Improved Adversarial Robustness via Uncertainty Targeted Attacks

Gilberto Manunza [*1] Matteo Pagliardini [*1] Martin Jaggi [1] Tatjana Chavdarova [1]

Abstract

Deploying deep learning models in real-world applications often raises security and reliability concerns, due to their sensitivity to small input perturbations. While adversarial training methods aim at training more robust models, these techniques often result in a lower unperturbed (clean) test accuracy, including the most widely used Projected Gradient Descent (PGD) method. Furthermore, fast adversarial training methods often overfit the specific perturbation used during training. In this work, we propose *uncertainty-targeted attacks* (UTA), where the perturbations are obtained by maximizing the model's estimated uncertainty. We demonstrate on MNIST and CIFAR-10 that this approach—when implemented both in image and latent space—does not drastically deteriorate the clean test accuracy relative to PGD, its fast variant does not suffer from catastrophic overfitting, and it is robust to PGD attacks.

1. Introduction

It has been shown that small perturbations added to the input can easily "fool" well-performing deep neural networks (DNNs) into making wrong predictions (Biggio et al., 2013; Szegedy et al., 2014), which limits their application in real-world tasks due to security risks. The goal of *adversarial training* (AT) methods is improving the robustness to small perturbations of a trained classifier $\mathcal{C}_{\boldsymbol{\omega}} : \boldsymbol{x} \mapsto \hat{\boldsymbol{y}}$, with $\boldsymbol{x} \in \mathbb{R}^d$ denoting a data sample of finite dataset $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$ drawn from the data distribution p_d, $\hat{\boldsymbol{y}} \in \mathbb{R}^c$ its prediction of c possible classes, and $\boldsymbol{\omega} \in \mathbb{R}^m$ the parameters of the model. AT methods *directly* target this weakness of DNNs by training the model with *modified training samples* as follows:

$$\min_{\boldsymbol{\omega}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_d}[\max_{\boldsymbol{\delta}\in\Delta} \mathcal{L}(\mathcal{C}_{\boldsymbol{\omega}}(\boldsymbol{x}+\boldsymbol{\delta}), \boldsymbol{y})] \,, \qquad \text{(AT)}$$

where $\mathcal{L}$ denotes the loss function, and the added *worst-case* perturbation $\boldsymbol{\delta} \in \mathbb{R}^d$ is constrained to a small region around the sample $\boldsymbol{x}$, typically a ball $\Delta \triangleq [-\varepsilon, \varepsilon]^d$ with $\varepsilon > 0$. Popular implementations of the (AT) objective are the *projected gradient descent* (PGD, Madry et al., 2018)—where the inner maximization step is implemented with k gradient ascent steps, and its "fast variants"—which take only one step to compute the perturbation $\boldsymbol{\delta}$, e.g., *fast gradient sign method* (FGSM, Goodfellow et al., 2015), see § 2. However, further empirical studies showed that AT methods often reduce the average accuracy on "clean" unperturbed test samples, indicating the two objectives—robustness and clean accuracy—might be competing (Tsipras et al., 2019; Su et al., 2018). Moreover, Wong et al. (2020) further pointed out a phenomenon referred to as *catastrophic overfitting* where the robustness of fast AT methods rapidly drops to almost zero, within a *single* training epoch, see § 4.

A separate line of work aims at increasing the interpretability of the model by associating to each of its predictions an *uncertainty estimate* (Kim et al., 2016; Doshi-Velez and Kim, 2017). Two main uncertainty types in machine learning are considered: (i) *aleatoric*–describing the *noise* inherent in the observations, as well as (ii) *epistemic*–uncertainty originating *from the model*. While the former *cannot* be reduced, the latter arises due to insufficient data to train the model, and it *can* be explained away given enough data. In the context of classification, apart from capturing high-uncertainty due to overlapping regions of different classes, the epistemic uncertainty also captures which regions of the data space are not "visited" by the training samples.

