# FIGHTING GRADIENTS WITH GRADIENTS: DYNAMIC DEFENSES AGAINST ADVERSARIAL ATTACKS

**Dequan Wang**[1], **An Ju**[1], **Evan Shelhamer**[2], **David Wagner**[1], **Trevor Darrell**[1]
dqwang@eecs.berkeley.edu, an_ju@berkeley.edu
UC Berkeley[1] Imaginary Number[2]

## ABSTRACT

Adversarial attacks optimize against models to defeat defenses. We argue that models should fight back, and optimize their defenses against attacks at test-time. Existing defenses are static, and stay the same once trained, even while attacks change. We propose a dynamic defense, defensive entropy minimization (dent), to adapt the model and input during testing by gradient optimization. Our dynamic defense adapts fully at test-time, without altering training, which makes it compatible with existing models and standard defenses. Dent improves robustness to attack by 20+ points absolute for state-of-the-art static defenses against AutoAttack on CIFAR-10 at $\epsilon_\infty = 8/255$.

## 1 INTRODUCTION: ATTACK, DEFEND, AND THEN?

Deep networks are vulnerable to adversarial attacks: input perturbations that alter natural data to cause errors or exploit predictions Szegedy et al. (2014). As deep networks are deployed in real systems, these attacks are real threats Yuan et al. (2019), and so defenses are needed. For every new empirical defense, a new attack follows, in a loop Tramer et al. (2020). The strongest attacks, armed with gradient optimization, adapt to circumvent defenses that do not. Their iterative updates form an even tighter loop to ensnare models that remain fixed during testing. In a cat and mouse game, the mouse must keep moving to survive.

Current defenses, deterministic or stochastic, stand still: once trained, they are *static* and do not adapt during testing. Adversarial training Goodfellow et al. (2014); Madry et al. (2018) learns from attacks during training, but suffers when the attacks differ during testing, such as by optimizing over larger perturbations or measuring distortion by a different norm. Stochastic defenses alter the network Dhillon et al. (2018) or input Guo et al. (2018); Cohen et al. (2019) during testing, but adaptive attacks can still optimize in expectation Athalye et al. (2018). *Static* defenses that stay the same are at a disadvantage against adaptive attacks that change.

Our *dynamic* defense fights adversarial gradients with defensive gradients to adapt during testing (Figure 1). These defensive gradients update the model and input transformations on every input,

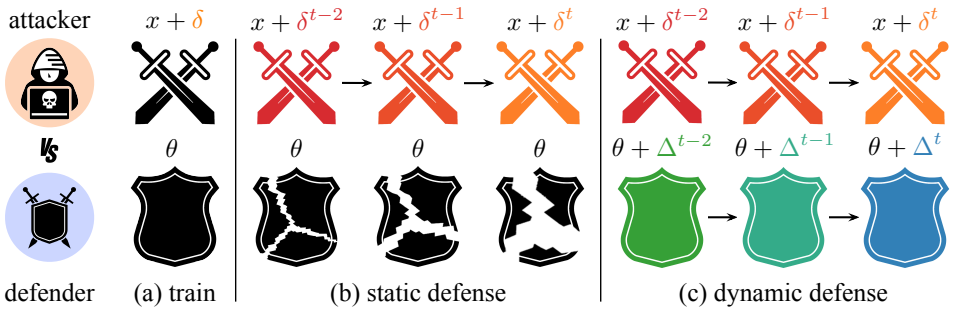


Figure 1: Attacks defeat defenses by optimization of the input at test-time. Adversarial training optimizes at train-time (a), but is static at test-time (b). We fight gradients with gradients by counter-optimization of model and input transformations. Our *dynamic* defense adapts during testing (c), so the attack cannot hit the same defense twice, to improve robustness.

natural or adversarial. Depending on the input makes the model a moving target that is more difficult to attack. Our method relies on gradients and batch statistics, inspired by domain adaptation approaches that update on test data Sun et al. (2020); Schneider et al. (2020); Liang et al. (2020a;b); Wang et al. (2021). Our defense objective is entropy minimization to maximize model confidence, so we call our method *dent* for defensive entropy. In pivoting from train-time to test-time, we equip the model to defend itself and keep changing, so the attacker never hits the same defense twice. By adapting, dent can change after the attack, and has the last move advantage.

Our experiments show that dent raises the accuracy of state-of-the-art static defenses on adversarial and natural data. We evaluate on CIFAR-10 against AutoAttack, including white-box and black-box attacks, and improve robustness in all cases. Dent defends models with or without adversarial training, and its model adaptation and input adaptation both help when jointly optimized for test-time defense. When combined with adversarial training, dent can adapt not just batch-wise but instance-wise, by optimizing a different defense for each input.

