Enhancing Adversarial Example Transferability with an Intermediate Level Attack Supplementary Material

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

This supplementary file consists of:

- A more thorough description of the networks used, the layers selected for attack, and full results on other attacks tested
- Complete disturbance graphs across layers of an attacking comprised of ILAP and I-FGSM
- A more complete result showing the comparison between ILAP and ILAF
- Full results for ILAP's performance on ImageNet
- Visualization of decision boundary
- Results for different L_∞ norm values
- Results for ablating the learning rate used in ILAP
- Results comparing ILAP to TAP [8] on CIFAR-10

A. ILAP Network Overview and Results for Other Base Attacks

As shown in the main paper, we tested ILAP against MI-FGSM, C&W, and TAP. We also tested I-FGSM, DeepFool, and FGSM. We test on a variety of models, namely: ResNet18 [1], SENet18 [2], DenseNet121[3] and GoogLeNet [7] trained on CIFAR-10. For each source model, each large block output in the source model and each attack A, we generate adversarial examples for all images in the test set using A with 20 iterations as a baseline. We then generate adversarial examples using A with 10 iterations as input to ILA, which will then run for 10 iterations. The learning rate is set to 0.002 for I-FGSM, 0.002 for I-FGSM with momentum and 0.006 for ILAP. We are in the L_∞ norm setting with $\epsilon = 0.03$ for all attacks. We then evaluate transferability of baseline and ILA adversarial examples over the other models by testing their accuracies, as shown in Figure 3.

Below is the list of layers (models from [5]) we picked for each source model, which is indexed starting from 0 in the experiment results:

- ResNet18: conv, bn, layer1, layer2, layer3, layer4, linear (layer1-4 are basic blocks)
- GoogLeNet: pre_layers, a3, b3, maxpool, a4, b4, c4, d4, e4, a5, b5, avgpool, linear
- DenseNet121: conv1, dense1, trans1, dense2, trans2, dense3, trans3, dense4, bn, linear
- SENet18: conv1, bn1, layer1, layer2, layer3, layer4, linear (layer1-4 are pre-activation blocks)

Additional results for the I-FGSM, FGSM, and DeepFool attacks are given in tables 1 and 2. Note that the output of DeepFool is clipped to satisfy our ϵ-ball constraint.

B. Disturbance graphs

In this experiment, we used the same setting as our main experiment in Appendix A to generate adversarial examples, with only I-FGSM used as the reference attack. The average disturbance of each set of adversarial examples is calculated at each layer. We repeated the experiment for all four models described in Appendix A, as shown in Figure 4. Observe that the l in the legend refers to the hyperparameter set in the ILA attack, and afterwards the disturbance values were computed on layers indicated by the l in the x-axis.

*Equal contribution.

Table 1: ILAP vs. I-FGSM and DeepFool Results

Source	Transfer	I-FGSM			DeepFool		
		20 Itr	10 Itr ILAP	Opt ILAP	50 Itr	25 Itr ILAP	Opt ILAP
ResNet18 ($l = 4$)	ResNet18†	3.3%	7.6%	**1.8%** (5)	48.7%	12.9%	**5.4%** (5)
	SENet18	44.4%	**27.5%**	**27.5%** (4)	87.4%	**43.7%**	**43.7%** (4)
	DenseNet121	45.8%	**27.7%**	**27.7%** (4)	89.1%	**43.8%**	**43.8%** (4)
	GoogLeNet	58.6%	**35.8%**	**35.8%** (4)	89.3%	**50.7%**	**50.7%** (4)
SENet18 ($l = 4$)	ResNet18	36.8%	**25.8%**	**25.8%** (4)	91.9%	40.3%	**39.9%** (5)
	SENet18†	2.4%	7.9%	**2.3%** (6)	56.8%	11.4%	**5.1%** (6)
	DenseNet121	38.0%	**25.9%**	**25.9%** (4)	92.9%	41.3%	**41.1%** (5)
	GoogLeNet	48.4%	**33.7%**	**33.7%** (4)	92.3%	**48.7%**	**48.7%** (4)
DenseNet121 ($l = 6$)	ResNet18	45.1%	**26.7%**	**26.7%**(6)	81.6%	**30.1%**	**30.1%** (6)
	SENet18	43.4%	**26.1%**	**26.1%**(6)	81.5%	29.0%	**28.9%** (7)
	DenseNet121†	2.6%	1.7%	**0.8%**(9)	34.9%	4.1%	**3.3%** (9)
	GoogLeNet	47.3%	**28.6%**	**28.6%**(6)	82.3%	**32.4%**	**32.4%** (6)
GoogLeNet ($l = 9$)	ResNet18	55.9%	34.0%	**32.7%**(3)	92.3%	**44.0%**	**44.0%** (9)
	SENet18	55.6%	33.1%	**31.8%** (3)	92.1%	**42.9%**	**42.9%** (9)
	DenseNet121	48.9%	28.7%	**28.1%**(3)	93.1%	**38.1%**	**38.1%** (9)
	GoogLeNet†	0.9%	0.8%	**0.4%** (11)	51.5%	4.2%	**3.9%** (11)

Table 1. Accuracies after attack using ILAP based on I-FGSM and DeepFool. Note that although significant improvement for transfer is exhibited for DeepFool, the original attack transfer rates are quite poor (the accuracies are still quite high after a DeepFool transfer attack).

