David J. Miller
Zhen Xiang
George Kesidis

Adversarial Learning and Secure AI

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

Chapter 04

Test-Time Evasion Attacks (Adversarial Inputs)

Outline

- "Adversarial inputs" a.k.a. TTEs
- Why do TTEs exist?
- TTE attack/defense scenarios
  - grey-box and white-box attacks
  - physical vs. digital attacks, attack feasibility
  - supervised and unsupervised defenses
- Existing TTEs based on ideas from
  - neural network inversion
  - discriminative learning

Outline (cont)

- “Robust" and "certified" defenses for TTEs
- Anomaly Detection (AD) of TTEs
- Background on generative modeling and GANs
- GANs-based AD against TTEs
- Experimental Results
- Post-detection actions

Adversarial Inputs at Test-Time

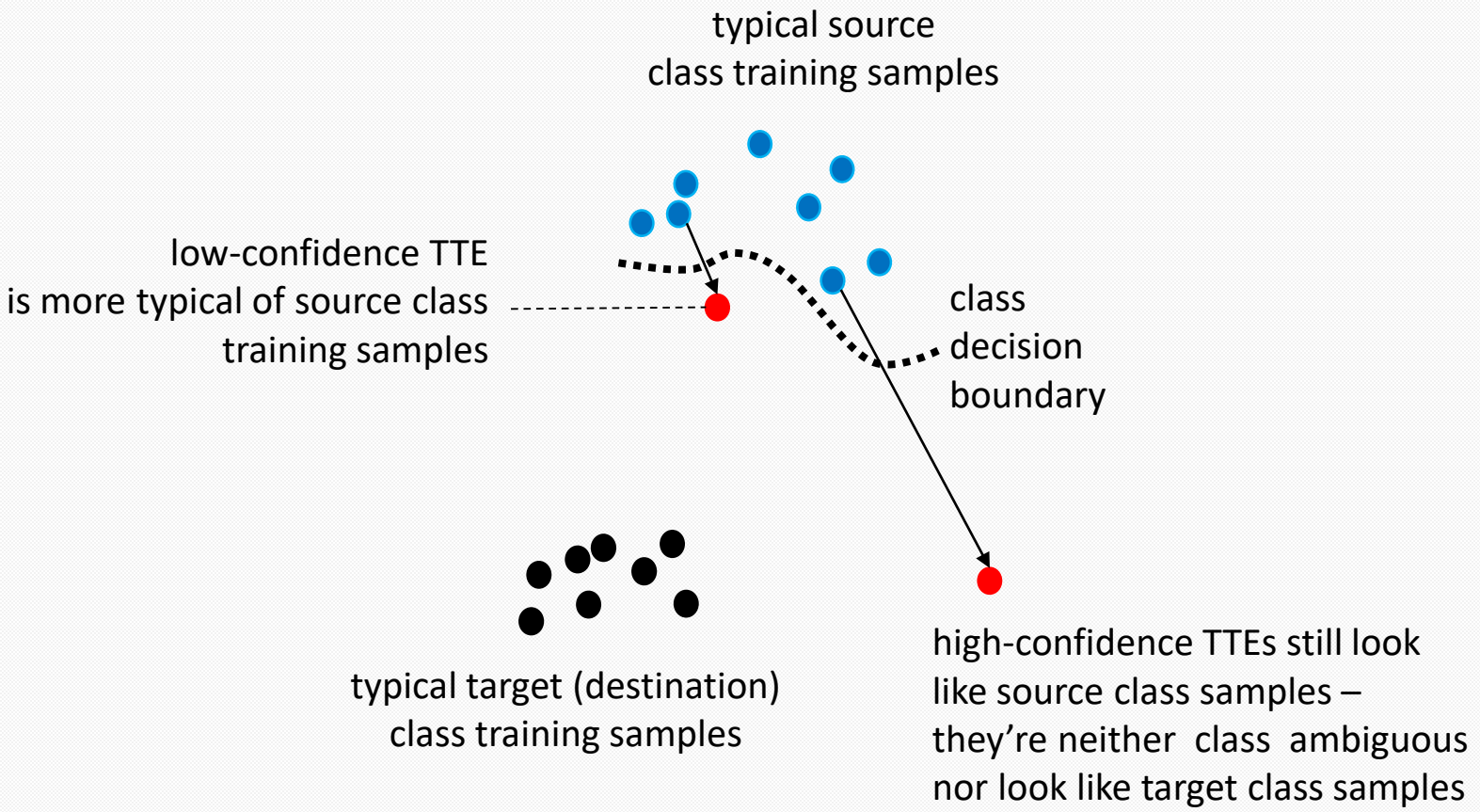
- Recall three types of test-time (operational, online) attacks:
  - Probing (for reverse engineering)
  - Triggers of planted backdoors (Trojans)
  - Test Time Evasion Attacks (TTEs)

- The focus of this chapter is TTEs.

Overview of TTEs

- TTEs are
  - classified to the target class t, but
  - “look like” undoctored source-class s≠t samples, i.e.,
  - they’re hard to detect as anomalous source-class samples.
- **Physical** attacks, e.g., camouflage, have domain-specific constraints.
- **Digital** modifications of observations can achieve a TTE without the severe constraints of physical attacks:
  - Blending a backdoor pattern with the background image.
  - Malware may engage in poly/metamorphism to evade detection by antivirus or Behavioral Anomaly Detection (BAD) systems.
  - Subtle market manipulation may cause dramatic changes in how an AI evaluates a financial option.