## 2 Dynamic Defense by Test-Time Adaptation

Defensive entropy minimization (dent) is a dynamic defense: it adapts to the data during testing. Dent does not alter training, and so it is compatible with existing models, and it can extend static defenses like adversarial training. For compatibility with our defense, we simply need the model to be differentiable for gradient optimization and probabilistic for entropy measurement. Therefore, we apply dent to models with a static defense (adversarial training), and also apply it to "bare" models (without any defense). We review preliminaries on adversarial attacks, describe the static defense of adversarial training, and explain adaptation with our dynamic defense.

### 2.1 Preliminaries

Let $x \in \mathbb{R}^d$ and $y \in \{1, \dots, C\}$ be an input and its corresponding ground truth. Given a model $f(\cdot; \theta) \colon \mathbb{R}^d \to \mathbb{R}^C$ parameterized by $\theta$, the goal of the adversary is to craft a perturbation $\delta \in \mathbb{R}^d$ such that the perturbed input $\tilde{x} = x + \delta$ causes a prediction error $f(x + \delta; \theta) \neq y$.

A targeted attack aims for a specific prediction of $y'$, while an untargeted attack seeks any incorrect prediction. The perturbation $\delta$ is constrained by a choice of $\ell_p$ norm and threshold $\epsilon$: $\{\delta \in \mathbb{R}^d \mid \|\delta\|_p < \epsilon\}$. We consider the two most popular norms for adversarial attacks: $\ell_\infty$ and $\ell_2$.

Adversarial training is a standard defense, formulated by Madry et al. (2018) as a saddle point problem,

$$\operatorname*{argmin}_\theta \mathbb{E}_{(x,y)} \max_\delta L(f(x + \delta; \theta), y), \tag{1}$$

which the model minimizes and the adversary maximizes with respect to the loss $L(\hat{y}, y)$, such as cross-entropy for classification. The adversary iteratively optimizes $\delta$ by projected gradient descent (PGD), a standard algorithm for constrained optimization, for each step $t$ via

$$\delta^t = \Pi_p(\delta^{t-1} + \alpha \cdot \operatorname{sign}(\nabla_{\delta^{t-1}} L(f(x + \delta^{t-1}; \theta), y))), \tag{2}$$

for projection $\Pi_p$ onto the norm ball for $\ell_p < \epsilon$, step size hyperparameter $\alpha$, and random initialization $\delta^0$. The model optimizes $\theta$ against $\delta$ to minimize the loss of its predictions on perturbed inputs. This is accomplished by augmenting the training set with adversarial inputs from PGD attack.

### 2.2 Defensive optimization

As the adversary optimizes its perturbation $\delta$, our dynamic defense optimizes its model adaptation $\Delta$ and input transformation $\Sigma$. Our defense is dynamic because both $\Delta$ and $\Sigma$ depend on the input, whether natural $x$ or adversarial $x + \delta$. In contrast, static defenses depend only on the training data through the model parameters $\theta$, and stochastic defenses depend on randomness $z \sim \mathcal{P}_{\text{defense}}$ that is independent of $x, \delta$.

The purpose of dynamic defenses is to move when the adversary moves. When the adversary submits an attack $x + \delta^t$, the defense can counter with $\Sigma^t$, $\Delta^t$. In this way, the defense always has the last move, and therefore an advantage.

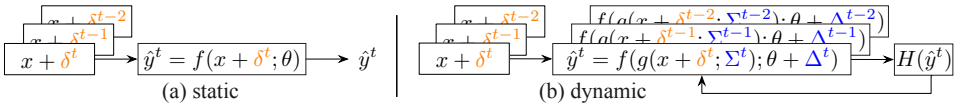


Figure 2: Against an adaptive attack, a defense must not remain static, but adapt in turn. The adversary iteratively optimizes its attacks $\delta^1, \cdots, \delta^t$ against the model $f$. A static defense (left) does not adapt, and may fail after enough attacks. Our dynamic defense (right) does adapt, and updates its defense parameters $\Delta, \Sigma$ every time the adversary updates its attack $\delta$.