C. ILAP vs ILAF Full Result

As described in the main paper, we compared the performace of ILAP and ILAF with a range of α. We used the same setting as our main experiment in Appendix A for ILAP and ILAF to generate adversarial examples, with only I-FGSM used as the reference attack. The result is shown in Figure 5.

D. ILAP on ImageNet Full Result

We tested ILAP against I-FGSM and I-FGSM with momentum on ImageNet similarly to the experiment on CIFAR-10. The models we used are ResNet18, DenseNet121, SqueezeNet1.0 and AlexNet. The learning rate is set to 0.008 for I-FGSM, 0.01 for ILAP plus I-FGSM, 0.018 for I-FGSM with momentum and 0.018 for ILAP plus I-FGSM with momentum. To evaluate transferability, we test the accuracies of different models over adversarial examples generated from all 50000 ImageNet test images, as shown in Figure 6.

Below is the list of layers (models from [6]) we picked for each source model:

- ResNet18: conv1, bn1, layer1, layer2, layer3, layer4, fc
- DenseNet121: conv0, denseblock1, transition1, denseblock2, transition2, denseblock3, transition3, denseblock4, norm5, classifier
- SqueezeNet1.0: Features: 0 3 4 5 7 8 9 10 12, classifier
- AlexNet: Features: 0 3 4 6 8 10, classifiers: 1 4

E. Visualization of the Decision Boundary

To gain some understanding over how ILA interplays with the decision boundaries, we visualize the two dimensional plane between the initial I-FGSM perturbation and the ILA perturbation for some examples. Visualization is done on Resnet with layer 4, and I-FGSM as the starting perturbation. See Figure 7.

F. Fooling with Different L_∞ Values

In this experiment, we use ILAP to generate adversarial examples with an I-FGSM baseline attack on ResNet18 with $\epsilon = 0.02, 0.03, 0.04, 0.05$, while other settings are kept the same as in section A. We then evaluated their transferability against I-FGSM baseline on the adversarial examples of the whole test set, as shown in Figure 8.

Table 2: ILAP vs. FGSM Results

		FGSM		
Source	Transfer	20 Itr	10 Itr ILAP	Opt ILAP
ResNet18 ($l = 6$)	ResNet18†	47.7%	**2.0%**	**2.0%** (6)
	SENet18	63.6%	**42.6%**	**42.6%** (6)
	DenseNet121	64.9%	44.6%	**44.5%** (5)
	GoogLeNet	66.5%	55.5%	**54.2%** (4)
SENet18 ($l = 6$)	ResNet18	60.7%	37.4%	**36.1%** (5)
	SENet18†	40.7%	**3.0%**	**3.0%** (6)
	DenseNet121	61.8%	37.0%	**36.3%** (5)
	GoogLeNet	63.8%	46.3%	**45.3%** (5)
DenseNet121 ($l = 7$)	ResNet18	65.0%	36.4%	**36.2%** (6)
	SENet18	65.0%	**35.5%**	**35.5%** (7)
	DenseNet121†	47.3%	5.8%	**0.9%** (9)
	GoogLeNet	64.6%	37.6%	**37.4%** (6)
GoogLeNet ($l = 9$)	ResNet18	64.9%	**43.5%**	**43.5%** (9)
	SENet18	65.1%	**43.8%**	**43.8%** (9)
	DenseNet121	63.7%	**39.7%**	**39.7%** (9)
	GoogLeNet†	36.6%	5.9%	**0.6%** (12)

† Same model as source model.

Table 2. Accuracies after attack based on FGSM. Note that significant improvement occurs in the ILAP settings.

G. Learning Rate Ablation

We set iterations to 20 for both I-FGSM and I-FGSM with Momentum and experimented different learning rates on ResNet18. We then evaluate different models' accuracies on the generated $50 \times 32 = 1600$ adversarial examples, as shown in Table 3 and 4.

H. Comparison to TAP [8]

CIFAR-10 [4] results comparing a 20 iteration TAP [8] baseline to 10 iterations of ILAP using the output of a 10 iteration TAP attack are shown in Table 5.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[2] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017.

[3] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[4] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

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

[6] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *ACM Multimedia*, 2010.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[8] W. Zhou, X. Hou, Y. Chen, M. Tang, X. Huang, X. Gan, and Y. Yang. Transferable adversarial perturbations. In *ECCV*, 2018.

