Evaluating and Understanding the Robustness of Adversarial Logit Pairing

Logan Engstrom[*] engstrom@mit.edu

Andrew Ilyas[*] ailyas@mit.edu

Anish Athalye[*] aathalye@mit.edu

MIT, LabSix

Abstract

We evaluate the robustness of Adversarial Logit Pairing, a recently proposed defense against adversarial examples. We find that a network trained with Adversarial Logit Pairing achieves 0.6% correct classification rate under targeted adversarial attack, the threat model in which the defense is considered. We provide a brief overview of the defense and the threat models/claims considered, as well as a discussion of the methodology and results of our attack, which may offer insights into the reasons underlying the vulnerability of ALP to adversarial attack.

1 Contributions

For summary, the contributions of this note are as follows:

1. **Robustness**: Under the white-box targeted attack threat model specified in Kannan et al. [9], we upper bound the correct classification rate of the defense to **0.6%** (Table 1). We also perform targeted and untargeted attacks and show that the attacker can reach success rates of 98.6% and 99.9% respectively (Figures 1, 2).
2. **Formulation**: We analyze the ALP loss function and contrast it to that of Madry et al. [10], pointing out several differences from the robust optimization objective (Section 4.1).
3. **Loss landscape**: We analyze the loss landscape induced by ALP by visualizing loss landscapes and adversarial attack trajectories (Section 4.2).

2 Introduction

Neural networks and machine learning models in general are known to be susceptible to adversarial examples, low-magnitude perturbations that induce specific and unintended behaviour [11, 3]. Defenses against these adversarial attacks are of great significance and value. Unfortunately, many proposed defenses have had their claims invalidated by new attacks within their corresponding threat models [4, 8, 5, 6, 2, 12, 1]. A notably robust defense has been that of Madry et al. [10], which proposes a "robust optimization"-based view of defense against adversarial examples, in which the defender tries to find parameters θ^* minimizing the following objective:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in S} L(\theta, x+\delta, y) \right]. \tag{1}$$

Here, L is a prespecified loss function, $\mathcal{D}$ is the labeled data distribution, and $\mathcal{S}$ is the set of admissible adversarial perturbations (specified by a threat model). In practice, the defense is implemented through

[*] Equal contribution

adversarial training, where adversarial examples are generated during the training process and used as inputs. The resulting classifiers have been empirically evaluated to offer increased robustness to adversarial examples on the CIFAR-10 and MNIST datasets under small ℓ_∞ perturbations.

In Kannan et al. [9], the authors claim that the defense of Madry et al. [10] is ineffective when scaled to an ImageNet [7] classifier, and propose a new defense – Adversarial Logit Pairing (ALP). In the ALP defense, a classifier is trained with an alternative training objective that enforces similarity between the model's *logit* activations on unperturbed and adversarial versions of the same image. The loss additionally has a term meant to maintain accuracy on the original training set.

$$\begin{aligned} \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[L(\theta, x, y) + \lambda D\left(f(\theta, x), f(\theta, x+\delta^*)\right)\right] \\ \text{where } \delta^* = \arg\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y), \end{aligned}$$

Here, D is a distance function, f is a function mapping parameters and inputs to logits (via the given network), λ is a hyperparameter, and the rest of the notation is as in (1). This objective is intended to promote "better internal representations of the data" [9] by providing an extra regularization term. In the following sections, we show that ALP can be circumvented using Projected Gradient Descent (PGD) based attacks.

2.1 Setup details

We analyze Adversarial Logit Pairing as implemented by the authors [1]. We use the "models pre-trained on ImageNet" [1] from the code release to evaluate the claims of Kannan et al. [9]. Via private correspondence, the authors acknowledged our result but stated that the results in Kannan et al. [9] were generated with different, unreleased models not included in the official code release.

Our evaluation code is publicly available. [2].

3 Threat model and claims

Table 1: The claimed robustness of Adversarial Logit Pairing against targeted attacks on ImageNet, from [9], compared to the lower bound on attacker success rate from this work. Attacker success rate in this case represents the percentage of times an attacker successfully induces the adversarial target class, whereas accuracy measures the percentage of times the classifier outputs the *correct* class.