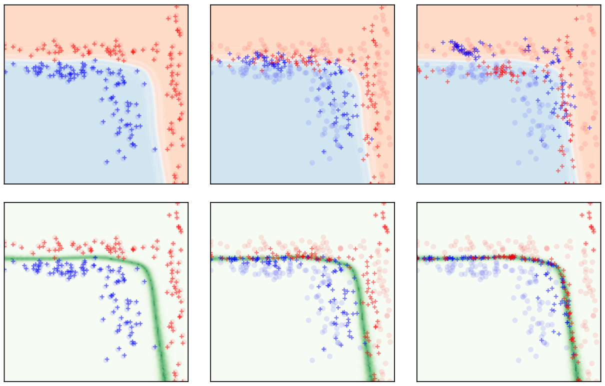
In this work we consider "uncertainty targeted attacks" (UTA), motivated by the insights that (i) as standard AT methods find a permutation $\boldsymbol{\delta}$ which maximizies the loss, the perturbed sample $\tilde{\boldsymbol{x}} \triangleq \boldsymbol{x} + \boldsymbol{\delta}$ is moved toward the decision boundary; and (ii) an uncertainty maximizing perturbation would move $\tilde{\boldsymbol{x}}$ either towards the decision boundary or toward non-visitied regions in data space, depending on the proximity. More precisely, we investigate if finding a perturbation $\boldsymbol{\delta}$ which *maximizes the model's estimated uncertainty* can provide a better trade-off between generalization and robustness, as well as improve the reported problem of catastrophic overfitting.

Related work: AT & maximum entropy. Stutz et al. (2019) show that adversarial examples *leave* the data mani-

[*] Equal contribution [1] EPFL. Correspondence to: Correspondence to: <firstname.lastname@epfl.ch>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

fold and that on-manifold adversarial training boosts generalization on synthetic datasets. The authors thus propose to use perturbations in the latent space of a VAE-GAN (Larsen et al., 2016; Rosca et al., 2017). Similarly, several methods traverse the latent space to find data samples that are misclassified (Baluja and Fischer, 2017; Song et al., 2018; Xiao et al., 2018; Zhang et al., 2020). In the context of standard classification, Pereyra et al. (2017) penalize the confident predictions by adding a regularizer that maximizes the entropy of the output distribution. Moreover, in the context of adversarial training, the results of (Cubuk et al., 2017, §3.2) indicate that adding such a regularizer improves the model robustness to PGD attacks.



(a) 1 step (b) 10 steps (c) 20 steps

Figure 1: 2D experiment: **PGD** (top) Vs. **UTA** (bottom) attacks to *the same trained classifier*, for varying number of steps (columns). ∘ and + depict the clean and perturbed samples, resp., and their color blue/red depicts their class. *Top row:* background depicts the assigned probability to each class of the attacked model, including its decision boundary. *Bottom row:* background depicts the uncertainty estimates of 10 model ensemble, where darker green is higher. As PGD is maximizing the cross-entropy loss, the perturbed sample can change its class—see (b) & (c), top—requiring smaller ε-ball, and on the other hand, small ε will imply reduced robustness around samples that are on a larger distance from the opposite class. See § 3.1 for discussion.

Related work: uncertainty estimation. While standard DNN training performs a maximum likelihood estimation of the parameters $\boldsymbol{\omega} \in \Omega$, training Bayesian Neural Networks (BNNs) extends to estimating the posterior distribution, providing a mathematically grounded framework for uncertainty. However, due to the integration with respect to the whole parameter space Ω, BNNs come at a prohibitive computational cost that is often intractable for DNNs. A popular epistemic uncertainty estimation method is *deep ensembles* (Lakshminarayanan et al., 2017), which trains a large number of models on the dataset and combines their predictions to estimate a predictive distribution over the weights. Gal and Ghahramani further show that *Dropout* (Srivastava et al., 2014) when applied to a neural network approximates Bayesian inference of a Gaussian processes (Rasmussen and Williams, 2005). The proposed *MC Dropout*—which applies Dropout at inference time—allows for computationally efficient uncertainty estimation.