High & Low Confidence TTEs



Why TTEs Exist

- Even when the training dataset is very large, training examples may not be proximal to large portions of input space $\mathbb{R}^N$ of a DNN, including regions close to the class-decision boundaries.
- So, it's not surprising that (obviously misclassified) TTEs exist even for "well trained" DNNs.
- This motivates TTE defenses and explainable AI (XAI) research.
- Note that other types of classifiers with simpler and directly engineered decision boundaries, e.g., SVMs, may also be vulnerable to TTEs.

Overview of TTE Attacks on Images

- TTE samples can be constructed in different ways,
  - e.g., JSMA, DeepFool, FGSM, iFGSM/BIM, PGD, CW, ZOO
  - with targeted and untargeted variants
  - as "classical" NN inversion using discriminative learning objectives
- Attack construction typically assumes
  - access to the targeted AI (DNN classifier), or to some reasonably accurate surrogate, i.e., "grey-box" scenario, and
  - clean, correctly classified samples are available.
- For example, the s→t targeted CW attack objective starting from v is

$$\min_{\underline{z}} \left[\|\underline{z} - \underline{v}\|_q + a \max\{ \max_{j \neq t} p(j|\underline{z}) - p(t|\underline{z}) , -\kappa \} + a' D(\underline{z}) \right]$$

  s.t. $c(\underline{v})=s$, parameters $q,a,\kappa > 0$, where
- $D>0$ iff a detection is made, with parameter $a' > 0$ for the case of a white-box (scenario), otherwise $a'=0$ (grey-box scenario).
- Aim to classify TTE z to t with margin/confidence $\leq \kappa$.

Overview of TTE Attacks on Images

- These attacks produce artifacts in greyscale MNIST images of hand-written digits, e.g.,
  - salt & pepper noise (JSMA, ZOO) or
  - grey ghosting (FGSM, CW).

- FGSM and CW TTEs are more evasive for color images, e.g., CIFAR or ImageNet datasets.

- Note that, generally, a grey-box adversary could
  - perturb the activations of an **internal layer** of the DNN,
  - or some other representation of the input sample, and
  - then solve an inverse problem to reconstruct an adversarial example that can be applied as input to the targeted DNN.
- Working on internal layers allows generalization of attacks and defenses to other domains.

JSMA on MNIST images - a weak digital TTE

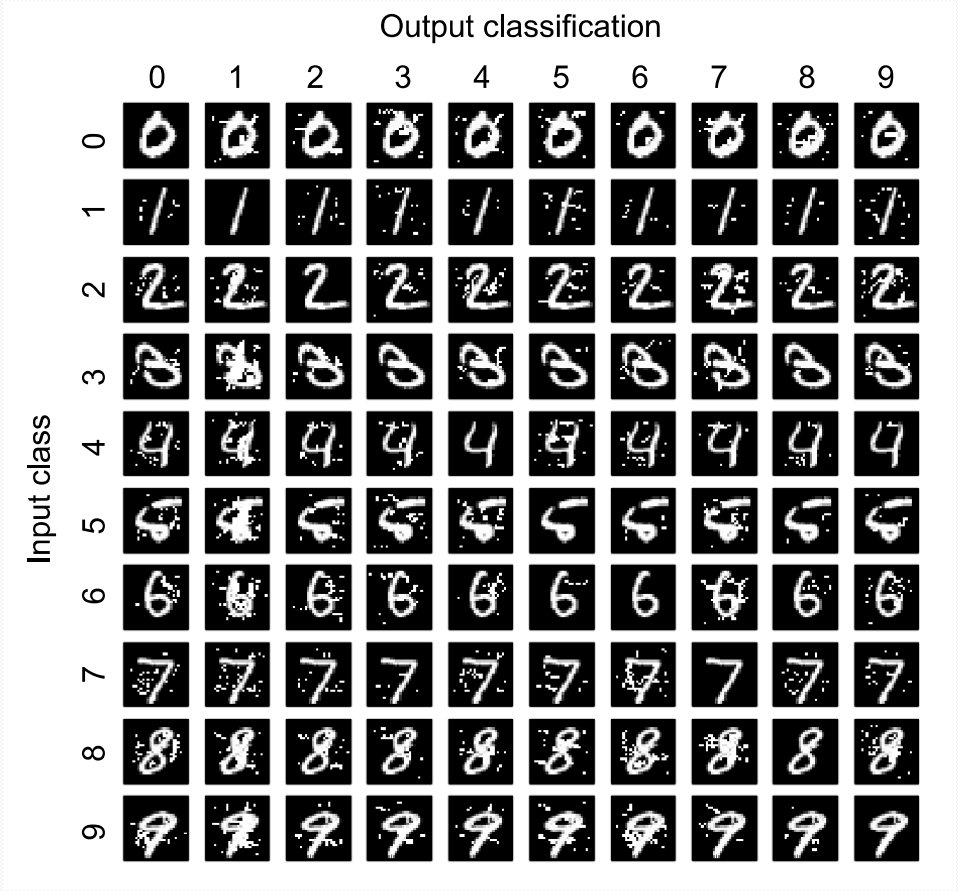


Fig. 1: *Adversarial sample generation* - Distortion is added to input samples to force the DNN to output adversary-selected classification (min distortion $= 0.26\%$, max distortion $= 13.78\%$, and average distortion $\varepsilon = 4.06\%$).