Source **Defense** ($\epsilon = 16/255$)	Kannan et al. [9] **Claimed Accuracy**	this work **Defense Accuracy**[3]	this work **Attacker Success**
Madry et al. [10]	1.5%	–	–
Kannan et al. [9]	27.9%[4]	**0.6%**	**98.6%**

ALP is claimed secure under a variety of white-box and black-box threat models; in this work, we consider the *white-box* threat model, where an attacker has full access to the weights and parameters of the model being attacked. Specifically, we consider an Residual Network ALP-trained on the ImageNet dataset, where ALP is claimed to achieve state-of-the-art accuracies in this setting under an ℓ_∞ perturbation bound of 16/255, as depicted in Table 1. The defense is originally evaluated against targeted adversarial attacks, and thus Table 1 refers to the attacker success rate on targeted adversarial attacks. For completeness, we also

[1] https://github.com/tensorflow/models/tree/master/research/adversarial_logit_pairing

[2] https://github.com/labsix/adversarial-logit-pairing-analysis

[3] We calculate this as in [9], i.e. correct classification rate under targeted adversarial attack.

[4] As noted in §2.1, via private correspondence, the authors state that unreleased models were used to generate the results in Kannan et al. [9]. The authors are currently investigating these models; for the sake of comparison, we give the claim from Kannan et al. [9] here.

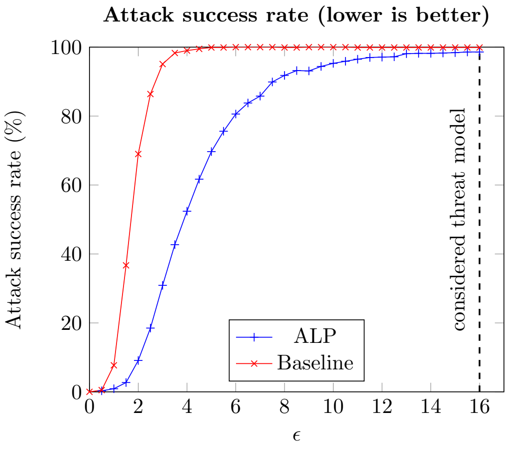

Figure 1: Comparison of ALP-trained model with baseline model under **targeted** adversarial perturbations (with random labels) bounded by varying ϵ from 0 to 16/255. Our attack reaches 98.6% success rate (and 0.6% correct classification rate) at $\epsilon = 16/255$.

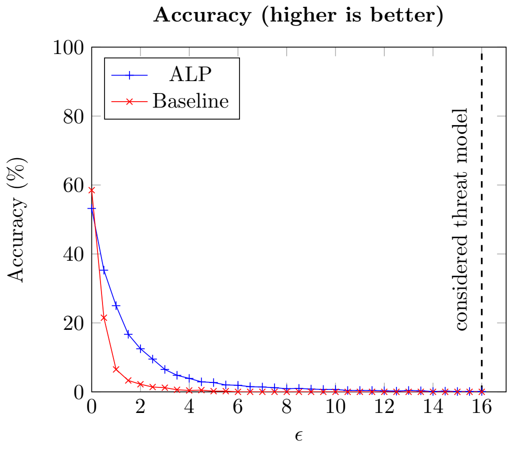

Figure 2: Comparison of ALP-trained model with baseline model under **untargeted** adversarial perturbations bounded by varying ϵ from 0 to 16/255. The ALP-trained model achieves 0.1% accuracy at $\epsilon = 16/255$.

perform a brief analysis on untargeted attacks to show lack of robustness (Figure 2), but do not consider this in the context of the proposed threat model or claims.

Adversary objective. When evaluating attacks, an attack that can produce targeted adversarial examples is *stronger* than an attack that can only produce untargeted adversarial examples. On the other hand, a defense that is only robust against targeted adversarial examples (e.g. with random target classes) is *weaker* than a defense that is robust against untargeted adversarial examples. The ALP paper only attempts to show robustness to targeted adversarial examples.

4 Evaluation

4.1 Analyzing the defense objective

Adversarial Logit Pairing is proposed as an augmentation of adversarial training, which itself is meant to approximate the robust optimization approach outlined in Equation 1. The paper proposes that by adding a “regularizer” to the adversarial training objective, better results on high-dimensional datasets can be achieved. In this section we outline several conceptual differences between ALP as formulated and the robust optimization perspective offered by Madry et al. [10].

Training on natural vs. adversarial images. A key part in the formulation of the robust optimization objective is that minimization with respect to theta is done over the inputs that have been crafted by the max player; θ is not minimized with respect to any “natural” $x \sim \mathcal{D}$. In the ALP formulation, on the other hand, regularization is applied to the loss on *clean* data $L(\theta, x, y)$. This fundamentally changes the optimization objective from the defense of Madry et al. [10].

