ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Breaking the Trade-off between Performance and Cost with an Improved Untargeted APGD

Ammar Al-Najjar[1], and Márk Jelasity[1,2] *

1- University of Szeged, Hungary
2- HUN-REN–SZTE Research Group on Artificial Intelligence, Hungary

Abstract. Auto-PGD (APGD), a component of AutoAttack [1], relies on two surrogate loss functions: cross entropy (CE) and Difference of Logits Ratio (DLR). AutoAttack includes untargeted APGD variants that are fast but less effective and targeted variants that are very effective but expensive. In this work, we introduce a low-cost untargeted variant of APGD that represents a substantial improvement over previous untargeted variants, and is almost as effective as the targeted APGD. Our simplest APGD variant begins with a CE-based initialization, and then transitions to DLR-based optimization. We then introduce an improvement to this idea, and propose a novel initialization approach. Our experimental results demonstrate that our hybrid attacks have a cost similar to standard untargeted APGD variants while they achieve attack success rates comparable to targeted APGD.

1 Introduction

Adversarial examples are small, carefully designed perturbations that cause a model to produce incorrect predictions, posing a significant challenge to the reliability of deep learning models [2]. Projected Gradient Descent (PGD) and its adaptive variants like APGD [1] are widely used both to generate such adversarial examples and to evaluate model robustness [3]. PGD attacks can be untargeted, when the ground-truth class loss is maximized, or targeted, when the loss of a different class is minimized. Targeted attacks are typically repeated with several targets; a very effective but expensive approach [1].

Here, we investigate the problem of whether the same effectiveness can be achieved with a single untargeted attack. With this goal we examine the different attack loss functions in related work. PGD-style attacks often use cross-entropy (CE) or the Difference of Logits Ratio (DLR) [1]. While CE provides strong initial gradients, DLR stabilizes the optimization by focusing on the relative difference between logits, making it invariant to shifts or rescaling of the output scores, an issue also highlighted in [4], where logit scaling can mask the true effective margin. Carlini and Wagner [5] also propose a loss similar to the DLR loss. Lin et al. [6] introduced the Minimal Difference (MD) loss, which instead prioritizes reducing any non-target logits that exceed the target logit, rather than directly maximizing the target logit itself. Weng et al. [7] show that carefully calibrating the logit margin between the target and non-target classes, using

*We thank the support by the the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

techniques such as temperature scaling and adaptive margin adjustment, can further improve the transferability of targeted adversarial attacks.

While most attacks use a single surrogate loss, e.g., the MOS-Attack [8] incorporates multiple losses simultaneously through a multi-objective (MOS) optimization framework. Here, instead, we experiment with much simpler hybrid loss strategies, where different losses are used sequentially. Our experiments show that our hybrid variants strengthen untargeted attacks, approximating targeted attack performance, with no extra computational cost.

2 Preliminaries

Let $z(x;\theta) \in \mathbb{R}^C$ denote the network logits (pre-softmax outputs) of a classifier for input $x \in \mathbb{R}^d$ and parameters θ, where C is the number of classes. Let the model class probabilities be $p(x;\theta) = \mathrm{softmax}(z(x;\theta)) \in [0,1]^C$. For ground-truth label $y \in \{1,\dots,C\}$, the cross-entropy loss is given by $\mathcal{L}_{\mathrm{CE}}(x,y;\theta) = -\log p_y(x;\theta)$.

Adversarial training optimizes the model against worst-case perturbations by solving

$$\min_{\theta} \mathbb{E}_{(x,y)\sim p_{\mathrm{data}}} \left[\max_{\|x'-x\|_p \le \varepsilon} \mathcal{L}(x',y;\theta) \right], \tag{1}$$

where $\mathcal{B}_p(x,\varepsilon) = \{x' \colon \|x'-x\|_p \le \varepsilon\}$ is the perturbation set.

The inner maximization is typically approximated by iterative first-order methods such as Projected Gradient Descent (PGD) [3] that projects the solution onto $\mathcal{B}_p(x,\varepsilon)$ after each update step. PGD runs for K iterations, optionally with R random restarts. In untargeted attacks, $\mathcal{L}$ is maximized to reduce confidence in the true class. In targeted attacks, a chosen target $t \neq y$ is promoted by maximizing an objective that encourages class t. This objective is often the negative classification loss of class t.

APGD extends PGD with adaptive step-size control, oscillation detection, and restarts [1]. The projection and restart mechanisms remain unchanged, but step sizes $\{\alpha_k\}$ are automatically adjusted. Standard APGD employs a single surrogate loss throughout the attack.

