems and self-driving cars [7, 15]. As a result, a great deal of effort

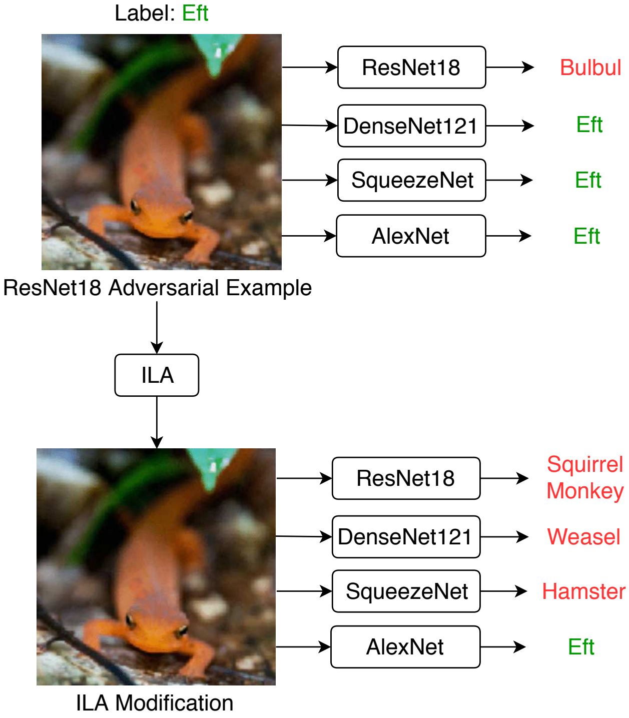

Figure 1: An example of an ILA modification of a pre-existing adversarial example for ResNet18. ILA modifies the adversarial example to increase its transferability. Note that although the original ResNet18 adversarial example managed to fool ResNet18, it does not manage to fool the other networks. The ILA modification of the adversarial example is, however, more transferable and is able to fool more of the other networks.

[*] Equal contribution.

has been dedicated to studying adversarial perturbations. Much of the literature has been dedicated to the development of new attacks that use different perceptibility metrics [2, 28, 26], security settings (black box/white box) [23, 1], as well as increasing efficiency [8]. Defending against adversarial attacks is also well studied. In particular, adversarial training, where models are trained on adversarial examples, has been shown to be effective under certain assumptions [18, 27].

Adversarial attacks can be classified into two categories: white-box attacks and black-box attacks. In white-box attacks, information of the model (i.e., its architecture, gradient information, etc.) is accessible, whereas in black-box attacks, the attackers have access only to the prediction. Black-box attacks are a bigger concern for real-world applications for the obvious reason that such applications typically will not reveal their models publicly, especially when security is a concern (e.g., CNN-based objectionable content filters in social media). Consequently, black-box attacks are mostly focused on the transferability of adversarial examples [17].

Moreover, adversarial examples generated using white-box attacks will sometimes successfully attack an unrelated model. This phenomenon is known as "transferability." However, black-box success rates for an attack are nearly always lower than those of white-box attacks, suggesting that the white-box attacks overfit on the source model. Different adversarial attacks transfer at different rates, but most of them are not optimizing specifically for transferability. This paper aims to achieve the goal of increasing the transferability of a given adversarial example. To this end, we propose a novel method that fine-tunes a given adversarial example through examining its representations in intermediate feature maps that we call *Intermediate Level Attack* (ILA).

Our method draws upon two primary intuitions. First, while we do not expect the direction found by the original adversarial attack to be the most optimal for transferability, we do expect it to be a reasonable proxy, as it still transfers far better than random noise would. As such, if we are searching for a more transferable attack, we should be willing to stray from the original attack direction in exchange for increasing the norm[1]. However, from the ineffectiveness of random noise on neural networks, we see that straying too far from the original direction will cause a decrease in effectiveness – even if we are able to increase the norm by a modest amount. Thus, we must balance staying close to the original direction and increasing norm. A natural way to do so is to maximize the projection onto the original adversarial perturbation.

Second, we note that although for transferability we would like to sacrifice some direction in exchange for increasing the norm, we are unable to do so in the image space without changing perceptibility, as norm and perceptibility are intrinsically tied[2]. However, if we examine the intermediate feature maps, perceptibility (in image space) is no longer intrinsically tied to the norm in an intermediate feature map, and we may be able to increase the norm of the perturbation in that feature space significantly with no change in perceptibility in the image space. We will investigate the effects of perturbing different intermediate feature maps on transferability and provide insights drawn from empirical observations.