Generating targeted adversarial examples. A notable implementation decision given in Kannan et al. [9] is to generate targeted adversarial examples during the training process. This again deviates from the robust optimization-inspired saddle point formulation for adversarial training, as the inner maximization player no longer maximizes $L(\theta, x + \delta, y)$, but rather minimizes $L(\theta, x + \delta, y_{adv})$ for another class y_{adv}. Note that although Athalye et al. [2] recommends that *attacks* on ImageNet classifiers be evaluated in the targeted threat model (which is noted in [9] in justifying this implementation choice), this recommendation does not extend to adversarial training or empirically showing that a defense is secure (a defense that is only robust to targeted attacks is *weaker* than one robust to untargeted attacks).

4.2 Analyzing empirical robustness

Empirical evaluations give upper bounds for the robustness of a defense on test data. Evaluations done with weak attacks can be seen as giving loose bounds, while evaluations done with stronger attacks give tighter bounds of true adversarial risk [12]. We find that the robustness of ALP as a defense to adversarial examples is significantly lower than claimed [9].

Attack procedure. We originally used the evaluation code provided by the ALP authors and found that setting the number of steps in the PGD attack to 100 from the default of 20 significantly degrades accuracy. For ease of use we reimplemented a standard PGD attack, which we ran for up to 1000 steps or until convergence. We evaluate both untargeted attacks and targeted attacks with random targets, measuring model accuracy on the former and adversary success rate (percentage of data points classified as the target class) for the latter.

Empirical robustness. We establish tighter upper bounds on adversarial robustness for both the ALP trained ImageNet classifier and the baseline ResNet-based ImageNet classifier with our attack. Our results, with a full curve of ϵ (allowed perturbation) vs attack success rate, are summarized in Figure 1. In the threat model with $\epsilon = 16$ our attack achieves a 98.6% success rate, and reduces the accuracy (percentage of correctly classified examples perturbed by the targeted attack) of the classifier to **0.6%**.

Figure 2 shows that untargeted attacks gives similar results: the ALP-trained model achieves 0.1% accuracy at $\epsilon = 16/255$.

Loss landscapes. We plot loss landscapes around validation inputs in Figure 3. In the loss landscapes we vary the input along a linear space defined by the sign of the gradient and a random rademacher vector, where the x and y axes represent the magnitude of the perturbation added in each direction and the z axis represents the loss. The plots provide evidence for ALP sometimes inducing a “bumpier,” depressed loss landscape tightly around the input points.

Attack convergence. As suggested by analysis of the loss surface, the optimization landscape of the ALP-trained network is less amenable to gradient descent. Examining, for a single data point, the loss over steps of gradient descent in targeted (Figure 4) and untargeted (Figure 5) attacks, we observe that the attack on the ALP-trained network takes more steps of gradient descent.

This was generally true over all data points. The attack on the ALP-trained network required more steps of gradient descent to converge, but true robustness had not increased (e.g. at $\epsilon = 16/255$, both networks have roughly 0% accuracy).

5 Conclusion

In this work, we perform an evaluation of the robustness of the Adversarial Logit Pairing defense (ALP) as proposed in Kannan et al. [9], and show that it is not robust under the considered threat model. We then study the formulation, implementation, and loss landscape of ALP. The evaluation methods we use are general and may help in enhancing evaluation standards for adversarial defenses.

Acknowledgements

We thank Harini Kannan, Alexey Kurakin, and Ian Goodfellow for releasing open-source code and pre-trained models for Adversarial Logit Pairing.