- Note the clearly visible “salt and pepper” noise and extra white pixels in JSMA attack examples.
- Simple detection by counting contiguous white regions gives 0.97 ROC AUC.

Overview of TTE Defenses & Need for AD

- Some published defenses: make black-box or only partial grey-box assumptions on the attacker;
- seek to make deep learning robust to TTEs, so they are supervised or bias the classifier;
- add noise or distortion to the input sample to destroy adversarial perturbation but hopefully preserve accuracy; or
- employ classification ensembles.
- Previous defenses may not work well for high-confidence TTEs and rely too much on security through obscurity.
- More promising **test-time Anomaly Detection (AD)** augments the classification capability, identifying whether a given input is anomalous w.r.t. the training set.

Certified Learning Defense – An Example Theorem

- Let f be the vector of rectified logit outputs.
- Assume f is Λ-Lipschitz continuous as:

$$\forall x,y, \quad \| f(x)-f(y) \|_\infty \leq \Lambda \| x-y \|_2$$

- Let $m(x) = f_{c(x)}(x) - \max_{k \neq c(x)} f(\underline{x})$ be the classification margin of x, with ground truth class label c(x).
- So, x is correctly classified if and only if $m(x)>0$.
- Let $B_2(x,r)$ be the l_2 ball with center x and radius r.
- **Theorem:** If $m(x)>0$ then $B_2(x, m(x)/(2\Lambda))$ is class-pure.
- But may not be a useful result in practice because a DNN's Λ is hard to accurately estimate.

Certified Learning Defense by Maximum Margin Training

- Consider the training objective with parameters $\lambda>0$:

$$\min_{\theta} \sum_{x \in V} \lambda_x \left(\max_{k \neq c(x)} f(\underline{x}) + \mu - f_{c(x)}(x) \right)$$

- If any summand term x is positive, increase λ_x & retrain.
- This results in classification margin $\geq \mu > 0$ for all training samples x.
- But this may exacerbate overfitting, causing the classifier to be biased to the training set and leading to poor generalization performance on "clean" test examples (distributed as training examples).
- Similar issue with other types of "certified" defenses, e.g., blurring test samples with additive, isotropic Gaussian noise to "smoothen" the class decisions [Cohen et al. '19, Roth et al. '19].
- Note: The training dataset distribution is *not* the same as the union of "balls" around the training samples.

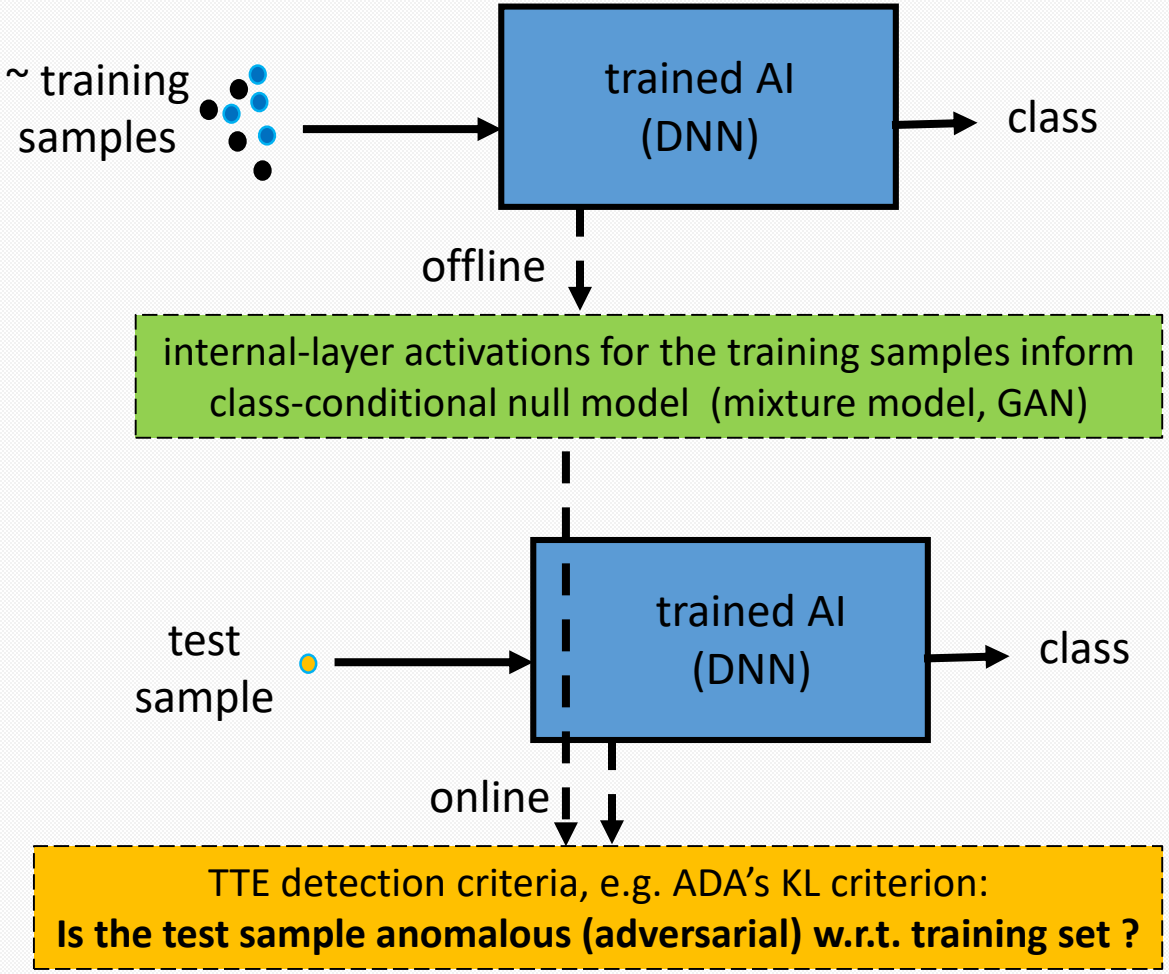
Remarks on "Adversarially" Robust Training