Note that our optimization-based defense does not fit in the framework of obfuscated gradients Athalye et al. (2018). Our defense does not rely on (1) shattered gradients, as the update does not cause non-differentiability or numerical instability; (2) stochastic gradients, as the update is deterministically computed from the input, model, and prior updates; nor (3) exploding/vanishing gradients, as the update improves robustness with even a single step (although more steps do improve further in practice). Rather the updates $\Sigma^t$, $\Delta^t$ change at every step $t$, such that the adversary is always one step behind. This could be considered a *stale* gradient, as the attack $\delta^t$ is optimized against to $\Sigma^{t-1}$, $\Delta^{t-1}$, while the defense is adapted by $\Sigma^t$, $\Delta^t$.

**Defense objective** Test-time optimization requires an unsupervised objective. Following tent Wang et al. (2021), we adopt entropy minimization as our adaptation objective. Specifically, our defense objective is to minimize the Shannon entropy Shannon (1948) $H(\hat{y})$ of the model prediction during testing $\hat{y} = f(x; \theta)$ for the probability $\hat{y}_c$ of class $c$: $H(\hat{y}) = -\sum_{c \in 1,\dots,C} p(\hat{y}_c) \log p(\hat{y}_c)$.

For defense, this objective and its gradients are computed over batches. Batches are needed because entropy minimization can have degenerate solutions for a single prediction (such as predicting any class with probability one).

**Defense parameters** Dent updates input transformation $\Sigma$ and model adaptation $\Delta$. For the input, dent updates Gaussian smoothing by gradient optimization to control the degree of blurring, with parameter $\Sigma$. To blur dynamically and efficiently, the Gaussian filter size is adjusted on the fly Shelhamer et al. (2019). While only a single parameter, it can have a strong effect on the local statistics of the input. Furthermore, existing defenses motivate static smoothing against attacks Guo et al. (2018); Cohen et al. (2019), which we now make dynamic. For the model, dent adapts affine scale $\gamma$ and shift $\beta$ parameters by gradient optimization and adapts mean $\mu$ and variance $\sigma^2$ statistics by estimation. These are a small portion of the full model parameters $\theta$ (¡1% for ResNet-50 for example), concerning only the batch normalization layers Ioffe & Szegedy (2015). However, they have proven effective for conditioning a model on changes in the task Perez et al. (2018) or data Schneider et al. (2020); Wang et al. (2021).

By default the scale $\gamma$ and shift $\beta$ parameters are shared across inputs, and so adaptation updates batch-wise. For further adaptation, dent can update instance-wise, with different affine parameters for each input. In this way it adapts more than prior test-time adaptation methods with batch-wise updates Wang et al. (2021); Schneider et al. (2020).

Our adaptation of the input and model is a novel joint defense. Both transformations are differentiable, so end-to-end optimization coordinates them against attacks as layered defenses. This coordination is inspired by domain adaptation, but dent differs in its purpose and its unified loss. For domain adaptation, CyCADA Hoffman et al. (2018) also optimizes input and model transformations but does so in parallel with separate losses. Our defensive optimization is joint and shares the same loss.

**Defense updates** In summary, the parameters of the model $f$ and smoothing $g$ are updated by $\operatorname{argmin}_{\Sigma,\Delta} H(f(g(x + \delta; \Sigma); \theta + \Delta))$, through test-time optimization. At each iteration, we first estimate the normalization statistics $\mu$, $\sigma$ and then update the transformation parameters $\gamma$, $\beta$, $\Sigma$ by the gradient of entropy minimization.

When the adversary attacks with a perturbation $\delta^t$, our dynamic defense reacts with its own $\Sigma^t, \Delta^t$. In this way dent keeps pace with the attack and always has the last move. Figure 2 shows how standard static defenses do not update while dynamic defenses like dent do. Between batches, the defense parameters are reset.

| Accuracy(%) | Natural | | Adversarial | |
|---|---|---|---|---|
| | static | dent | static | dent |
| batch-wise $\Delta$: $[\gamma, \beta] \times 1$(shared) | | | | |
| Carmon et al. (2019) | 89.7 | 90.8 | 59.5 | 74.7 |
| Ding et al. (2020) | 84.4 | 87.6 | 41.4 | 47.6 |
| instance-wise $\Delta$: $[\gamma, \beta] \times$ batch size | | | | |
| Carmon et al. (2019) | 89.7 | 89.3 | 59.5 | 82.3 |
| Ding et al. (2020) | 84.4 | 84.7 | 41.4 | 64.4 |

Table 1: Dent boosts the accuracy of state-of-the-art static defenses.