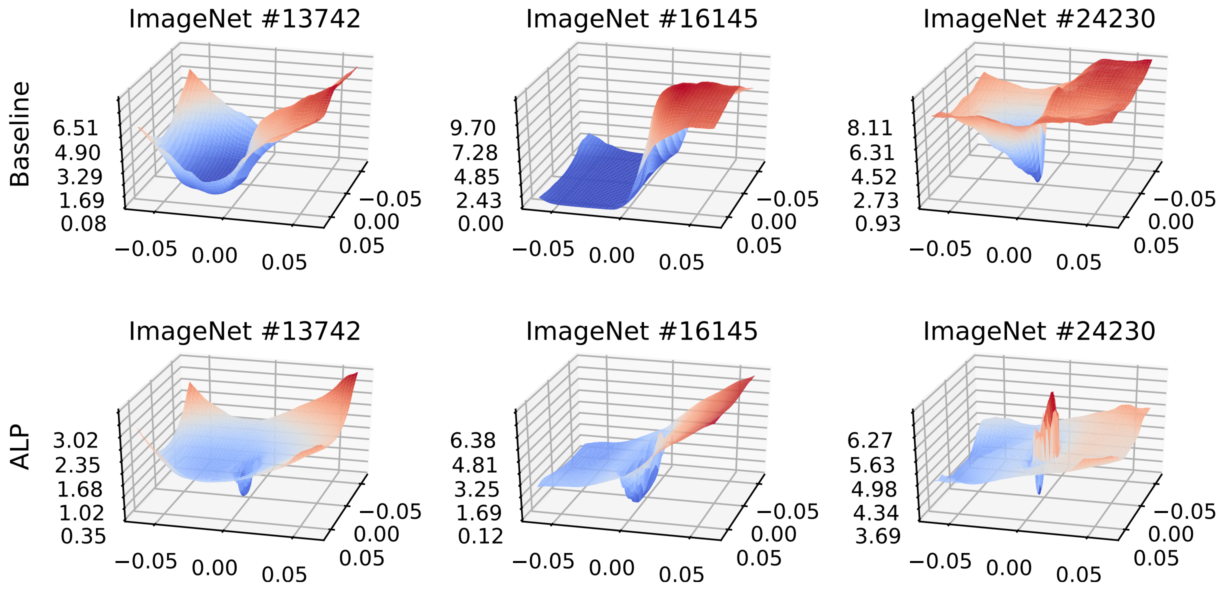

Figure 3: Comparison of loss landscapes of ALP-trained model and baseline model. Loss plots are generated by varying the input to the models, starting from an original input image chosen from the validation set. We see that ALP sometimes induces decreased loss near the input locally, and gives a "bumpier" optimization landscape. The z axis represents the loss. If $\hat{x}$ is the original input, then we plot the loss varying along the space determined by two vectors: $r_1 := \text{sign}(\nabla_x f(x))$ and $r_2 :=\sim \text{Rademacher}(0.5)$. We thus plot the following function: $z = \text{loss}(x \cdot r_1 + y \cdot r_2)$. The classifier here takes inputs scaled to $[0, 1]$.

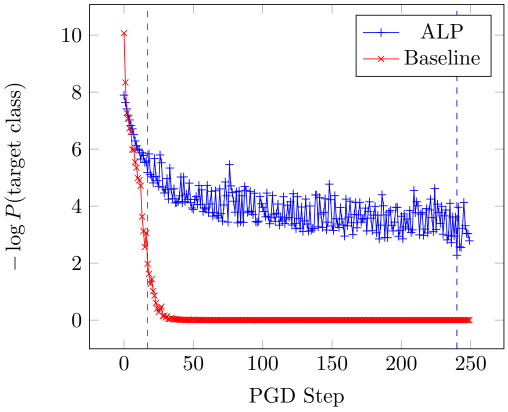

Figure 4: Comparison of targeted attack on ALP-trained model versus baseline model on a single data point, showing loss over PGD steps. Vertical lines denote the step at which the attack succeeded (in causing classification as the target class). The optimization process requires more gradient descent steps on the ALP model but still succeeds.

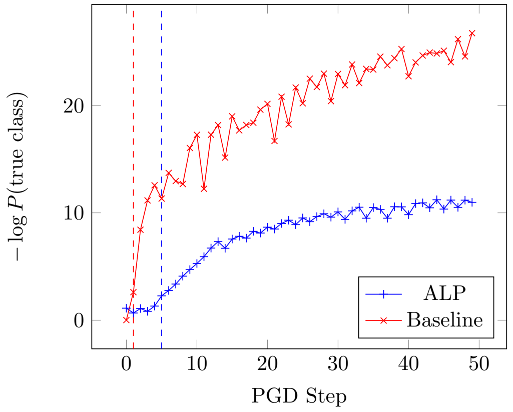

Figure 5: Comparison of untargeted attack on ALP-trained model versus baseline model on a single data point, showing loss over PGD steps. Vertical lines denote the step at which the attack succeeded (in causing misclassification). The optimization process requires more gradient descent steps on the ALP model but still succeeds.

References

[1] A. Athalye and N. Carlini. On the robustness of the CVPR 2018 white-box adversarial example defenses. *arXiv preprint*. URL https://arxiv.org/abs/1804.03286.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. URL https://arxiv.org/abs/1802.00420.

[3] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013.

[4] N. Carlini and D. Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[5] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *AISec*, 2017.

[6] N. Carlini and D. Wagner. Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*, 2017.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.

[8] W. He, J. Wei, X. Chen, N. Carlini, and D. Song. Adversarial example defenses: Ensembles of weak defenses are not strong. *arXiv preprint arXiv:1706.04701*, 2017.

[9] H. Kannan, A. Kurakin, and I. Goodfellow. Adversarial logit pairing. *arXiv preprint*. URL https://arxiv.org/abs/1803.06373.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*. URL https://arxiv.org/abs/1706.06083.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *ICLR*, 2013.

[12] J. Uesato, B. O'Donoghue, A. van den Oord, and P. Kohli. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018. URL https://arxiv.org/abs/1802.05666.