Overview of contributions. We propose *uncertainty-targeted attacks* (UTA), where the perturbations are obtained by maximizing the model's estimated uncertainty. Our 2D illustrative example shows that UTA perturbations have the advantage of on average *decreased* mislabeled perturbed samples, see Fig. 1 and § 3.1, which could explain the reduced clean accuracy of the standard loss-based adversarial methods. The presented preliminary results on MNIST and CIFAR-10 show that this approach, when implemented *either in image or latent space*: (i) does not drastically decrease the clean test accuracy relative to PGD, (ii) its fast variant does not suffer from catastrophic overfitting, (iii) and it is robust to PGD attacks.

2. Preliminaries

Adversarial training. The inner maximization problem of Eq. AT can be implemented in several ways. As in general the optimization is non-convex, Lyu et al. (2015) propose approximating the inner maximization problem with Taylor expansion and then applying Lagrangian multiplier. For ℓ_∞ bounded attacks, this linearization yields the FGSM (Goodfellow et al., 2015), with its perturbation defined as:

$$\boldsymbol{\delta}_{\text{FGSM}} \triangleq \varepsilon \cdot \text{Sign}\big(\nabla_{\boldsymbol{x}} \mathcal{L}(\mathcal{C}_{\boldsymbol{\omega}}(\boldsymbol{x}), \boldsymbol{y})\big)\,, \qquad \text{(FGSM)}$$

where Sign$(\cdot)$ denotes the sign function. To improve the catastrophic overfitting of FGSM, Wong et al. (2020) propose adding a random vector ξ to FGSM as follows:

$$\boldsymbol{\delta}_{\text{R-FGSM}} \triangleq \prod_{||\cdot||_\infty \le \varepsilon} \Big(\xi + \alpha \cdot \text{Sign}\big(\nabla_{\boldsymbol{x}} \mathcal{L}(\mathcal{C}_{\boldsymbol{\omega}}(\boldsymbol{x}), \boldsymbol{y})\big)\Big)\,, \qquad \text{(R-FGSM)}$$

where $\xi \sim U([-\varepsilon, \varepsilon]^d)$, $\alpha \in [0, 1]$ is selected step size, and Π is projection on the ℓ_∞–ball. The PGD method (Madry et al., 2018) applies FGSM for $i = 1, \dots, k$ steps:

$$\boldsymbol{\delta}^i_{\text{PGD}} \triangleq \prod_{||\cdot||_\infty \le \varepsilon} \Big(\alpha \cdot \text{Sign}\big(\nabla_{\boldsymbol{x}} \mathcal{L}(\mathcal{C}_{\boldsymbol{\omega}}(\boldsymbol{x} + \boldsymbol{\delta}^{i-1}_{\text{PGD}}), \boldsymbol{y})\big)\Big)\,. \qquad \text{(PGD)}$$

PGD with k steps is often referred to as PGD-k.

Uncertainty estimation. We use *MC Dropout* to sample M models $\{\mathcal{C}^{(m)}_{\boldsymbol{\omega}}\}^M_{m=1}$, with the ensemble's prediction $\hat{\boldsymbol{y}} \in \mathbb{R}^C$ defined as the average prediction:

$$\hat{\boldsymbol{y}} = \frac{1}{M} \sum_{m=1}^{M} \text{Softmax}\big(\mathcal{C}^{(m)}_{\boldsymbol{\omega}}(\boldsymbol{x})\big)\,.$$

Finally, we use the *entropy* of the output distribution to quantify the uncertainty estimate of a given sample $\boldsymbol{x}$:

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{\omega}) = -\sum_{c \in C} \hat{\boldsymbol{y}}_c \log \hat{\boldsymbol{y}}_c \,. \tag{E}$$