| $\Delta$ | $\Sigma$ | Step | Time | Natural | Adversarial | |
|---|---|---|---|---|---|---|
| | | | | | $\epsilon_\infty = \frac{1.5}{255}$ | $\epsilon_2 = 0.2$ |
| $\times$ | None | 0 | $1.0\times$ | 95.6 | 8.8 | 9.2 |
| $\surd$ | None | 1 | $3.6\times$ | 95.6 | 15.0 | 13.5 |
| $\times$ | Stat. | 0 | $1.0\times$ | 86.2 | 25.8 | 23.6 |
| $\surd$ | Stat. | 1 | $3.6\times$ | 86.3 | 27.5 | 24.4 |
| $\surd$ | Stat. | 10 | $25.9\times$ | 86.3 | 37.6 | 30.9 |
| $\surd$ | Dyna. | 10 | $26.1\times$ | 92.5 | 45.4 | 36.5 |

Table 2: Ablation of dent's model ($\Delta$) and input ($\Sigma$) adaptation on a model without adversarial training.

## 3 Experiments

We evaluate the effectiveness of dent for dynamic defense across multiple choices of attack and base static defense. For attacks, we make use of AutoAttack Croce & Hein (2020), which includes four different white-box (gradient) and black-box attacks, and report the worst across all attacks. For static defenses to extend, we consider bare models without a defense, and robust models with adversarial training, as it is the most resilient. For architectures, we experiment with residual networks and their wide variants He et al. (2016); Zagoruyko & Komodakis (2016). For datasets, we evaluate on CIFAR-10 Krizhevsky (2009) as the most popular benchmark for adversarial robustness.

### 3.1 Dynamic Defense Fortifies Static Defense

We extend static adversarial training defenses with dent. Compared to "bare" models without defense, the static defense of adversarial training achieves higher adversarial accuracy but lower natural accuracy. Dent boosts adversarial accuracy further while reducing the natural accuracy gap.

**Dent improves state-of-the-art static defenses.** Table 1 shows static and dynamic results on CIFAR-10. Dent improves in every case on adversarial accuracy, while improving or maintaining natural accuracy. Dent adapts for 30/6 steps for batch/instance-wise model adaptation (without input adaptation for these experiments, as we found this to conflict with adversarial training). Instance-wise adaptation is more robust and more efficient.

### 3.2 Dynamic Defense Helps without Static Defense

Dent does not alter model training or architecture, so it applies to various models at test time. For instance, it does not assume adversarial training or a static defense of any kind. We demonstrate that dent significantly improves the adversarial accuracy of standard, off-the-shelf models without defenses. For these experiments, we evaluate against $\ell_\infty$ and $\ell_2$ norm-bounded attack on CIFAR-10. As the standard models have no static defense, we constrain the adversaries to smaller $\epsilon$ perturbations.

**Dent defends models without a static defense.** Table 2 inspects how each part of dent improves adversarial accuracy and natural accuracy. When applying sample-agnostic dent to standard models without defenses, the model transformation with $\Delta$ is further helped by input transformation with $\Sigma$. It already improves the adversarial accuracy from 8.8% to 15.0% against $\ell_\infty$ attacks with just a single step. With 10 steps and the dynamic Gaussian defense, we can further improve the model's adversarial accuracy from 8.8% to 45.4% against $\ell_\infty$ attacks and from 9.2% to 36.5% against $\ell_2$ attacks, achieving a significant boost of adversarial robustness with an acceptable sacrifice of natural accuracy and no alteration of training (or re-training).

**Dynamic input adaptation preserves natural accuracy.** Gaussian blur significantly improves adversarial accuracy. This agrees with prior work on denoising by optimization Guo et al. (2018) or randomized smoothing Cohen et al. (2019). When tuned as a fixed hyperparameter, Gaussian blur helps adversarial accuracy but hurts natural accuracy. In contrast, optimizing the Gaussian blur not only improves adversarial accuracy, but also significantly reduces the side effect of natural accuracy loss. Our dynamic Gaussian defense achieves $92.5\%$ natural accuracy, which is comparable to the $95.6\%$ accuracy of the standard model without adversarial training. It does so by test-time adaptation to the data: on natural data, the learned $\Sigma$ for the blur decreases to blur less and approximates the identity transformation.

## REFERENCES


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

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *ICLR*, 2018.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020a.

Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *arXiv preprint arXiv:2012.07297*, 2020b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020.

C.E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

Evan Shelhamer, Dequan Wang, and Trevor Darrell. Blurring the line between structure and learning to optimize and adapt receptive fields. *arXiv preprint arXiv:1904.11487*, 2019.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. In *ICML*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *TNNLS*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.