3 Hybrid APGD

We propose hybrid APGD attack variants that combine the CE and DLR losses, and we also propose a novel initialization approach. Other than this, we keep APGD unchanged [1]. Let us introduce some common notation and then describe our variants in detail.

In APGD [1], the attack is divided into two phases: an initialization phase and a main iterative optimization loop. The *initialization phase* executes a single Expectation over Transformation (EOT) loop prior to the main optimization. This computes an averaged gradient over multiple stochastic forward passes, providing a stable estimate of the expected gradient and loss values before entering

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

the iterative update loop, improving robustness against models with stochastic components. The *main iterative loop* performs multiple iterative updates to maximize a chosen surrogate loss. Each iteration follows the standard APGD update rule, projecting the perturbed input back onto the allowed perturbation set after each step.

Let π be the permutation that sorts logits in descending order, so $z_{\pi_1} \geq z_{\pi_2} \geq \cdots \geq z_{\pi_C}$. The *surrogate losses* used in untargeted APGD variants are the CE loss $\mathcal{L}_{\mathrm{CE}}$ and the *Untargeted DLR loss*

$$\mathcal{L}_{\mathrm{DLR}}(x, y) \;=\; -\,\frac{z_y(x) - \max_{i \neq y} z_i(x)}{z_{\pi_1}(x) - z_{\pi_3}(x)}. \tag{2}$$

Targeted APGD (APGD-T) uses the *Targeted DLR loss*

$$\mathcal{L}_{\mathrm{DLR-t}}(x, y, t) \;=\; -\,\frac{z_y(x) - z_t(x)}{z_{\pi_1}(x) - \frac{1}{2}\big(z_{\pi_3}(x) + z_{\pi_4}(x)\big)}, \tag{3}$$

where t is the target class. Both losses are *maximized* during an attack.

Variant A is our first proposed algorithm. It modifies the standard two-phase APGD by employing different surrogate losses in each phase: the cross-entropy (CE) loss is used during the initialization phase, and the untargeted DLR loss is used during the main optimization phase. In contrast, conventional APGD variants use a single surrogate loss for both phases.

Variant B is a similar approach with the loss-types swapped: the DLR loss is used in the initialization phase and CE in the main optimization phase.

Variant A+ includes an additional stabilization technique for *Variant A*. At the start of Phase II, prior to the standard untargeted DLR updates, a small number of preliminary steps are performed, in which we utilize the negated targeted DLR loss $-\mathcal{L}_{\mathrm{DLR-t}}(x, y, t)$. Maximizing this objective pushes the solution away from the given target t.

Let us describe this technique in more detail. Let us assume that the attack is executed on a batch of N samples and let $x_i^{(k)}$ denote sample i in update step k, where $1 \leq i \leq N$. The main idea is that at the beginning of each APGD update step, we assign a target class $t_i^{(k)}$ to $x_i^{(k)}$ for all $1 \leq i \leq N$. This target class is selected to be the rth most confident class, that is, $t_i^{(k)} = \pi_r(x_i^{(k)})$. In this work, we set $r = 10$, but we note that this hyperparameter would be worth analyzing in more detail. We then maximize the loss function

$$\sum_{i=1}^{N} -\mathcal{L}_{\mathrm{DLR-t}}(x_i^{(k)}, y_i, t_i^{(k)}) \tag{4}$$

in step k. This method is followed only in the first few initial steps until the selected targets become stable. We empirically observed that after a few iterations these targets stabilize, at which point we stop this technique and switch to the standard DLR-based untargeted APGD (APGD-DLR). Thus, the initial steps using this technique essentially work as initialization for APGD-DLR.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

To detect stabilization, we track the similarity of the selected targets to the targets of the previous round. In update k, we define $s_k = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}(t_n^{(k)} = t_n^{(k-1)})$, which is the probability that a target remains the same. We wish to detect the first large jump of s_k as k grows. To do this, we first define the finite difference $\Delta s_i = s_i - s_{i-1}$, and mean absolute difference $\overline{\Delta s}_{1:i} = \frac{1}{i}\sum_{j=1}^{i} |\Delta s_j|$. We identify the first large jump using the rule $\Delta s_k > \overline{\Delta s}_{1:k-1}$. At this point we switch to APGD-DLR, and we reuse the same k for all subsequent batches.