- One general idea is to augment the training dataset with TTEs constructed from training samples using known attack techniques (and supervised with the training samples' labels)
- But the TTEs used are only small perturbations to avoid inducing bias in the learned classifier (i.e., only low-confidence TTEs).
- Also, this supervised approach may not protect against new TTEs (zero-days), even low-confidence ones, and
- may not protect against existing TTEs if constructed from clean samples not used for training.

Detection vs Robustness

- **Operationally**, may be more important to **detect** an attack than to “correctly classify” in the face of it:
- Should a perturbed tumor image be diagnosed ?
- Does it matter which category a program falls into if it contains malware ?
- But for the case of physical (e.g. camouflage) attacks, it may be important to do both detection and corrective classification.

TTE Detection-Based Defense



- Can use detected anomalous test/operational samples for active or reinforcement learning.
- As our approach is based on internal layer activations, it also applies to other (e.g., discrete) data domains.

ADA detector – set-up

- Consider a clean, correctly classified class-s sample $\underline{v}$, i.e., $f(\underline{v})=s$, that is adversarially perturbed to $\underline{x}= \underline{v}+\underline{\delta}$ **just until** it becomes a class-t sample, i.e., $f(\underline{v}+\underline{\delta})=t$, i.e., a low confidence TTE attack.

- The defender is presented with a possibly adversarial input $\underline{x}$ with inferred class $f(\underline{x})=t$.

- Suppose class conditional nulls $g_{k|i}$, $i \in C$, are learned for the activations of one or more internal layers k, as elicited by the training set examples.

Baseline ADA detector - details

For test sample $\underline{x}$ with $f(\underline{x})=t$:

- $\underline{w} := f^{(k)}(\underline{x})$ the vector of k^{th} internal layer activations of the DNN.
- $P(k) := \{ r_0\, g_{k|t}(\underline{w}),\ r_0\, g_{k|s'}(\underline{w}) \}$

  where $s' = \text{argmax}_{i \neq t}\, g_{k|i}(\underline{w})$, the "most likely" class for $\underline{x}$ except t.
- $Q(k) := \{ q_0\, p(t|\underline{x}),\ q_0\, p(s'|\underline{x}) \}$ (DNN output layer, s' dep. on k)
- r_0, q_0 are normalizations chosen so that P(k) and Q(k) are PMFs.
- Declare $\underline{x}$ a TTE attack instance if

  $$D_{KL}(P(k) \| Q(k)) > \text{threshold},$$

  with target class t and source class s';
- else accept the class decision $f(\underline{x})=t$ for $\underline{x}$.
- Set the threshold in an unsupervised way to limit false positives.

ADA detection

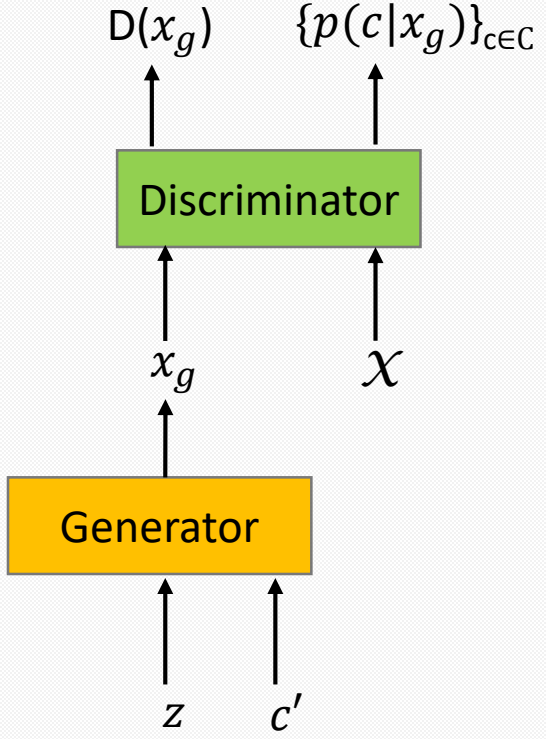
- Variations of ADA include those which, e.g.,
  - consider different layers simultaneously (max-ADA),
  - employ different null models, and
  - account for class confusion.

- Note, for example, that [Feinman et al. ‘17]
  - uses a Gaussian kernel to model penultimate layer activations
  - for *supervised* detection (leveraging known TTE methods).

- For another example, though the “decision smoothing” methods of [Cohen et al. ’19, Roth et al. ‘19] are unsupervised and the latter accounts for class confusion, they use the penultimate layer.