Table 3: Learning rate ablation for I-FGSM

learning rate	ResNet18[†]	SENet18	DenseNet121	GoogLeNet
0.002	3.3%	44.9%	47.1%	59.3%
0.008	0.8%	45.6%	46.8%	60.0%
0.014	0.6%	47.2%	49.4%	59.5%
0.02	1.3%	46.8%	51.4%	59.8%

Table 4: Learning rate ablation for I-FGSM with Momentum

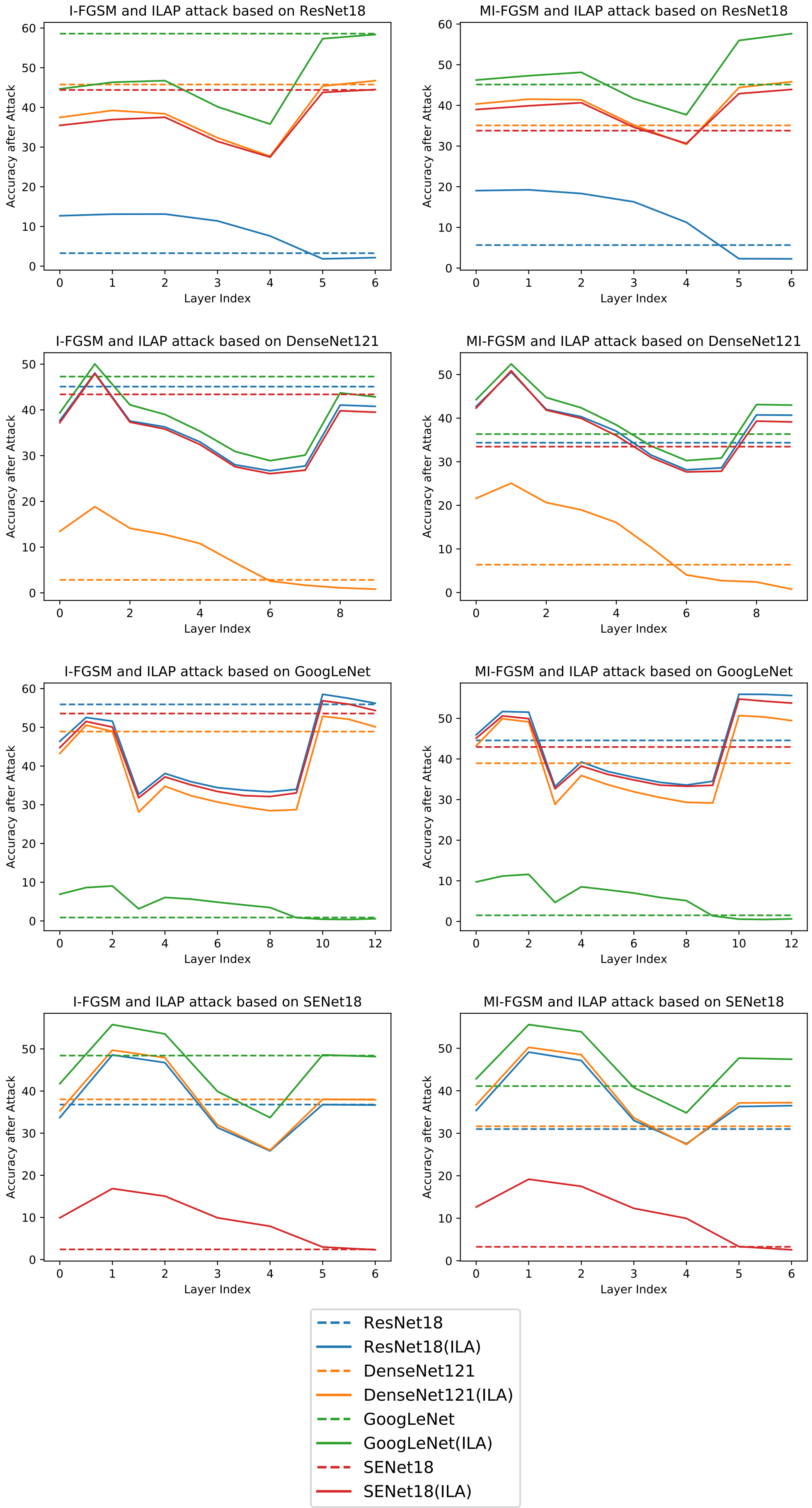
learning rate	ResNet18[†]	SENet18	DenseNet121	GoogLeNet
0.002	5.9%	35.0%	36.6%	46.1%
0.008	0.6%	43.0%	43.8%	56.1%
0.014	0.4%	43.6%	45.2%	55.9%
0.02	0.4%	44.1%	46.4%	57.2%