4 Experimental Results

Experiments used a single NVIDIA H100 GPU (80 GB, 1755 MHz) with no additional accelerators or distributed resources [9]. We evaluate pretrained robust WideResNets [10, 11] on CIFAR-10 [12], CIFAR-100 [12], SVHN [13], and Tiny-ImageNet [14]. We used a batch size of $N = 128$ and performed 100 gradient steps with a single restart. For the RobustBench [15] models, we employed five restarts. We collected execution time information in a controlled environment using a single GPU.

Table 1 contains our results. The leftmost column contains the attack type (norm and ϵ), clean accuracy, dataset and architecture. The rest of the columns contain results with a given untargeted attack, the same untargeted attack ensembled with APGD-T (the targeted APGD-DLR variant), and APGD-T on its own. In each column we report robust accuracy and attack time in seconds (in parenthesis).

The results show that *Variant A* yields a consistent improvement over standard APGD-DLR across every dataset and architecture, while preserving attack efficiency. The stabilization procedure in *Variant A+* provides modest further gains for every setting. In contrast, reversing the loss order (*Variant B*) is consistently weaker, confirming that CE is more effective for initialization and DLR is better suited for the main optimization phase. Overall, these findings support the value of mixing surrogate losses within APGD, demonstrating that a simple CE-to-DLR transition, optionally enhanced with targeted DLR refinement, improves attack strength without increasing computational cost. Also, the results suggest that it is worth replacing APGD-DLR with our *Variant A or A+* in the AutoAttack ensemble.

5 Conclusion

We introduced a hybrid APGD variant that combines a CE based initialization with DLR driven optimization, together with a lightweight stabilization step. Our experiments across multiple datasets and robust architectures show that this strategy (*Variant A+*) consistently strengthens untargeted APGD attacks without increasing computational cost. These results highlight that carefully allocating surrogate losses in APGD might increase effectiveness close to APGD-T, but with the cost of an untargeted APGD.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Table 1: Results with WideResNet models from [11] and RobustBench [15] models. See text for explanation. The best attack (or the cheapest if there is a tie) is indicated in bold.

Model Info.	**APGD-U**		**Ens. w. APGD-T**	**only APGD-T**
ℓ_∞, 8/255 92.44 CIFAR-10 WRN-28-10 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	70.28 (272.4) 68.23 (276.0) 67.67 (275.7) **67.44 (275.3)** 69.55 (274.4)	67.31 (2105.7) 67.27 (2091.8) 67.28 (2087.5) **67.25 (2079.8)** 67.30 (2114.2)	67.29 (1938.4)
ℓ_∞, 8/255 93.26 CIFAR-10 WRN-70-16 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	73.48 (1370.1) 71.61 (1373.6) 71.01 (1371.4) **70.74 (1372.2)** 72.88 (1375.4)	70.73 (10906.3) 70.68 (10802.8) 70.69 (10757.6) **70.65 (10743.4)** 70.72 (10897.8)	70.73 (9897.2)
ℓ_∞, 8/255 72.58 CIFAR-100 WRN-28-10 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	44.06 (212.9) 39.59 (215.9) 39.00 (215.1) **38.94 (215.4)** 43.01 (214.7)	38.82 (1282.3) 38.79 (1267.1) **38.76 (1258.4)** 38.79 (1258.4) 38.84 (1293.5)	38.81 (1241.2)
ℓ_∞, 8/255 75.22 CIFAR-100 WRN-70-16 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	48.12 (1105.9) 43.92 (1103.1) 43.14 (1103.4) **42.88 (1107.1)** 47.19 (1107.1)	42.67 (6859.1) 42.67 (6851.9) 42.67 (6794.3) **42.67 (6785.9)** 42.69 (6935.7)	42.70 (6343.1)
ℓ_∞, 8/255 95.56 SVHN WRN-28-10 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	68.27 (731.9) 66.21 (739.2) 65.11 (739.2) **64.72 (742.2)** 67.51 (735.7)	63.99 (5317.3) 63.99 (5290.8) 63.98 (5252.5) **63.98 (5231)** 63.98 (5319.7)	64.02 (4964.5)
ℓ_∞, 8/255 65.19 TinyImageNet WRN-28-10 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	37.43 (647.7) 32.52 (647.0) 31.44 (648.4) **31.4 (648.3)** 36.45 (648.4)	31.30 (3565.6) 31.29 (3489.2) 31.28 (3461.2) **31.28 (3458.6)** 31.30 (3553.2)	31.28 (3293.8)
ℓ_2, 128/255 95.16 CIFAR-10 WRN-28-10 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	84.45 (282.7) 83.86 (288.9) 83.68 (286.8) **83.65 (288)** 84.06 (285.0)	83.63 (2557.7) 83.63 (2561.8) **83.63 (2553.1)** 83.63 (2557) 83.63 (2565.1)	83.63 (2338.8)
ℓ_2, 128/255 95.54 CIFAR-10 WRN-70-16 [11]	APGD-CE: APGD-DLR: Variant-A: Variant-A+: Variant-B:	85.63 (1405.4) 85.20 (1415.3) 84.96 (1414.2) **84.89 (1413.1)** 85.30 (1411.3)	84.85 (12713.6) 84.86 (12721.8) 84.85 (12693.1) **84.85 (12685.4)** 84.86 (12728.2)	84.85 (11382.8)
ℓ_∞, 8/255, 91.58 CIFAR-10 WRN-A4 [16]	APGD-DLR: Variant-A: Variant-A+:	66.85 (6325.1) 66.30 (6342.8) **65.89 (6339.6)**	65.75 (16804.7) **65.74 (16124.1)** 65.75 (16834.5)	65.82 (11163.3)
ℓ_2, 0.5, 94.73 CIFAR-10 WRN-70-16 [17]	APGD-DLR: Variant-A: Variant-A+:	80.88 (6723.2) 80.61 (6753.7) **80.56 (6788.4)**	80.54 (14554.2) **80.52 (14512.1)** 80.53 (14578.5)	80.53 (12884.1)
ℓ_∞, 8/255, 75.11 CIFAR-100 WRN-70-16 [18]	APGD-DLR: Variant-A: Variant-A+:	45.10 (5248.8) 44.89 (5243.6) **44.67 (5224.9)**	**44.31 (13298.4)** 44.43 (13341.7) 44.39 (13265.7)	44.51 (8990.8)
ℓ_∞, 8/255, 62.40 CIFAR-100 WRN-28-10 [19]	APGD-DLR: Variant-A: Variant-A+:	33.42 (761.6) 32.43 (748.4) **32.24 (748.9)**	32.04 (1861.1) 32.06 (1822.5) **32.05 (1825.1)**	32.09 (1272.2)

