CENG7880
Trustworthy and Responsible AI

Instructor: Sinan Kalkan

(https://ceng.metu.edu.tr/~skalkan)

For course logistics and materials:
https://metu-trai.github.io

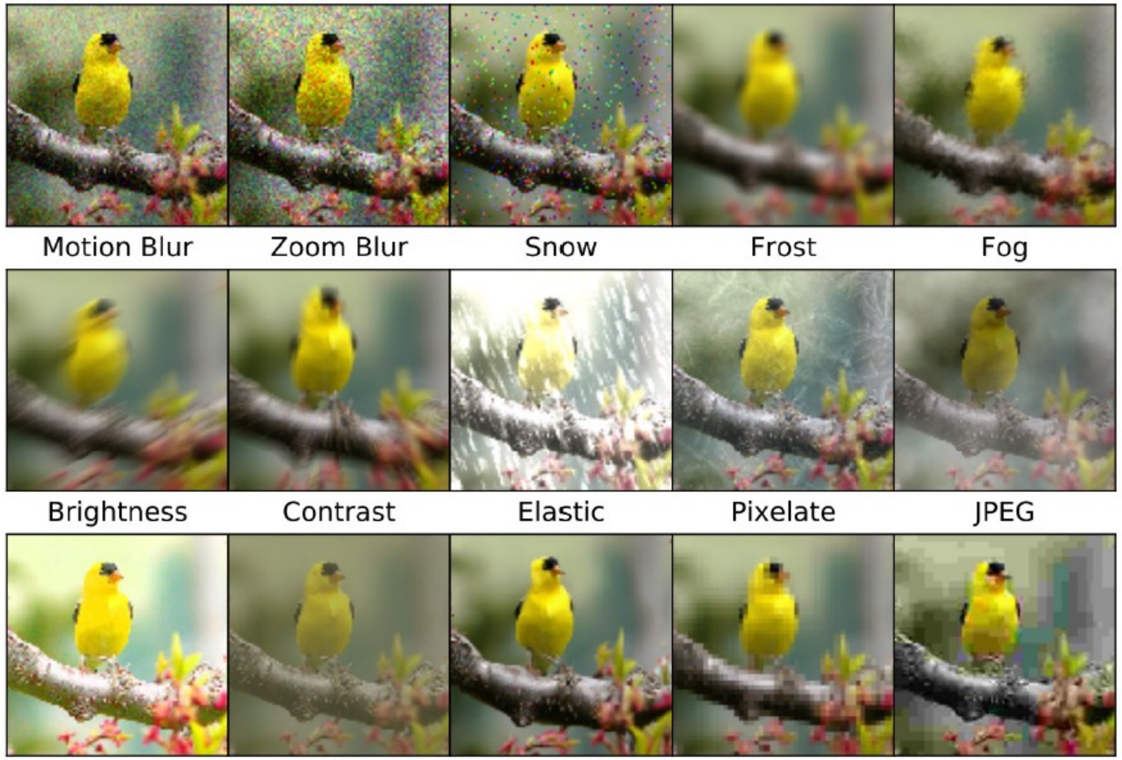
MIDDLE EAST TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Robustness: Agenda

Previously on CENG7880

Robustness to Distribution Shifts

- Images
  - Noise, color changes, light changes, day/night, summer/winter, …
- Audio
  - Noise, background, gain level, …
- Text
  - Noise, synonyms, alternative phrasing, ..



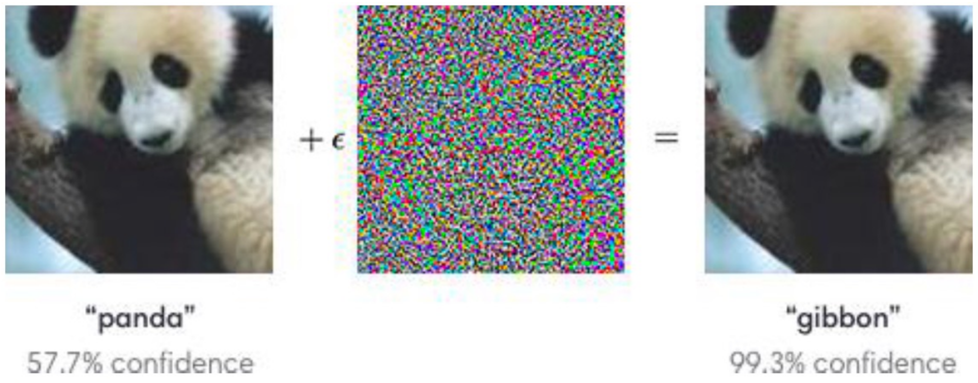
Hendrycks & Dietterich, Benchmarking Neural Network Robustness to Common Corruptions and Perturbations