Table 5: ILAP vs. TAP Results

		TAP [8]	
Source	Transfer	20 Itr	Opt ILAP
ResNet18 ($l = 4$)	ResNet18[†]	6.2%	**1.9%** (6)
	SENet18	31.6%	**28.4%** (4)
	DenseNet121	32.7%	**28.5%** (4)
	GoogLeNet	41.6%	**36.8%** (4)
SENet18 ($l = 4$)	ResNet18	31.4%	**23.5%** (4)
	SENet18[†]	2.0%	**1.7%** (5)
	DenseNet121	31.3%	**24.1%** (4)
	GoogLeNet	41.5%	**33.1%** (4)
DenseNet121 ($l = 6$)	ResNet18	35.2%	**27.4%** (6)
	SENet18	34.2%	**26.8%** (7)
	DenseNet121[†]	4.8%	**1.0%** (9)
	GoogLeNet	37.8%	**29.8%** (6)
GoogLeNet ($l = 9$)	ResNet18	37.1%	**33.6%** (9)
	SENet18	36.5%	**32.9%** (9)
	DenseNet121	32.6%	**28.1%** (9)
	GoogLeNet[†]	1.3%	**0.4%** (12)

[†] Same model as source model.

Table 5. Same as experiment in Table 2 of the main paper but with TAP. Hyperparameters for TAP are set to $lr = 0.002, \epsilon = 0.03, \lambda = 0.005, \alpha = 0.5, s = 3, \eta = 0.01$.



I-FGSM and ILAP attack based on ResNet18

MI-FGSM and ILAP attack based on ResNet18

I-FGSM and ILAP attack based on DenseNet121

MI-FGSM and ILAP attack based on DenseNet121

I-FGSM and ILAP attack based on GoogLeNet

MI-FGSM and ILAP attack based on GoogLeNet

I-FGSM and ILAP attack based on SENet18

MI-FGSM and ILAP attack based on SENet18

Figure 1: Visualizations for ILAP against I-FGSM and MI-FGSM baselines on CIFAR-10