Detection based defenses using GANs - background

- A class-conditional GAN consists of a generator G followed by a discriminator D.
- G's input is "white noise" $\underline{z}$ and a class label c.
- Once "well trained", $x_g = G(\underline{z}, c') \sim \mathcal{X}_{c'}$ (is distributed as nominal class-c' data $\mathcal{X}_{c'}$) and
- $D(\underline{x}_g)$ indicates whether $\underline{x}_g$ is "typical" of the training dataset $\mathcal{X}$ (or "real").
- Optimal D minimizes Jensen-Shannon divergence between distributions of V and $G(\underline{z}, c)$.
- AC-GAN: D also outputs class posteriors,



GANs training configuration

AC-GAN – training objective functions

- AC-GAN objective functions

$$L_s = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_w} [\log (1 - D(G(z, c')))]$$

$$L_c = \mathbb{E}_{x \sim p_{\text{data}}} [\log p(c^*(x)|x)] + \mathbb{E}_{z \sim p_w} [\log p(c'|G(z, c'))]$$

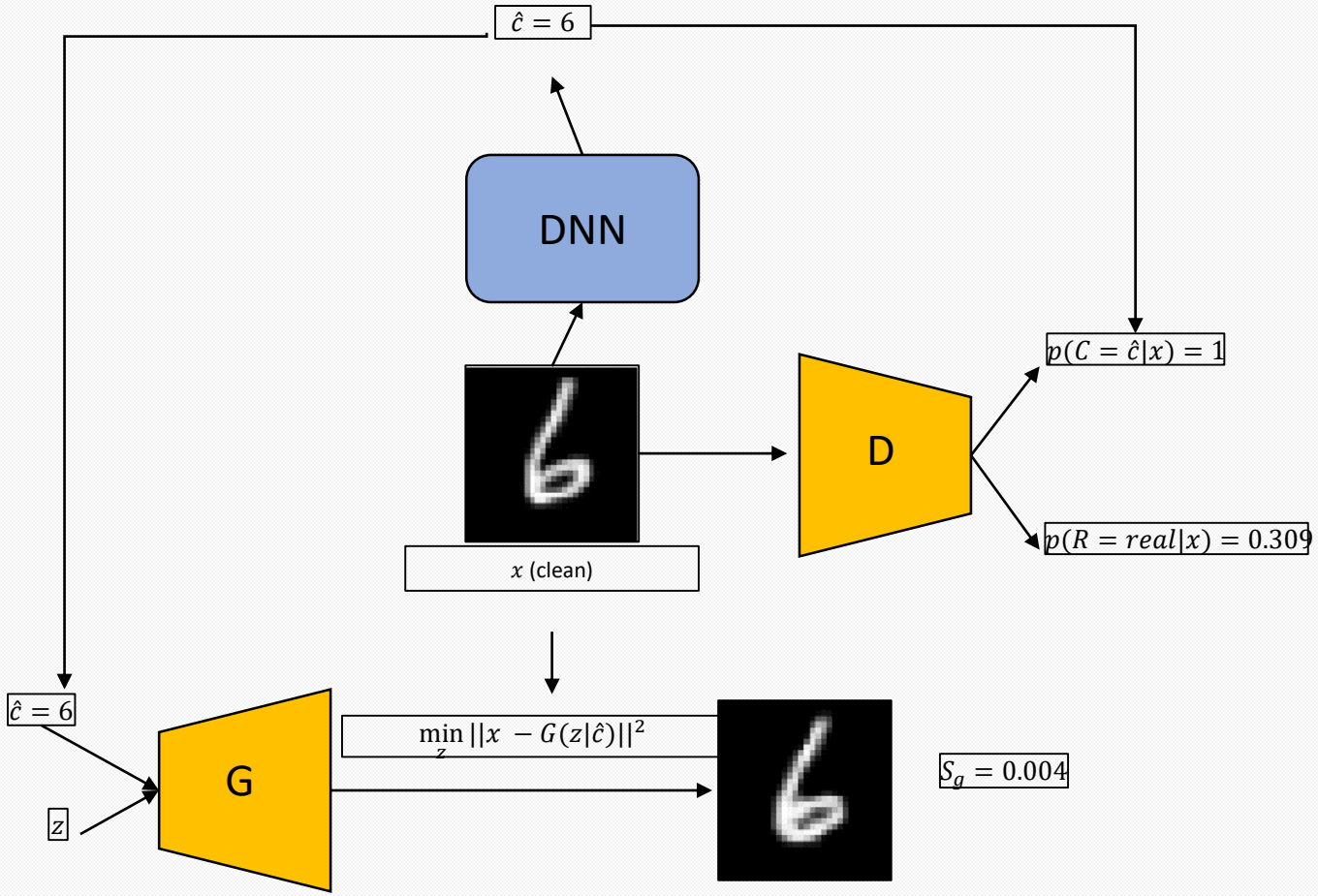
- where here $c^*(x)$ is the ground-truth class label of x, and c' is chosen uniformly at random from Y.
- The Discriminator is trained to maximize $L_s + L_c$ while the Generator is trained to minimize $L_s - L_c$
- Considering the last term of L_c, D and G seem to *cooperate* to train D's class posteriors p.