Our contributions are as follows:

- We propose a novel method, ILA, that enhances black-box adversarial transferability by increasing the perturbation on a pre-specified layer of a model. We conduct a thorough evaluation that shows our method improves upon state-of-the-art methods on multiple models across multiple datasets. See Sec. 4.

- We introduce a procedure, guided by empirical observations, for selecting a layer that maximizes the transferability using the source model alone, thus obviating the need for evaluation on transfer models during hyperparameter optimization. See Sec. 4.2.

- Additionally, we provide explanatory insights into the effects of optimizing for adversarial examples using intermediate feature maps. See Sec. 5.

2. Background and Related Work

2.1. General Adversarial Attacks

An adversarial example for a given model is generated by augmenting an image so that in the model's decision space its representation moves into the wrong region. Most prior work in generating adversarial examples for attack focuses on disturbing the softmax output space via the input space [8, 18, 21, 6]. Some representative white-box attacks are the following:

Gradient Based Approaches The Fast Gradient Sign Method (FGSM) [8] generates an adversarial example with the update rule:

$$x' = x + \epsilon \cdot sign(\nabla_x J(x, y))$$

It is the linearization of the maximization problem

$$\max_{|x'-x|<\epsilon} J(M(x'), y)$$

where x represents the original image; x' is the adversarial example; y is the ground-truth label; J is the loss function; and M is the model until the final softmax layer. Its iterative version (I-FGSM) applies FGSM iteratively [15]. Intuitively,

[1] Perturbations with a higher norm are generally more effective, regardless of layer (holds true for black-box attacks as well).

[2] Under the standard ϵ-ball constraints.

this fools the model by increasing its loss, which eventually causes misclassification. In other words, it finds perturbations in the direction of the loss gradient of the last layer (i.e., the softmax layer).

Decision Boundary Based Approaches Deepfool [21] produces approximately the closest adversarial example iteratively by stepping towards the nearest decision boundary. Universal Adversarial Perturbation [20] uses this idea to craft a single image-agnostic perturbation that pushes most of a dataset's images across a model's classification boundary.

Model Ensemble Attack The methods mentioned above are designed to yield the best performance only on the model they are tuned to attack; often, the generated adversarial examples do not transfer to other models. In contrast, [17] proposed the Model-based Ensembling Attack that transfers better by avoiding dependence on any specific model. It uses k models with softmax outputs, notated as $J_1, \ldots, J_k$, and solves

$$\min_{|x'-x|<\epsilon} -\log\left(\sum_{i=1}^{k}\alpha_i J_i(x')1_y\right) + \lambda d(x,x')$$

Using such an approach, the authors showed that the decision boundaries of different CNNs align with each other. Consequently, an adversarial example that fools multiple models is likely to fool other models as well.

2.2. Intermediate-layer Adversarial Attacks

A small number of studies have focused on perturbing mid-layer outputs. These include [22], which perturbs mid-layer activations by crafting a single universal perturbation that produces as many spurious mid-layer activations as possible. Another is Feature Adversary Attack [32, 25], which performs a targeted attack by minimizing the distance of the representations of two images in internal neural network layers (instead of in the output layer). However, instead of emphasizing adversarial transferability, it focuses more on internal representations. Results in the paper show that even when given a guide image and a dissimilar target image, it is possible to perturb the target image to produce an embedding similar to that of the guide image.

Two other related works [12, 24] focus on perturbing intermediate activation maps for the purpose of increasing adversarial transferability in a method similar to that of [32, 25] except they focus on black-box transferability. Their method does not focus on fine-tuning existing adversarial examples and differs significantly in attack methodology from ours.

Another recent work that examines intermediate layers for the purposes of increasing transferability is TAP [33]. The TAP attack attempts to maximize the norm between the original image x and the adversarial example x' at all layers. In contrast to our approach, they do not attempt to take advantage of a specific layer's feature representations, instead choosing to maximize the norm of the difference across all layers. In addition, unlike their method which generates an entirely new adversarial example, our method fine-tunes existing adversarial examples, allowing us to leverage existing adversarial attacks.