Previously on CENG7880

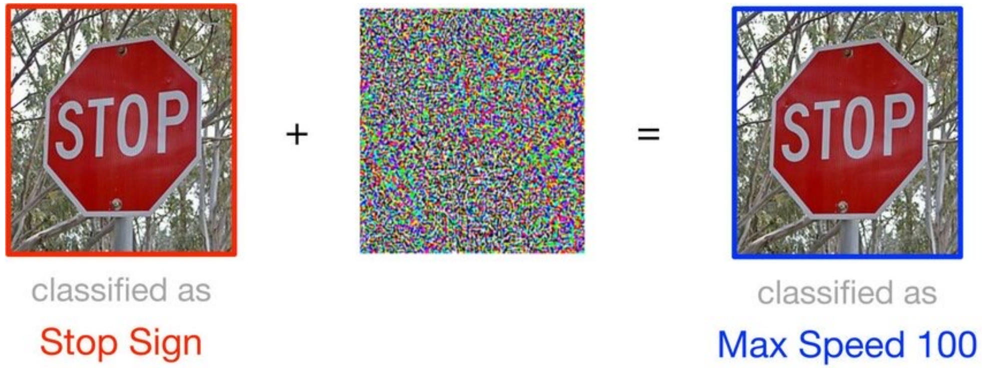
Robustness: Agenda

Adversarial Robustness

- Types of adversarial attacks
- Adversarial training



Szegedy et al., Intriguing Properties of Neural Networks, 2014



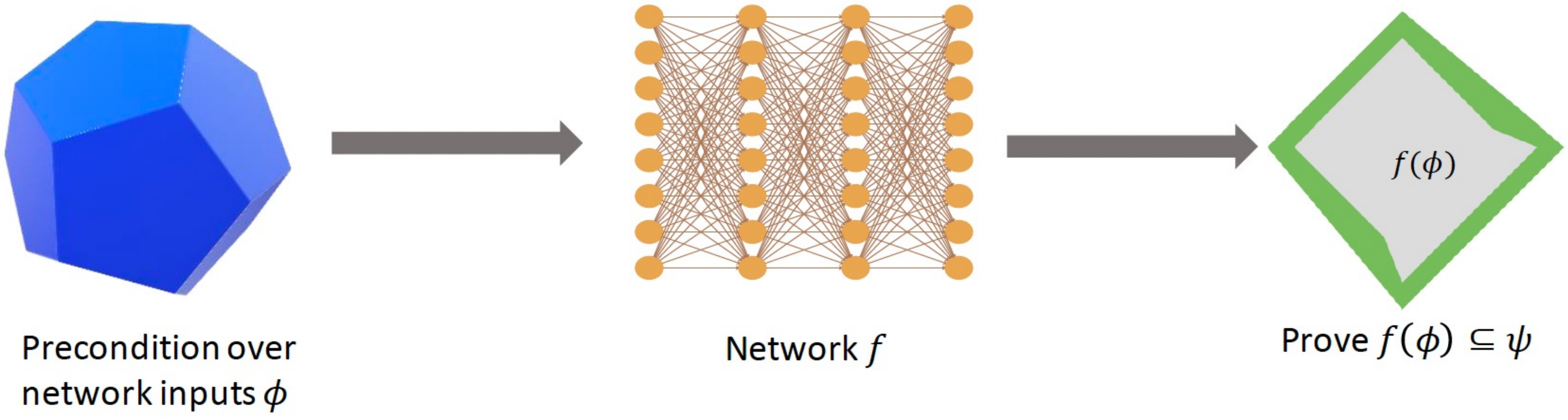
https://www.altacognita.com/robust-attribution/