AC-GAN TTE-detection statistics [WXMK'21]

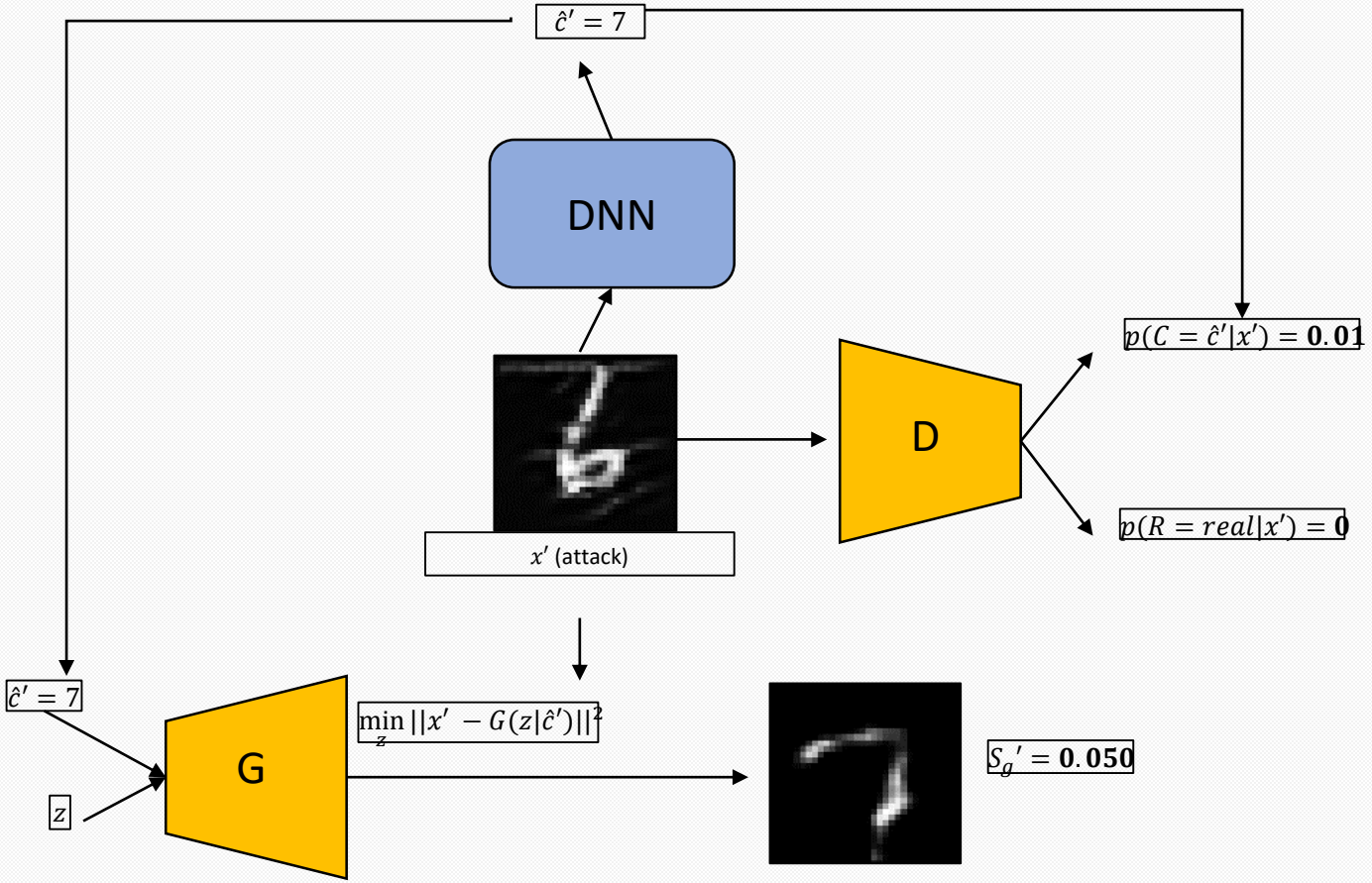
- Test sample x (or internal-layer activation) with decided-upon class $\hat{c}(x)$ by the defended DNN.
- $S_R = D(x)$… probability x is real (i.e., x~X or x~p_{data})
- $S_C = p(\hat{c}(x)|x)$ …class posterior of AC-GAN's Discriminator
- $S_D = \log S_R + \log S_C$
- $S_G = \min_Z \| x - G(z, \hat{c}(x)) \|^2$ … reconstruction error/loss

- Can also use all the statistics $S = [S_R, S_C, S_G]$ (All-AD), with anomalies then detected, e.g., based on a p-value assessed w.r.t. a null e.g. GMM for S.