3. Uncertainty targeted attacks (UTA)

Unlike standard loss-based attacks, we propose uncertainty-guided exploration in either the data or the latent space. Uncertainty Targeted Attacks (UTA) aim at finding perturbations which *maximize the uncertainty estimate*:

$$\begin{aligned} &\min_{\boldsymbol{\omega}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_d}[\mathcal{L}(\mathcal{C}_{\boldsymbol{\omega}}(\mathcal{E}(\boldsymbol{x}) + \boldsymbol{\delta}_u), \boldsymbol{y})] \\ \text{s.t.} \quad &\boldsymbol{\delta}_u = \arg\max_{\boldsymbol{\delta}\in\Delta} \mathcal{H}(\mathcal{E}(\boldsymbol{x}) + \boldsymbol{\delta}, \boldsymbol{\omega}) \,, \end{aligned} \tag{UTA}$$

where the encoder $\mathcal{E}$ is the identity when perturbations are applied in the input space. The above formulation also applies for latent space UTA perturbations, where with abuse of notation, the model can be seen as having an encoder part $\mathcal{E} : \boldsymbol{x} \mapsto \boldsymbol{z}$ and a classification part $\mathcal{C}_{\boldsymbol{\omega}} : \boldsymbol{z} \mapsto \boldsymbol{y}$, and in that case $\boldsymbol{\delta} \in \mathbb{R}^l$, where l is the dimension of the latent space.

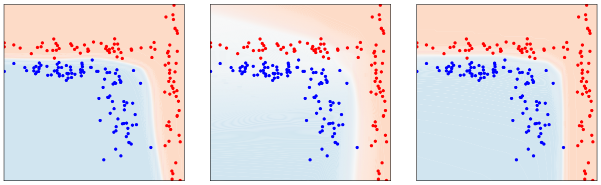
Similar to AT, the inner maximization of UTA—for both image and latent-space perturbations—can be implemented analogously to PGD and FGSM–see App. A, referred below as *UTA* and *UTA with one step*, resp.

3.1. Motivating example

Fig. 1 depicts a toy experiment with non-isotropic distance between samples of opposite class in $\mathbb{R}^2$. For PGD we observe that when the size of the ball ε is: (i) *small*: PGD does not improve the robustness around opposite class samples that are widely separated; (ii) *large*: PGD can perturb the sample by moving it on the opposite side of the boundary–resulting in mislabeled samples. Thus, to achieve as good robustness-generalization trade-off as possible, ε should be carefully selected for PGD. On the other hand, we observe that UTA is relatively less sensitive to the choice of ε. See Fig. 6 for such analysis on CIFAR-10. While Fig. 1 illustrates the difference between attacks for a *fixed model*, Fig. 2 illustrates the decision boundaries obtained with standard training, PGD, as well as UTA.

3.2. Advantages of UTA

In high dimensional space, such non-isotropic margins as in § 3.1 are more likely to occur, what could explain why more robust models trained with PGD on average degrade the clean test accuracy. Uncertainty based perturbations on the other hand, are only sensitive to the choice of ε at the beginning when the uncertainty guides the data space exploration, and are less likely to wrongfully label a perturbed sample that does not lie on the decision boundary. In addition, relative to standard AT methods, UTA: (i) is *unsupervised*–does not require the ground truth labels y, and can thus be extended to unsupervised methods which output probability estimates, and (ii) includes more general perturbations as, depending on the proximity of the data point at hand, will either move it toward the decision boundary, or toward "unexplored" regions of the training set.



(a) Baseline (b) PGD-AT (c) UTA-AT

Figure 2: Decision boundary obtained from: (a) regular training, (b) PGD with a large ε, $k = 15$, and a $\alpha = 0.05$, (c) UTA using a large ε, $k = 15$, and a $\alpha = 0.05$. We observe in (b) that some data points are misclassified in some regions, and well-classified in other regions, indicating that finding a value of ε that does good robustness/generalization trade-off *globally* is difficult for PGD. See § 3.1 & 3.2 for discussion.