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

References

[1] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in ICML, pp. 2206–2216, 2020.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in ICLR 2014.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in 6th ICLR, Vancouver, 2018.

[4] Z. Liu and A. B. Chan, "Boosting adversarial robustness from the perspective of effective margin regularization," in *British Machine Vision Conference (BMVC)*, 2022, pp. 1–3.

[5] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, May 22–26, 2017, pp. 39–57.

[6] W. Lin, K. Lucas, L. Bauer, M. K. Reiter, and M. Sharif, "Constrained gradient descent: A powerful and principled evasion attack against neural networks," in ICML 2022.

[7] J. Weng, Z. Luo, S. Li, N. Sebe, and Z. Zhong, "Logit margin matters: Improving transferable targeted adversarial attack by logit calibration," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3561–3574, 2023.

[8] P. Guo, C. Gong, X. Lin, F. Liu, Z. Lu, Q. Zhang, and Z. Wang, "MOS-Attack: A scalable multi-objective adversarial attack framework," in CVPR, pp. 5041–5051, 2025.

[9] NVIDIA Corporation, *NVIDIA H100 Tensor Core GPU Architecture*, 2022. Available: `https://resources.nvidia.com/en-us-hopper-architecture/nvidia-h100-tensor-c`. Accessed: Nov. 12, 2025.

[10] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2016, pp. 1–15.

[11] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, and S. Yan, "Better diffusion models further improve adversarial training," arXiv:2302.04638, 2023.

[12] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. of Comp. Sci., U. of Toronto, Toronto, ON, Canada, 2009.

[13] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in NIPS, 2011, pp. 1–9.

[14] Y. Le and X. Yang, "Tiny ImageNet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[15] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "RobustBench: a standardized adversarial robustness benchmark," in NeurIPS Datasets and Benchmarks, 2021

[16] Huang, S. and Lu, Z. and Deb, K. and Boddeti, V. N. "Revisiting residual networks for adversarial robustness." in CVPR, pp. 8202–8211, 2023.

[17] Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P., "Uncovering the limits of adversarial training against norm-bounded adversarial examples," arXiv:2010.03593, 2020.

[18] Amini, S., Teymoorianfard, M., Ma, S., and Houmansadr, A. "MeanSparse: Post-training robustness enhancement through mean-centered feature sparsification" arXiv:2406.05927, 2024.

[19] Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T., "Fixing data augmentation to improve adversarial robustness," arXiv:2103.01946, 2021.