3. Approach

Based on the motivation presented in the introduction, we propose the Intermediate Level Attack (ILA) framework, shown in Algorithm 2. We propose the following two variants, differing in their definition of the loss function L. Note that we define $F_l(x)$ as the output at layer l of a network F given an input x.

```
Require: Original image in dataset $x$; Adversarial example $x'$ generated for $x$ by baseline attack; Function $F_l$ that calculates intermediate layer output; $L_\infty$ bound $\epsilon$; Learning rate $lr$; Iterations $n$; Loss function $L$.
procedure ILA($x', F_l, \epsilon, lr, L$)
    $x'' = x$
    $i = 0$
    while $i < n$ do
        $\Delta y'_l = F_l(x') - F_l(x)$
        $\Delta y''_l = F_l(x'') - F_l(x)$
        $x'' = x'' - lr \cdot sign(\nabla_{x''} L(y'_l, y''_l))$
        $x'' = clip_\epsilon(x'' - x) + x$
        $x'' = clip_{\text{image range}}(x'')$
        $i = i + 1$
    end while
    return $x''$
end procedure
```

Figure 2: Intermediate Level Attack algorithm

3.1. Intermediate Level Attack Projection (ILAP) Loss

Given an adversarial example x' generated by attack method A for natural image x, we wish to enhance its transferability by focusing on a layer l of a given network F. Although x' is not the optimal direction for transferability, we view x' as a hint for this direction. We treat $\Delta y'_l = F_l(x') - F_l(x)$ as a directional guide towards becoming more adversarial, with emphasis on the disturbance at layer l. Our attack will attempt to find an x'' such that $\Delta y''_l = F_l(x'') - F_l(x)$ matches the direction of $\Delta y'_l$ while maximizing the norm of the disturbance in that direction. The high-level idea is that we want to maximize $\text{proj}_{\Delta y'_l} \Delta y''_l$ for the reasons expressed in Section 1. Since this is a maximization, we can disregard constants, and this simply becomes the dot product. The objective we solve is given below, and we term it the *ILA projection loss*:

$$L(y'_l, y''_l) = -\Delta y''_l \cdot \Delta y'_l \tag{1}$$

3.2. Intermediate Level Attack Flexible (ILAF) Loss

Since the image x' may not be the optimal direction for us to optimize towards, we may want to give the above loss greater flexibility. We do this by explicitly balancing both norm maximization and also fidelity to the adversarial direction at layer l. We note that in a rough sense, ILAF is optimizing for the same thing as ILAP. We augment the above loss by separating out the maintenance of the adversarial direction from the magnitude, and control the trade-off with the additional parameter α to obtain the following loss, termed the *ILA flexible loss*:

$$L(y'_l, y''_l) = - \underbrace{\alpha \cdot \frac{\|\Delta y''_l\|_2}{\|\Delta y'_l\|_2}}_{\text{maximize disturbance}} - \underbrace{\frac{\Delta y''_l}{\|\Delta y''_l\|_2} \cdot \frac{\Delta y'_l}{\|\Delta y'_l\|_2}}_{\text{maintain original direction}} \qquad (2)$$

3.3. Attack

In practice, we choose either the ILAP or ILAF loss and iterate n times to attain an approximate solution to the respective maximization objective. Note that the projection loss only has the layer l as a hyperparameter, whereas the flexible loss also has the additional loss weight α as a hyperparameter. The above attack assumes that x' is a pre-generated adversarial example. As such, the attack can be viewed as a fine-tuning of the adversarial example x'. We fine-tune for greater norm of the output difference at layer l (which we hope will be conducive to greater transferability) while attempting to preserve the output difference's direction to avoid destroying the original adversarial structure.

4. Results

We start by showing that ILAP increases transferability against I-FGSM, MI-FGSM [6] and Carlini-Wagner [4] in the context of CIFAR-10 (Sections 4.1 and 4.2). Results for FGSM and Deepfool are shown in Appendix A[3]. We test on a variety of models, namely: ResNet18 [9], SENet18 [10], DenseNet121 [11] and GoogLeNet [29]. Architecture details are specified in Appendix A; note that in the below results sections, instead of referring to the architecture specific layer names, we refer to layer indices (e.g. $l = 0$ is the last layer of the first block). Our models are trained on CIFAR-10 [13] with the code and hyperparameters in [16] to final test accuracies of 94.8% for ResNet18, 94.6% for SENet18, 95.6% for DenseNet121, and 94.9% for GoogLeNet.

For a fair comparison, we use the output of an attack A that was run for 20 iterations as a baseline. ILAP runs for 10 iterations starting from scratch using the output of attack A after 10 iterations as the reference adversarial example. The learning rate is set to 0.002 for both I-FGSM and MI-FGSM[4].