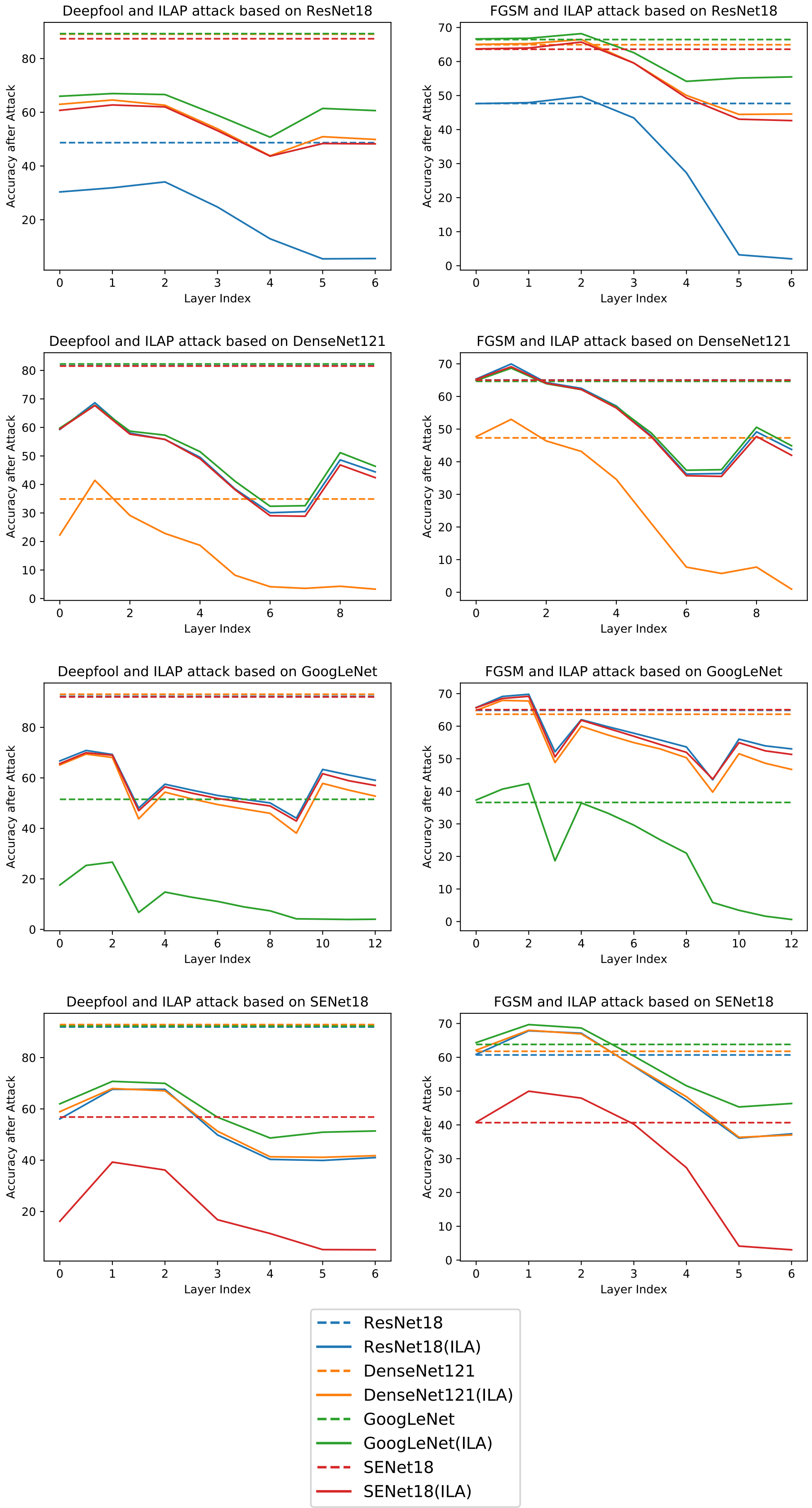


Figure 2: Visualizations for ILAP aganist Deepfool and FGSM with momentum baselines on CIFAR-10

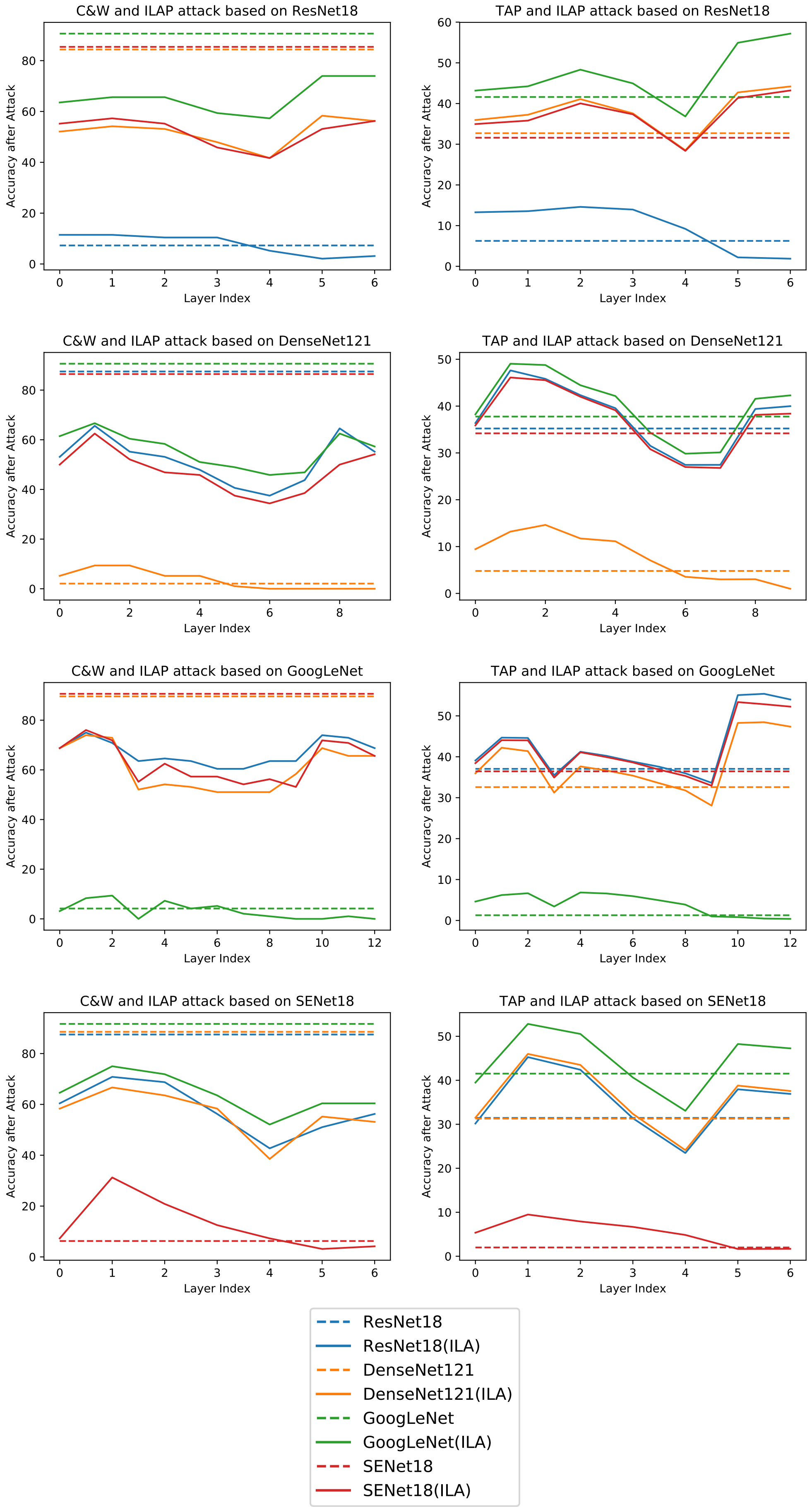


Figure 3: Visualizations for ILAP aganist Deepfool and FGSM with momentum baselines on CIFAR-10