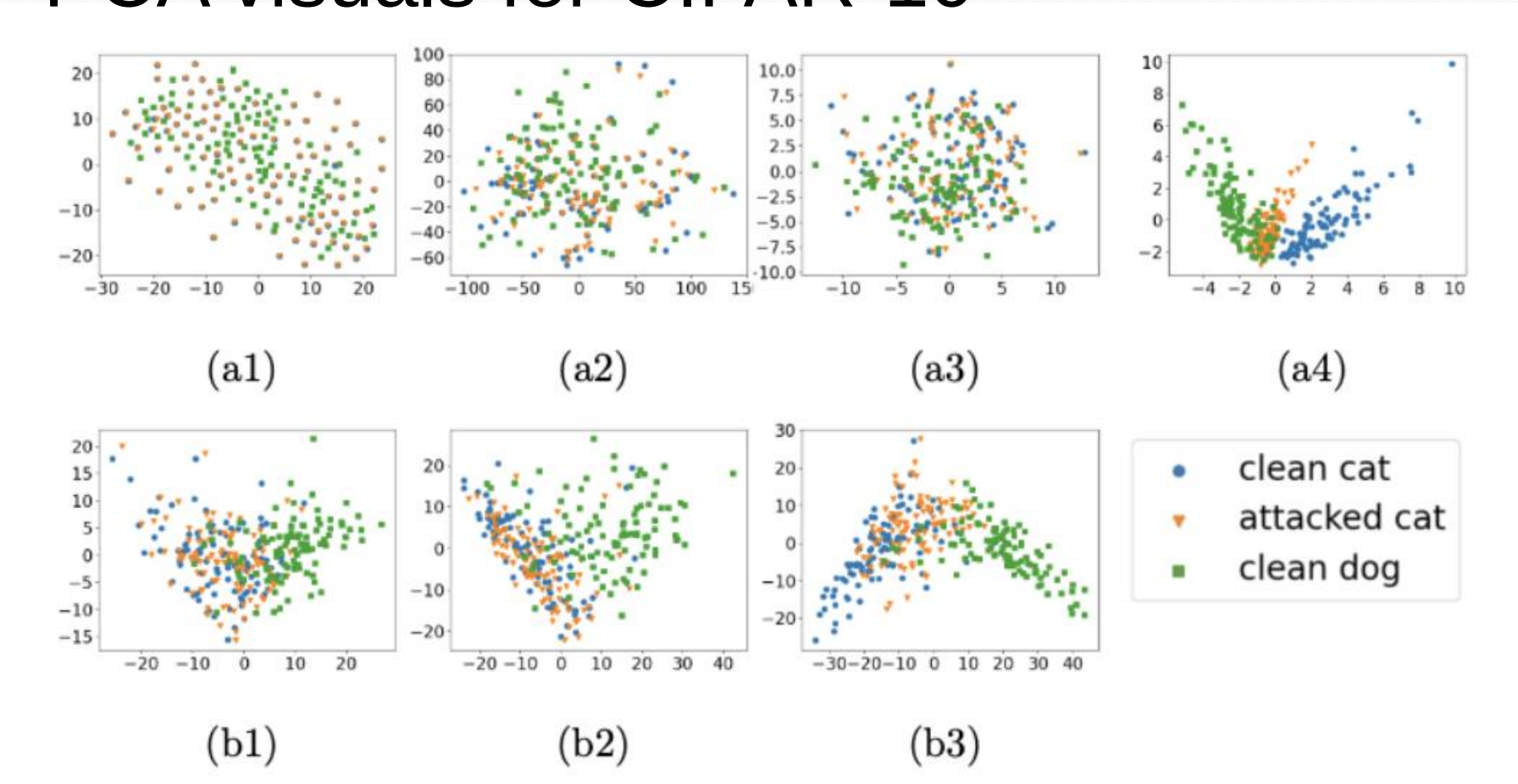
AC-GAN illustrative example: clean input x



AC-GAN illustrative example: adversarial input x’



PCA visuals for CIFAR-10



- n: 1 = input; 2 = 1st conv layer; 3 = 11th conv layer; 4 = penultimate layer
- (an): ResNet DNN activations of layer n
- (bn): penult. layer of discriminator when ACGAN trained on ResNet layer n

Grey-Box Defense Experiments

- Datasets: MNIST, CIFAR-10
- DNN: ResNet-18
- Attack methods (all targeted):
  - High-confidence CW
  - Low-confidence CW
  - FGSM
- Measurement: pAUC-0.2 = partial area under the ROC curve for False positive rate (FPR) below 0.2

Performance Results for Grey-Box Attacks

	MINST			CIFAR-10		
	CW-HC	CW-LC	FGSM	CW-HC	CW-LC	FGSM
f-AnoGAN [32]	0.0981	0.0995	0.0887	0.0576	0.0563	0.0566
KD [24]	0.1892	0.1887	0.1880	0.0533	0.0584	0.1642
MD [16]	0.1861	0.1832	0.1901	0.1042	0.1125	0.1783
ODDS [41]	0.0618	0.0412	0.0537	0.0910	0.0568	0.0436
SID [21]	0.1576	0.1628	0.1726	0.1489	0.1412	0.1388
ADA [17]	0.1715	0.1732	0.1823	0.1593	0.1601	0.1782
G-AD	0.1525	0.1517	0.1612	0.0181	0.0254	0.0203
D-AD	0.1915	**0.1970**	0.1862	**0.1881**	**0.1899**	**0.1819**
All-AD	**0.1923**	0.1964	**0.1905**	0.1798	0.1825	0.1618
D-AD-L1	0.1897	0.1873	0.1824	0.1805	0.1787	0.1768

Table 3: pAUC-0.2 results of different detection methods under different attacks.

Grey-Box Attacks (cont)

- Defenses evaluated considering only completely successful adversarial examples on correctly classified, clean (non-adversarial) examples.
- AnoGAN and F-AnoGAN only use reconstruction loss statistics, which are not significantly affected by TTEs & which also require a lot of computation at test-time.
- Several other ways to employ auxiliary GANs to achieve robust classification (including rejection of TTEs).
- GANs is complex to train for many application domains, but this complexity is not borne at test time.
- Can also extend simpler ADA approach by
  - using p-values of the decided-upon class null to detect high-confidence TTEs, and
  - using more sophisticated nulls (with BIC model-order control).