In Section 4.2 we also show that we can select a nearly-optimal layer for transferability using only the source model. Moreover, ILAF allows further tuning to improve the performance across layers (Section 4.3).

Finally, we demonstrate that ILAP also improves transferability under the more complex setting of ImageNet [5] and that it supercedes state-of-the-art attacks focused on increasing transferability, namely the Zhou et al. attack (TAP) [33] and the Xie et al. attack [31] (Section 4.4).

4.1. ILAP Targeted at Different L Values

To confirm the effectiveness of our attack, we fix a single source model and baseline attack method, and then check how ILAP transfers to the other models compared to the baseline attack. Results for ResNet18 as the source model and I-FGSM as the baseline method are shown in Figure 3. Comparing the results of both methods on the other models, we see that ILAP outperforms I-FGSM when targeting any given intermediate layer, and does especially well for the optimal hyperparameter value of $l = 4$. Note that the choice of layer is important for both performance on the source model and target models. Full results are shown in Appendix A.

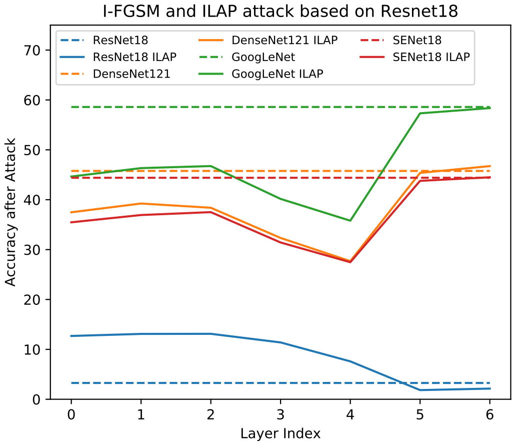

Figure 3: Transfer results of ILAP against I-FGSM on ResNet18 as measured by DenseNet121, SENet18, and GoogLeNet on CIFAR-10 (lower accuracies indicate better attack).

[3] We re-implemented all attacks except Deepfool, for which we used the original publicly provided implementation. For C&W, we used a randomized targeted version, since it has better performance.

[4] Tuning the learning rate does not substantially affect transferability, as shown in Appendix G.

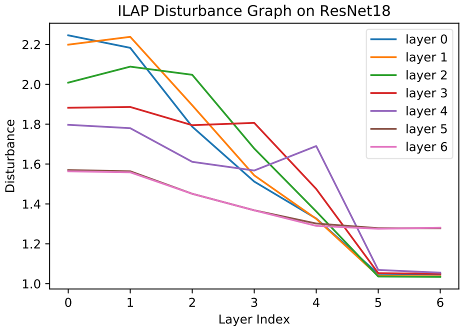

Figure 4: Disturbance values $\left(\frac{\|\Delta y_l''\|_2}{\|\Delta y_l'\|_2}\right)$ at each layer for ILAP targeted at layer $l = 0, 1, ..., 6$ for ResNet18. Observe that the l in the legend refers to the hyperparameter set in the ILAP attack, and afterwards the disturbance values were computed on layers indicated by the l in the x-axis. Note that the last peak is produced by the $l = 4$ ILAP attack.

Table 1: ILAP vs. State-of-the-art Transfer Attacks

	TAP [33]		DI^2-FGSM [31]	
Transfer	20 Itr	Opt ILAP	20 Itr	Opt ILAP
Inc-v4	36.3%	**15.2%**	50.2%	**26.7%**
IncRes-v2	40.7%	**20.1%**	54.6%	**29.3%**

† Same model as source model.

Table 1. Same as experiment in Table 2 but with TAP and DI^2-FGSM from Xie et al. [31]. Evaluation is performed with 5000 randomly selected ImageNet validation set images, and $\epsilon = 0.06$. The source model used is Inc-v3 and the target layer specified for ILAP is Conv2d_4a_3x3.

4.2. ILAP with Pre-Determined L Value

Above we demonstrated that adversarial examples produced by ILAP exhibit the strongest transferability when targeting a specific layer (i.e. choosing a layer as the l hyperparameter). We wish to pre-determine this optimal value based on the source model alone, so as to avoid tuning the hyperparameter l by evaluating on other models. To do this, we examine the relationship between transferability and the ILAP layer disturbance values for a given ILAP attack. We define the disturbance values of an ILAP attack perturbation x'' as values of the function $f(l) = \frac{\|\Delta y_l''\|_2}{\|\Delta y_l'\|_2}$ for all values of l in the source model. For each value of l in ResNet18 (the set of l is defined for each architecture in Appendix A) we plot the disturbance values of the corresponding ILAP attack in Figure 4. The same figure is given for other models in Appendix B.