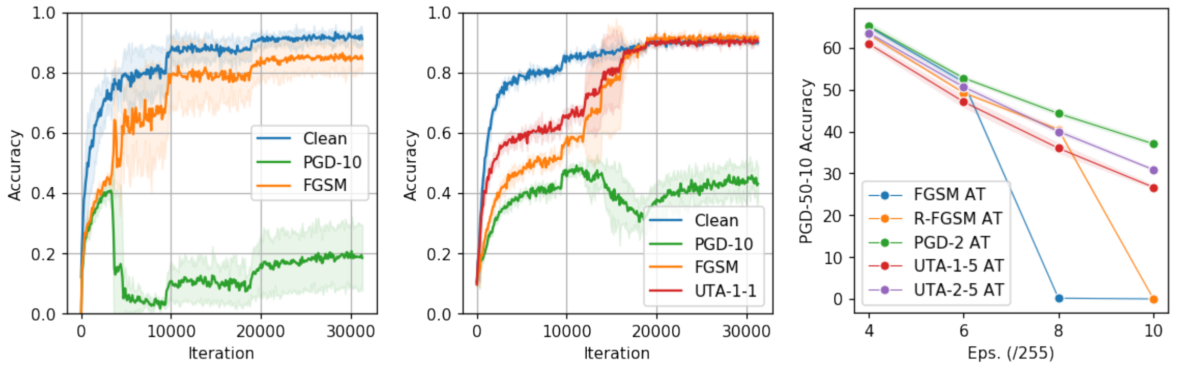
4. Experiments

Setup & methods. We evaluate on MNIST (Lecun and Cortes, 1998) and CIFAR-10 (Krizhevsky, 2009). We denote as *UTA-k-M*, when performing k steps of UTA, and sampling M models using MC-dropout. See App. A for details on the implementation. As in (Andriushchenko and Flammarion, 2020) we also evaluate against PGD with *random restarts* which as in R-FGSM adds random perturbation, restarts 10 times and finally selects the strongest perturbation, denoted as *PGD-50-10*.

4.1. Image space experiments

Since Babu (2020) observe that adding a Dropout layer after each convolutional layer helps to stabilize FGSM we use the identical setting of MC Dropout with $p = 0.2$ for all methods. Fig.3a and 3b depict the results on CIFAR-10 which evaluate if the methods are prone to Catastrophic Overfitting (CO). Relative to FGSM, UTA-1-1 notably improves CO, as although there is an accuracy drop relatively later, it does not reduce to 0.

Fig. 3c depicts the robustness to PGD-50-10 attacks of various training methods for $\varepsilon \in \{\frac{4}{255}, \frac{6}{255}, \frac{8}{255}, \frac{10}{255}\}$, including the R-FGSM and PGD-2 strong baselines. We compare these test-time attacks against UTA-1-5 and UTA-2-5 training with 5 sampled models. Although UTA perturbations are weaker in terms of PGD-50-10 robustness, relative to R-FGSM and PGD-2, they suffer less from CO and show competitive PGD-50-10 robustness even for large values of ε. In particular, for UTA-1-5 and UTA-2-5 we observe that



(a) training with FGSM

(b) training with UTA

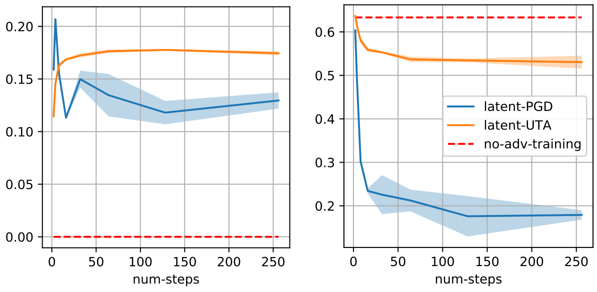
(c) testing with PGD-50-10

Figure 3: Catastrophic overfitting (CO) on CIFAR-10 using ResNet 18, averaged over 3 runs. (a): training with FGSM–with step size $\alpha = 8/255$, $\varepsilon = \alpha$; and testing against PGD–10 with $\varepsilon = 8/255$ and $\alpha = \varepsilon/4$. CO occurs at around iteration 4700. (b): training with UTA with 1 step (fast version), 1 sampled model and $\alpha = \varepsilon = 8/255$; testing against PGD-10 ($\varepsilon = 8/255$, $\alpha = \varepsilon/4$) and FGSM ($\varepsilon = \alpha = 8/255$). We observe that UTA is more robust to CO relative to 3a. (c): PGD-50-10 comparison of different AT and UTA methods for different values of the perturbation radius ε. We observe that the UTA methods, albeit being marginally less robust to PGD-50-10, do not suffer from CO even for large values of ε.

even for $\varepsilon = \frac{10}{255}$ there is no CO, which is not the case for the R-FGSM attack.