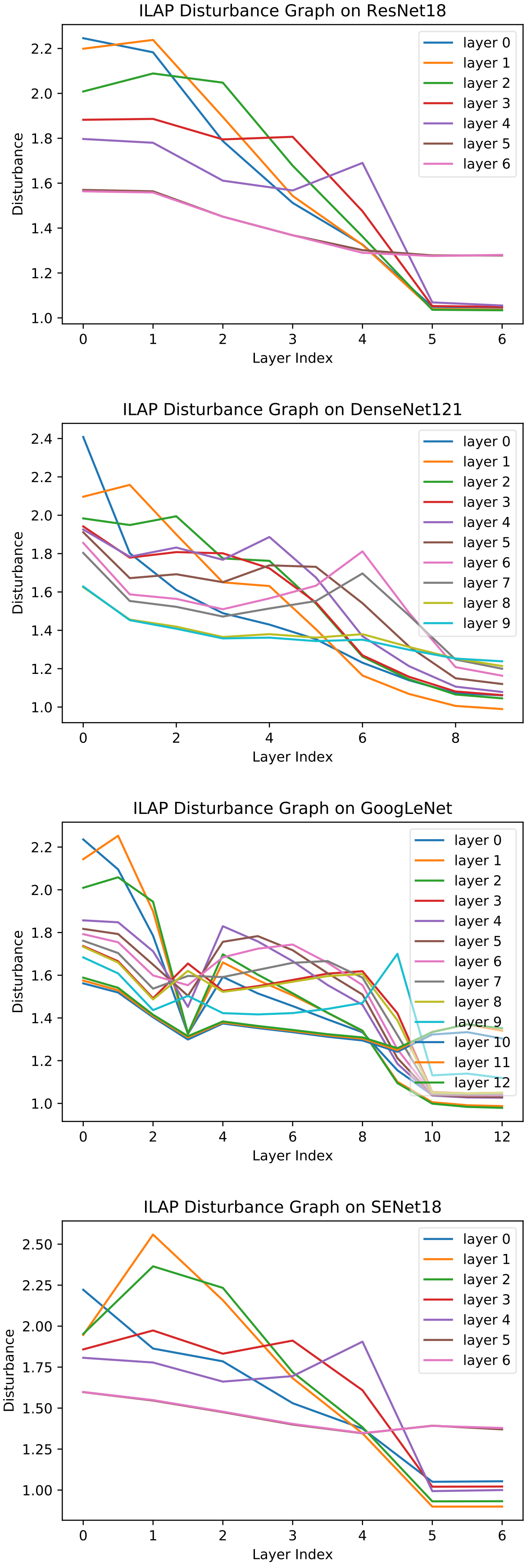


Figure 4: Disturbance graphs of ILAP with I-FGSM as reference

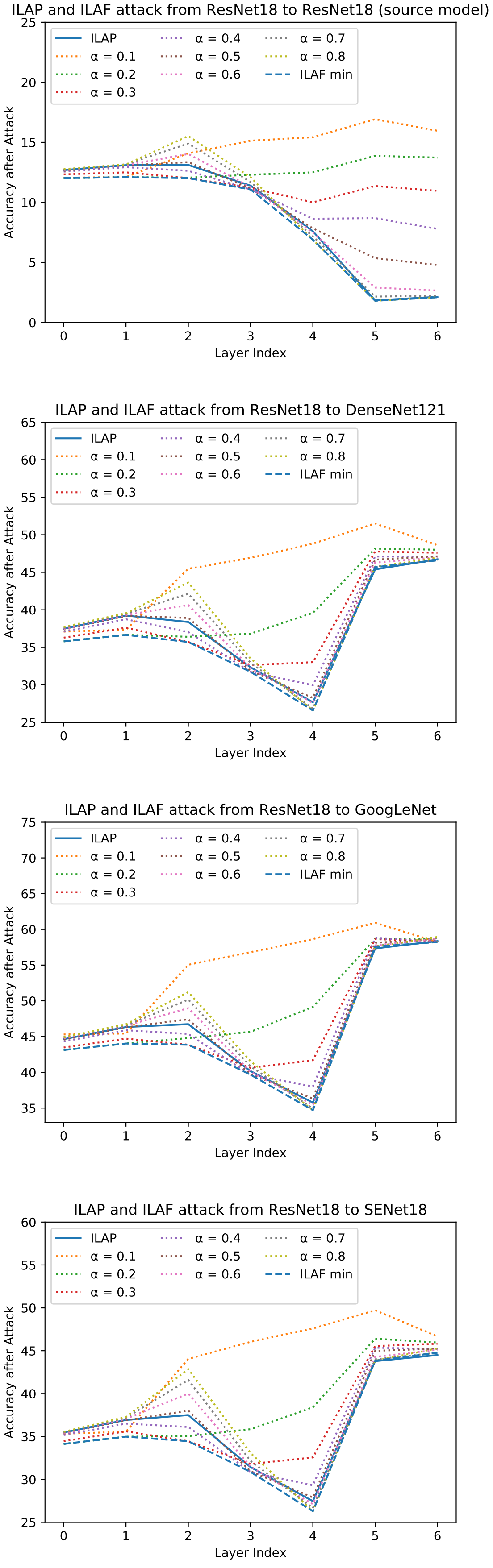