We notice that the adversarial examples that produce the latest peak in the graph are typically the ones that have highest transferability for all transferred models (Table 2). Given this observation, we propose that the latest l that still exhibits a peak is a nearly optimal value of l (in terms of maximizing transferability). For example, according to Figure 4, we would choose $l = 4$ as the expected optimal hyperparameter for ILAP with ResNet18 as the source model. Table 2 supports our claim and shows that selecting this layer gives an optimal or near-optimal attack. We discuss our discovered explanatory insights for this method in Section 5.3.

4.3. ILAF vs. ILAP

We show that ILAF can further improve transferability with the additional tunable hyperparameter α. The best ILAF result for each model improves over ILAP as shown in Table 3. However, note that the optimal α differs for each model and requires substantial hyperparameter tuning to outperform ILAP. Thus, ILAF can be seen as a more model-specific version that requires more tuning, whereas ILAP works well out of the box. Full results are in Appendix C.

4.4. ILAP on ImageNet

We also tested ILAP on ImageNet, with ResNet18, DenseNet121, SqueezeNet, and AlexNet pretrained on ImageNet (as provided in [19]). The learning rates for all attacks are tuned for best performance. For I-FGSM the learning rate is set to 0.008, for ILAP with I-FGSM to 0.01, for MI-FGSM to 0.018, and for ILAP with MI-FGSM to 0.018. To evaluate transferability, we tested the accuracies of different models over adversarial examples generated from all 50000 ImageNet test images. We observe that ILAP improves over I-FGSM and MI-FGSM on ImageNet. Results for ResNet18 as the source model and I-FGSM as the baseline attack are shown in Figure 5. Full results are in Appendix D.

In order to show our approach outperforms pre-existing methods, we tested ILAP against both TAP [33][5] and Xie et al. [31][6] in an ImageNet setting. The results are shown in Table 1[7].

5. Explaining the Effectiveness of Intermediate Layer Emphasis

At a high level, we motivated projection in an intermediate feature map as a way to increase transferability. We saw empirically that it was desireable to target the layer corresponding to the latest peak (see Figure 4) on the source

[5] Code was not made available for this paper, hence we reproduced their method to the best of our ability.

[6] Pretrained ImageNet models for Inc-v3, Inc-v4, and IncRes-v2 were obtained from Cadene's Github repo [3].

[7] Results indicating that ILAP is competitive with TAP on CIFAR-10 are in Appendix H.

Table 2: ILAP Results

Source	Transfer	MI-FGSM			C & W		
		20 Itr	10 Itr ILAP	Opt ILAP	1000 Itr	500 Itr ILAP	Opt ILAP
ResNet18 ($l = 4$)	ResNet18†	5.7%	11.3%	**2.3%** (6)	7.3%	5.2%	**2.1%** (5)
	SENet18	33.8%	**30.6%**	**30.6%** (4)	85.4%	**41.7%**	**41.7%** (4)
	DenseNet121	35.1%	**30.4%**	**30.4%** (4)	84.4%	**41.7%**	**41.7%** (4)
	GoogLeNet	45.1%	**37.7%**	**37.7%** (4)	90.6%	**57.3%**	**57.3%** (4)
SENet18 ($l = 4$)	ResNet18	31.0%	**27.5%**	**27.5%** (4)	87.5%	**42.7%**	**42.7%** (4)
	SENet18†	3.3%	10.0%	**2.6%** (6)	6.2%	7.3%	**3.1%** (5)
	DenseNet121	31.6%	**27.3%**	**27.3%** (4)	88.5%	**38.5%**	**38.5%** (4)
	GoogLeNet	41.1%	**34.8%**	**34.8%** (4)	91.7%	**52.1%**	**52.1%** (4)
DenseNet121 ($l = 6$)	ResNet18	34.4%	**28.1%**	**28.1%**(6)	87.5%	**37.5%**	**37.5%** (6)
	SENet18	33.5%	**27.7%**	**27.7%** (6)	86.5%	**34.4%**	**34.4%** (6)
	DenseNet121†	6.4%	4.0%	**0.8%**(9)	2.1%	**0.0%**	**0.0%** (9)
	GoogLeNet	36.3%	**30.3%**	**30.3%** (6)	90.6%	**45.8%**	**45.8%** (6)
GoogLeNet ($l = 9$)	ResNet18	44.6%	34.5%	**33.2%**(3)	89.6%	63.5%	**60.4%** (7)
	SENet18	43.0%	33.5%	**32.6%**(3)	90.6%	**53.1%**	**53.1%** (9)
	DenseNet121	38.9%	29.2%	**28.8%**(3)	89.6%	58.3%	**51.0%** (8)
	GoogLeNet†	1.5%	1.4%	**0.5%** (11)	4.2%	**0.0%**	**0.0%** (12)