4.2. Latent space experiments

We apply UTA in latent space on MNIST and CIFAR-10. For all experiments the encoder $\mathcal{E}(\boldsymbol{x})$ is a two-layer convolutional network, and the classifier part $\mathcal{C}_{\boldsymbol{\omega}}(\boldsymbol{z})$ is a multi-layer perceptron. For all experiments, $\mathcal{E}(\boldsymbol{x})$ is pre-trained and frozen. We compare UTA-k-10 with PGD-k for $k \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. In latent space, distances are arbitrary making the choice of ε itself arbitrary. In our experiments we set $\varepsilon = 5$. Fig. 4 shows on CIFAR-10 how latent-space PGD loses both its clean and robust (PGD-10) accuracy when the number of steps increases. On the other hand, UTA copes well with large k as it has a high clean accuracy,in line with the toy example introduced in § 3.1. We observe similar results on MNIST, see Fig. 5. Note that while PGD and UTA, when applied in latent space, are notably less robust against PGD-10 perturbations in image space—relative to when they are applied in image space—yet they improve the robustness relative to standard non-adversarial training.



(a) PGD-10 accuracy

(b) Clean accuracy

Figure 4: Robustness and generalization performance of PGD and UTA attacks in *latent space* after their full training, for a varying number of steps k (x-axis), on CIFAR-10. Dashed line depicts the performance of standard training (without adversarial training). See § 4.2 for discussion.

5. Discussion

Building on the well-established notion of uncertainty estimates, we proposed uncertainty-targeted attacks that perturb a training sample in a direction that maximizes the uncertainty of the model. Our preliminary results on MNIST and CIFAR-10, in input data and latent space, indicate that this approach is promising as it degrades less the clean test accuracy relative to PGD, it is more robust to PGD relative to standard training, and its one-step variant improves the reported catastrophic overfitting of FGSM. Interestingly, while this method does not directly target the adversarial training objective AT, it is nonetheless robust to standard AT methods.

As MC-Dropout is an approximate uncertainty estimation method, a potential direction includes exploring if using BNNs in some tractable setups could improve the performances of UTA in terms of robustness-generalization trade-off. More generally, it is promising to study if recently proposed methods of computing the DNN model's uncertainty such as (van Amersfoort et al., 2021) could further improve UTA methods.

References

M. Andriushchenko and N. Flammarion. Understanding and improving fast adversarial training. In *NeurIPS*, 2020.

R. V. Babu. Single-step adversarial training with dropout scheduling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 950–959, 2020.

S. Baluja and I. Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017.

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 *Machine Learning and Knowledge Discovery in Databases*, pages 387–402, 2013. ISBN 978-3-642-40994-3.

E. D. Cubuk, B. Zoph, S. S. Schoenholz, and Q. V. Le. Intriguing properties of adversarial examples, 2017.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608*, 2017.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, 2016.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Master's thesis, 2009.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc.

A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016.

Y. Lecun and C. Cortes. The MNIST database of handwritten digits. 1998. URL http://yann.lecun.com/exdb/mnist/.

C. Lyu, K. Huang, and H.-N. Liang. A unified gradient regularization family for adversarial examples. *arXiv:1511.06385*, 2015.

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*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv:1701.06548*, 2017.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed. Variational approaches for auto-encoding generative adversarial networks, 2017.

L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017. doi: 10.1109/WACV.2017.58.

Y. Song, R. Shu, N. Kushman, and S. Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/file/8cea559c47e4fbdb73b23e0223d04e79-Paper.pdf.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

D. Stutz, M. Hein, and B. Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6976–6987, 2019.

D. Su, H. Zhang, H. Chen, J. Yi, P. Chen, and Y. Gao. Is robustness the cost of accuracy? - A comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, 2018.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks, 2014.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2019.

J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression. *arXiv:2102.11409*, 2021.

E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

C. Xiao, B. Li, J. yan Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3905–3911, 7 2018. doi: 10.24963/ijcai.2018/543. URL https://doi.org/10.24963/ijcai.2018/543.