Figure 5: ILAP vs ILAF comparisions

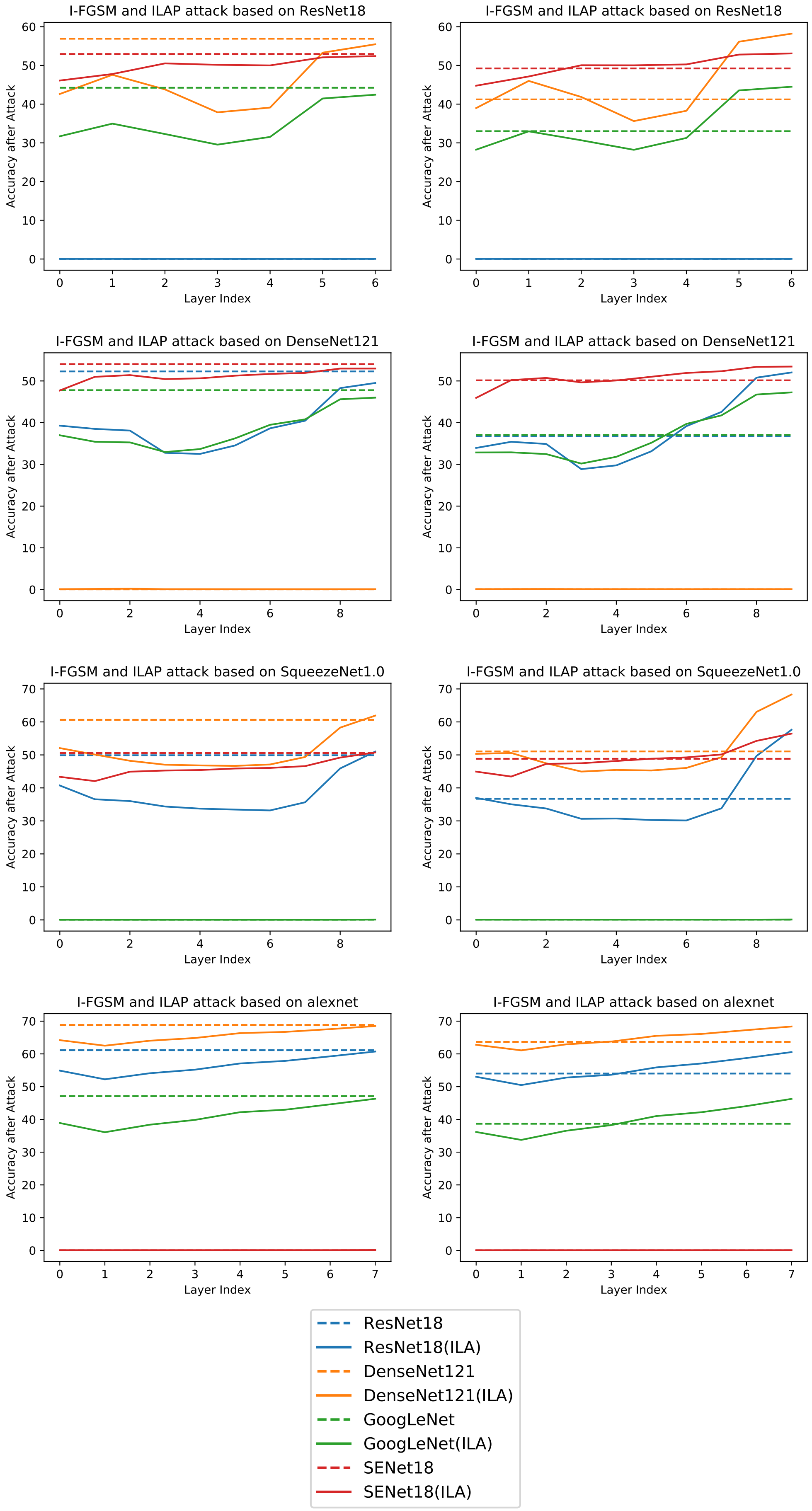


Figure 6: Visualizations for ILAP against I-FGSM and I-FGSM with momentum baselines on ImageNet