† Same model as the source model.

Table 2. Accuracies after attack are shown for the models (lower accuracies indicate better attack). The hyperparameter l in the ILAP attack is being fixed for each source model as decided by the layer disturbance graphs (e.g. setting $l = 4$ for ResNet18 since it was the last peak in Figure 4). "Opt ILAP" refers to a 10 iteration ILAP that chooses the optimal layer (determined by evaluating on transfer models). Perhaps surprisingly, ILAP beats out the baseline attack on the original model as well.

Table 3: ILAP vs. ILAF

Model	ILAP (best)	ILAF (best)
DenseNet121	27.7%	26.6%
GoogLeNet	35.8%	34.7%
SENet18	27.5%	26.3%

Table 3. Here we show the difference in transfer performance between ILAP vs. ILAF generated using ResNet18 (with optimal hyperparameters for both attacks).

model in order to maximize transferability. In this section, we attempt to explain the factors causing ILAP performance to vary across layers as well as what they suggest about the optimal layer for ILAP. As we iterate through layer indices, there are two factors affecting our performance: the angle between the original perturbation direction and best transfer direction (defined below in Section 5.1) as well as the linearity of the model decision boundary.

Below, we discuss how the factors change across layers and affect transferability of our attack.

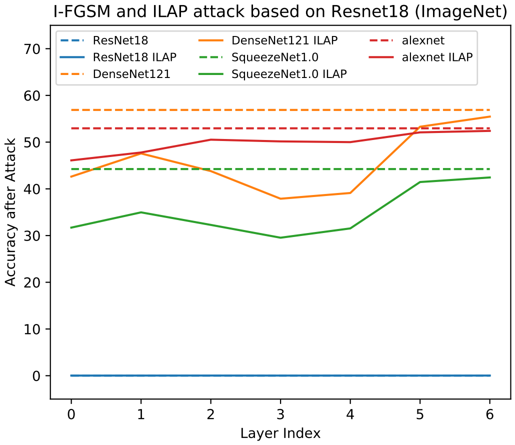

Figure 5: Transfer results of ILAP against I-FGSM on ResNet18 as measured by DenseNet121, SqueezeNet, and AlexNet on ImageNet (lower accuracies indicate better attack).

5.1. Angle between the Best Transfer Direction and the Original Perturbation

Motivated by [17] (where it is shown that the decision boundaries of models with different architectures often align) we define the *Best Transfer Direction* (BTD):

Best Transfer Direction: Let x be an image and M be a large (but finite) set of distinct CNNs. Find x' such that

$$x' = \underset{x' \ s.t. \ |x'-x|<\epsilon}{\arg\max} \sum_{m\in M} \mathbb{1}[m(x') \neq m(x)]$$

Then the *Best Transfer Direction* of x is $BTD_x = \frac{x'-x}{\|x'-x\|}$.

Since our method uses the original perturbation as an approximation for the BTD, it is intuitive that the better this approximation is in the current feature representation, the better our attack will perform.

We want to investigate the nature of how well a chosen source model attack, like I-FGSM, aligns with the BTD throughout layers. Here we measure alignment between an I-FGSM perturbation and an empirical estimate of the BTD (a multi-fool perturbation of the four models we evaluate on in the CIFAR-10 setting) using the angle between them. We investigate the alignment between the feature map outputs of the I-FGSM perturbation and the BTD at each layer. As shown in Figure 6, the angle between the perturbation of I-FGSM and that of the BTD decreases as we iterate the layer indices. Therefore, the later the target layer is in the source model, the better it is to use I-FGSM's attack direction as a guide. This is a factor *increasing* transfer attack success rate as layer indices increase.