Label-corrective methods

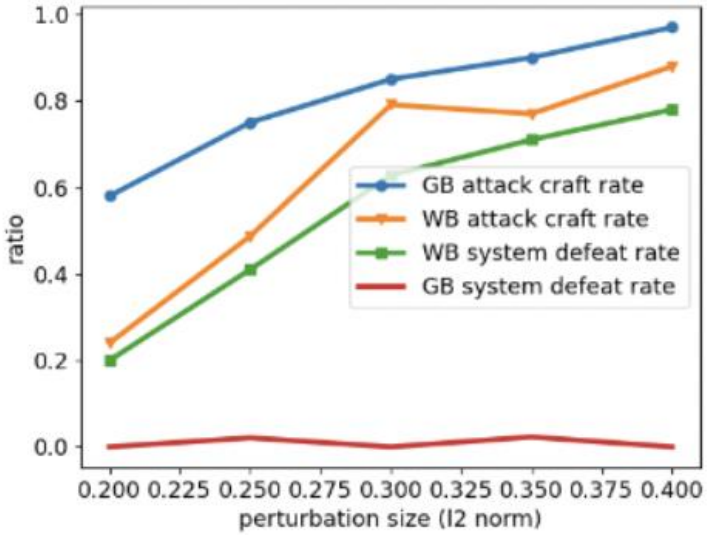
- Defense-GAN has the DNN classify $G(z^*)$ instead of x where
  $z^* = \text{argmin}_z \| G(z) - x \|^2$ is hard to compute.
- Our pix2pix just uses G(x) instead $G(z^*)$, i.e., use the Generator as an auto-encoder (trained on noisy images).
- Alternatively, we can take the AC-GAN's Discriminator class decision, recalling that AC-GAN can be trained based on
  - the input layer (D-AD) or
  - the first convolutional layer activations (D-AD-L1).
- Note that ADA could also be used for label correction.

Example Label-Corrective Measures

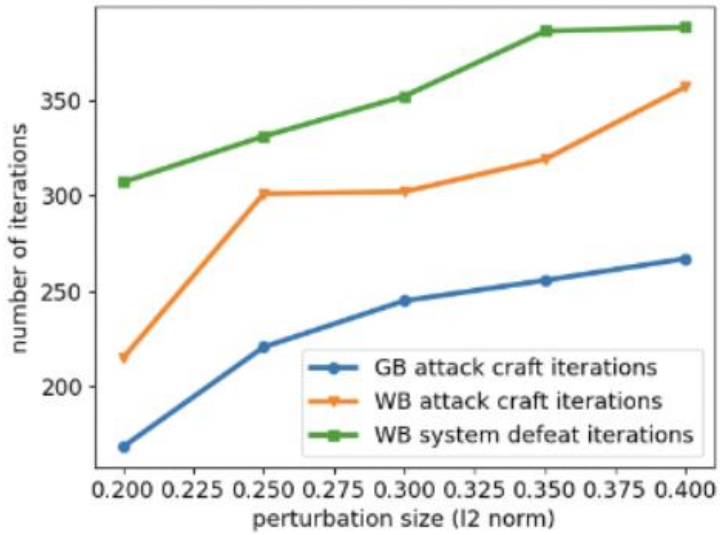
		MINST		CIFAR-10	
	Include $\hat{c}$	CW-HC	CW-LC	CW-HC	CW-LC
D-GAN[56]	✓	0.9247	0.9273	0.2089	0.2079
		0.9314	0.9291	0.3274	0.3782
D-AD	✓	0.8951	0.8816	0.8266	0.8191
		0.8964	0.8919	0.8423	0.8316
D-AD-L1	✓	0.8295	0.8204	0.7643	0.7420
		0.8421	0.8395	0.7874	0.7535
pix2pix	✓	**0.9504**	**0.9668**	**0.8539**	**0.8566**
		0.9612	**0.9671**	**0.8632**	**0.8788**

Table 3: Classification accuracy in correcting the DNN decision, for correctly detected adversarial examples. CW-HC means CW high confidence attack; CW-LC means CW low confidence attack. D-GAN means Defense-GAN method.

Increased Work-Factors of White-Box Attacks



(a)



(b)

Figure 4: (a): Several attack success measurements under white-box (WB) and grey-box (GB) attack scenarios. (b): Average iterations required for successful attacks. The experiments are conducted on the CIFAR-10 dataset and Projected Gradient Descent (PGD) is used to control the perturbation size.

Responses Post Detection

- Recall automated label-correction may be dangerous as test samples may be fundamentally class ambiguous,
- but label-correction may be very useful in some cases.
- An ensemble of detectors can be used.
- For a detected TTE, best response may be “suspicious” or “don’t know” together with context for security administration.
- Context may be supplemented by additional forensics identifying input features responsible for class decision, e.g., grad-CAM (Chapter 2).

Online Vigilance and Robust Adaptation by Out of Distribution Detection (OODD)

- During test/operation time between retraining epochs:
- OODD can be based on deep generative models (reconstruction loss) or traditional null models (p-values) of embedded features.
- OODD to detect adversarial inputs (TTEs) or RE probes, see Chapter 14, optionally in a class conditional fashion.
- OODD as part of more frequent, active-learning based model refinement, e.g., to address biased training set or model drift.
- OODD on (sample, inference) data for regression or (state, action, next-state) data for policy optimization, see chapter 12.
- Detection of backdoor triggers using (offline) backdoor detection, see chapter 10.

With Permission, Figures Reproduced From

- H. Wang, D.J. Miller, and G. Kesidis. Anomaly Detection of Test-Time Evasion Attacks using Class-Conditional Generative Adversarial Networks. *Computers and Security* **124**, Jan. 2023.