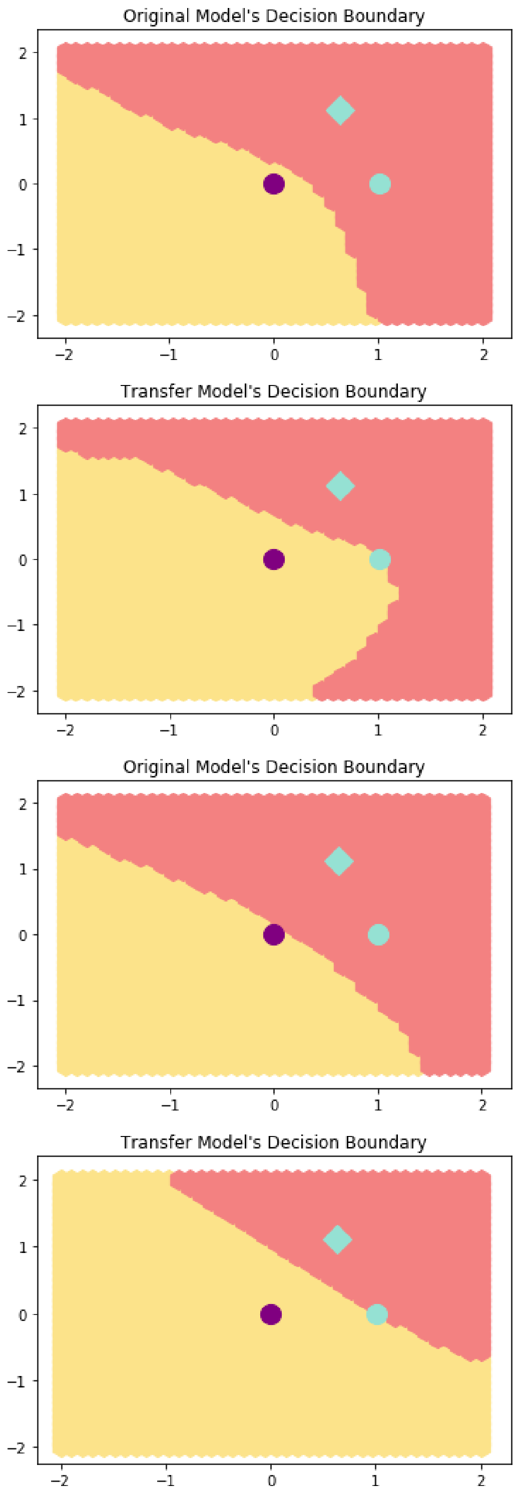


Figure 7: Visualization of the decision boundary relative to the two adversarial examples generated. Yellow is the correct label's decision space, red is the incorrect label's decision space. The purple dot is the original image's location, the green circle is the I-FGSM perturbation, and the green diamond is the ILA perturbation. Note that for the above, it seems the vector between the purple dot and green diamond is more orthogonal to the decision boundary than the vector between the purple dot and green dot (hence roughly indicating that ILA is working as intended in producing a more orthogonal transfer vector).

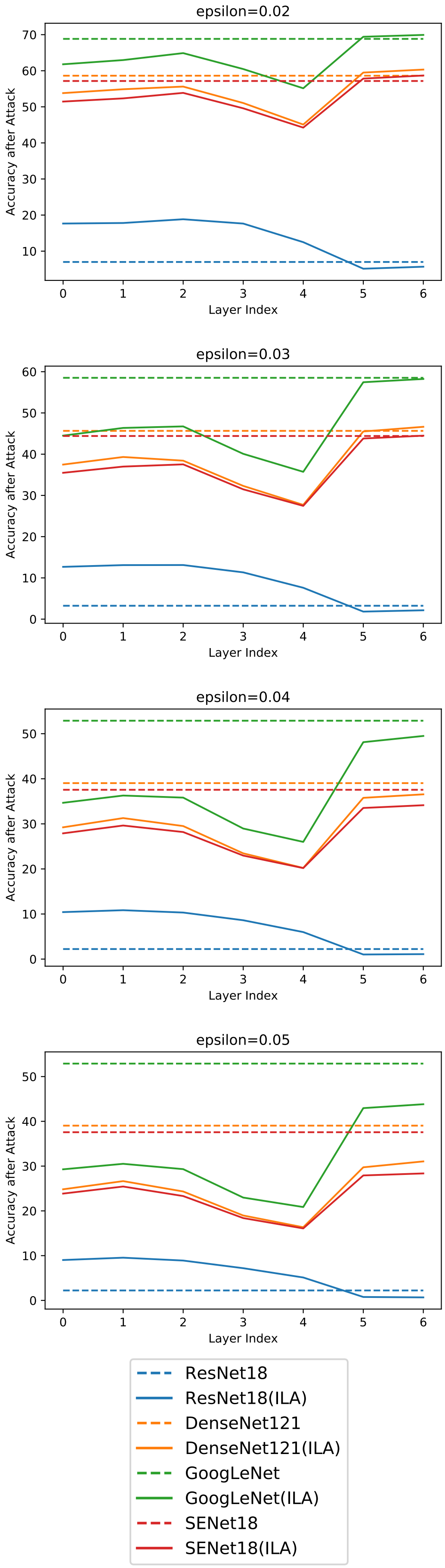


Figure 8: Transferability graphs for different epsilons