To test our hypothesis, we propose to eliminate this source of variation in performance by using a multi-fool perturbation as the starting perturbation for ILAP, which is a better approximation for the BTD. As shown in Figure 7, ILAP performs substantially better when using a multi-fool perturbation as a guide rather than an I-FGSM perturbation, thus confirming that using a better approximation of the BTD gives better performance for ILAP. In addition, we see that these results correspond with what we would expect from Figure 6. In the earlier layers, I-FGSM is a worse approximation of the BTD, so passing in a multi-fool perturbation improves performance significantly. In the later layers, I-FGSM is a much better approximation of the BTD, and we see that passing in a multi-fool perturbation does not increase performance much.

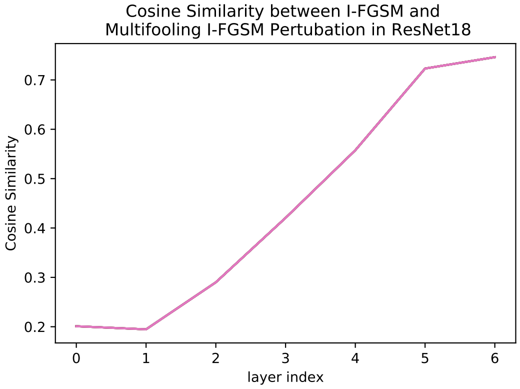

Figure 6: As shown in the above figure, in terms of angle, I-FGSM produces a better approximation for the estimated best transfer direction as we increase the layer index.

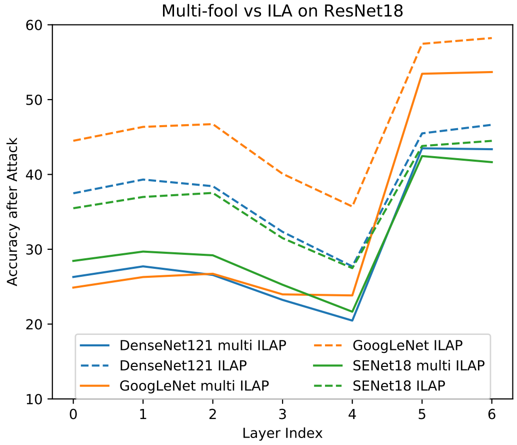

Figure 7: Here we show that ILAP with a better approximation for BTD (multi-fool) performs better. In addition, using a better approximation for BTD disproportionately improves the earlier layers' performance.

5.2. Linearity of Decision Boundary

If we view I-FGSM as optimizing to cross the decision boundary, we can interpret ILAP as optimizing to cross the decision boundary approximated with a hyper-plane perpendicular to the I-FGSM perturbation. As the layer indices increase, the function from the feature space to the final output of the source model tends to becomes increasingly linear (there are more nonlinearities between earlier layers and the final layer than there are between a later layer and the final layer). In fact, we note that at the final layer, the decision boundary is completely linear. Thus, our linear approximation of the decision boundary becoming more accurate is one factor in improving ILAP performance as we select the later layers.

We define the "true decision boundary" as a majority-vote ensemble of a large number of CNNs. Note that for transfer, we care less about how well we are approximating the source model decision boundary than we do about how well we are approximating the true decision boundary. In most feature representations we expect that the true decision boundary is more linear, as ensembling reduces variance.

However, note that at least in the final layer, by virtue of the source model decision boundary being exactly linear, the true decision boundary cannot be more linear, and is likely to be less linear.

We hypothesize that this flip is what causes us to perform worse in the final layers. In these layers, the source model decision boundary is more linear than the true decision boundary, so our approximation performs poorly. We test this hypothesis by attacking two variants of ResNet18 augmented with 3 linear layers before the last layer: one variant without activations following the added layers (var1) and one with (var2). As shown in Figure 8, ILAP performance decreases less in the second variant. Also note that these nonlinearities also cause worse ILAP performance earlier in the network.

Thus, we conclude that the extreme linearity of the last several layers is associated with ILAP performing poorly.

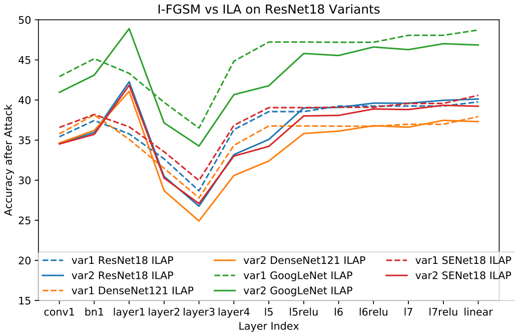

Figure 8: When there is more nonlinearity present in the later portion of the network, the performance of ILAP does not deteriorate as rapidly. Variant 1 (var1) is the version of ResNet18 with additional linear layers not followed by activations, while Variant 2 (var2) does have activations.