L. Zhang, M. Yu, T. Chen, Z. Shi, C. Bao, and K. Ma. Auxiliary training: Towards accurate and robust models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

A. Details on the implementation

A.1. UTA-PGD details

To solve the inner maximization problem of UTA, we implement a variation of PGD:

$$\boldsymbol{\delta}_{\mathrm{u}}^{i} \triangleq \prod_{||\cdot||_\infty \le \varepsilon} \Big(\alpha \cdot \mathrm{Sign}\big(\nabla_{\boldsymbol{x}} \mathcal{H}(\mathcal{E}(\boldsymbol{x}) + \boldsymbol{\delta}_{\mathrm{UTA}}^{i-1}, \boldsymbol{\omega})\big)\Big) . \tag{UTA-PGDA}$$

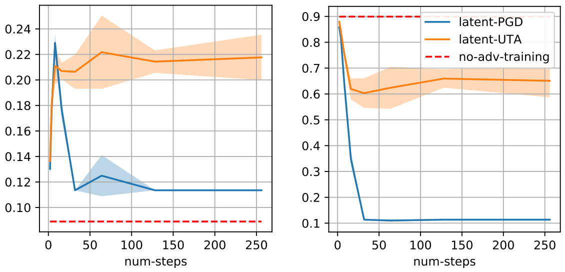
A.2. Input space experiments

Catastrophic overfitting experiments. We evaluate catastrophic overfitting on the CIFAR-10 (Krizhevsky, 2009) dataset. We use a ResNet18 (He et al., 2016) architecture, modified to accommodate the MC-dropout sampling procedure. The modification consists of adding a dropout layer with dropout probability $p = 0.2$ after each convolutional layer. In order to have a very fast version of the attack (same computational cost as FGSM) we use only one UTA step, and sample only one model with MC-dropout, e.g. $k = 1$ and $M = 1$. For Fig.3a and 3b, our models were trained for 200 epochs using the SGD optimizer with Nesterov momentum and with an initial learning rate of 0.1 decayed by factor of $\frac{1}{5}$ after $60, 120, 160$ epochs.

For Fig. 3c in order to reach faster convergence we trained a ResNet18 model for 90 epochs with MCD using a cyclic learning rate scheduling (Smith, 2017) with a maximum LR of 0.2 for the FGSM, R-FGSM and PGD-2 and of 0.1 for the UTA-1-5 and UTA-2-5 attacks.

B. Additional results

Fig. 5 shows results obtained applying PGD and UTA in latent space on the MNIST dataset. Fig. 6 shows differences of input perturbations on CIFAR-10 obtained with UTA and PGD with identical hyperparameters ($\alpha = 0.001$, $\varepsilon = \infty$) and a large $k = 1000$ steps.



(a) PGD-10 accuracy (b) Clean accuracy

Figure 5: MNIST results with PGD and UTA performed in latent space. The number of PGD and UTA steps is progressively increased.

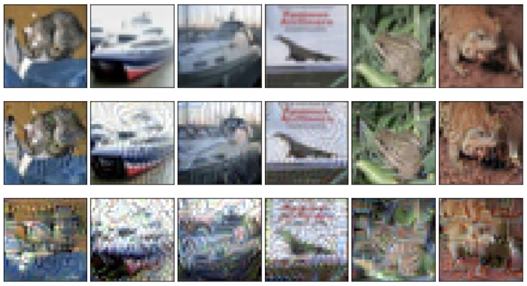


Figure 6: Illustration of the difference between PGD and UTA attacks to classifier trained on CIFAR-10: *(i)* top row: clean samples, *(ii)* middle row: UTA perturbations, *(iii)* bottom row: PGD attacks, where for UTA and PGD we use same setup (1000 steps, step size of 0.001, $\varepsilon = \infty$). We use large number of steps to verify empirically if the difference between UTA and PGD depicted in Fig. 1 holds on real-world datasets as well. Contrary to the PGD-perturbed samples, the correct class of the UTA-perturbed ones remains perceptible.