5.3. Explanation of the main result

In this section, we tie together all of the above factors to explain the optimal intermediate layer for transferability. Denote:

- the decreasing angle difference between I-FGSM's and BTD's perturbation direction as Factor 1
- the increasing linearity with respect to the decision boundary as we increase layer index as Factor 2, and
- the excessive linearity of the source model decision boundary as Factor 3

On the transfer models, as the index of the attacked source model layer increases, Factors 1 and 2 increase attack rate, while Factor 3 decreases the attack rate. Thus, before some layer, Factors 1 and 2 cause transferability to increase as layer index increases; however, afterward, Factor 3 wins out and causes transferability to decrease as the layer index increases. Thus the layer right before the point where this switch happens is the layer that is optimal for transferability.

We note that this explanation would also justify the method presented in Section 4.2. Intuitively, having a peak corresponds with having the linearized decision boundary (from using projection as the objective) be very different from the source model's decision boundary. If this were not the case, then I-FGSM would presumably have found this improved perturbation already. As such, choosing the last layer that we can get a peak at corresponds with both having as linear of a decision boundary as possible (as late of a layer as possible) while still having enough room to move (the peak).

On the source model, since there is no notion of a "transfer" attack, Factor 3 and Factor 1 do not have any effect. Therefore, Factor 2 causes the performance of the later layers to improve, so much so that at the final layer ILAP's performance on the source model is actually equal or better on all the attacks we used as baselines (see Figure 3). We hypothesize the improved performance on the source model is the result of a simpler loss and thus an easier to optimize loss landscape.

6. Conclusion

We introduce a novel attack, coined ILA, that aims to enhance the transferability of any given adversarial example. It is a framework with the goal of enhancing transferability by increasing projection onto the *Best Transfer Direction*. Within this framework, we propose two variants, ILAP and ILAF, and analyze their performance. We demonstrate that there exist specific intermediate layers that we can target with ILA to substantially increase transferability with respect to the attack baselines. In addition, we show that a near-optimal target layer can be selected without any knowledge of transfer performance. Finally, we provide some intuition regarding ILA's performance and why it performs differently in different feature spaces.

Potential future works include making use of the interactions between ILA and existing adversarial attacks to explain differences among existing attacks, as well as extending ILA to perturbations produced for different settings (universal or targeted perturbations). In addition, other methods of attacking intermediate feature spaces could be explored, taking advantage of the properties we explored in this paper.

Acknowledgements

We want to thank Pian Pawakapan, Prof. Kavita Bala, and Prof. Bharath Hariharan for helpful discussions. This work is supported in part by a Facebook equipment donation.

References

[1] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[2] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017.

[3] Remi Cadene. pretrained-models.pytorch. `https://github.com/Cadene/pretrained-models.pytorch`, 2019.

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

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. 2017.

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and D. Song. Robust physical-world attacks on deep learning models. 2017.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017.

[11] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[12] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[15] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[16] Kuang Liu. Pytorch cifar10. `https://github.com/kuangliu/pytorch-cifar`, 2018.

[17] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Xiaodong Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017.

[19] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *ACM Multimedia*, 2010.

[20] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94, 2017.

[21] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[22] Konda Reddy Mopuri, Aditya Ganeshan, and R. Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[23] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *AsiaCCS*, 2017.

[24] Andras Rozsa, Manuel Günther, and Terrance E. Boult. LOTS about attacking deep features. In *International Joint Conference on Biometrics (IJCB)*, 2017.

[25] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J. Fleet. Adversarial manipulation of deep representations. *CoRR*, abs/1511.05122, 2015.

[26] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Adversarial generative nets: Neural network attacks on state-of-the-art face recognition. *CoRR*, abs/1801.00349, 2017.

[27] Aman Sinha, Hongseok Namkoong, and John C. Duchi. Certifying some distributional robustness with principled adversarial training. 2017.

[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *CoRR*, abs/1710.08864, 2017.

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[31] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[32] Xiaoyong Yuan, Pan He, Qile Zhu, Rajendra Rana Bhat, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *CoRR*, abs/1712.07107, 2017.

[33] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *ECCV*, 2018.