Robustness: Agenda

Certifying/Verifying Robustness

Previously on CENG7880

Neural network certification: problem statement



Figs: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Robustness: Agenda

Previously on CENG7880

Calibrated Predictions & Uncertainty

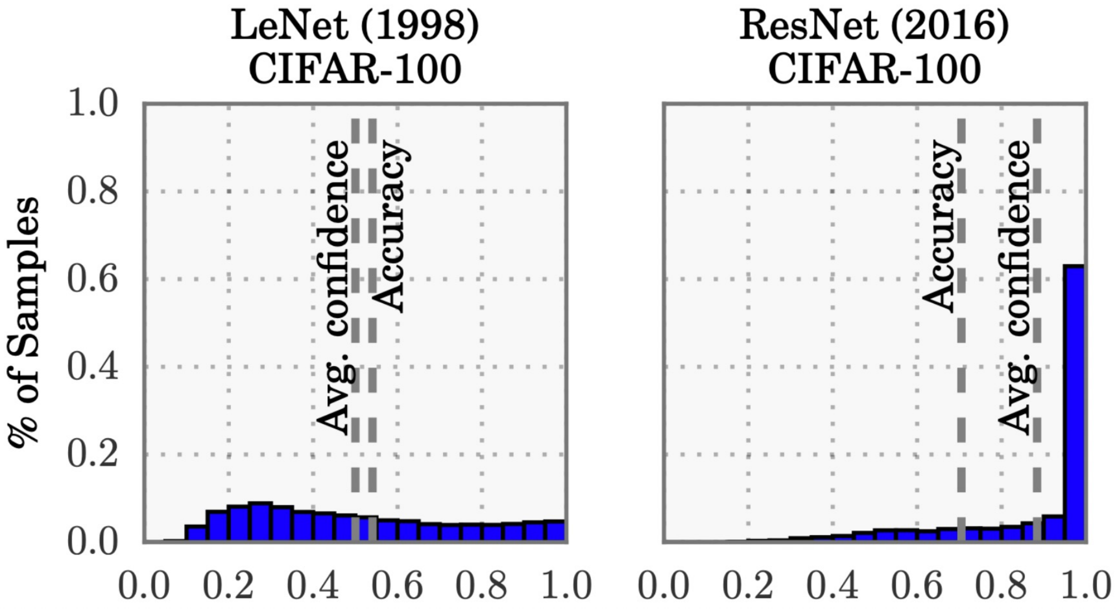


Fig: Uni of Pennsylvania, Trustworthy ML - CIS 7000

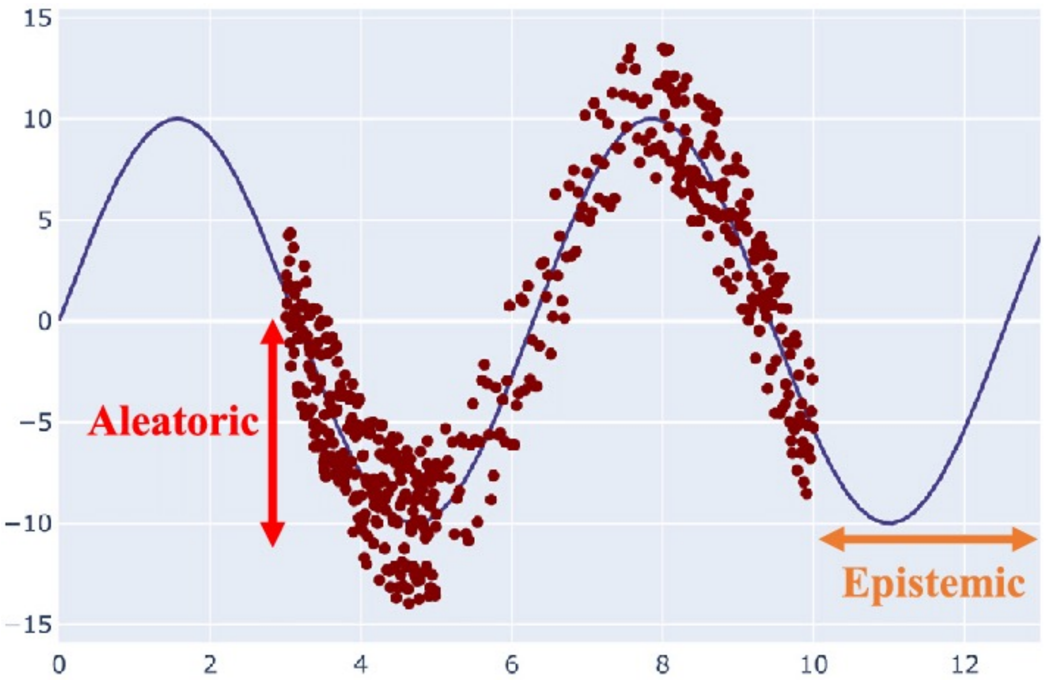


Fig: Abdar et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, 2021

Example: Real Perturbations

Previously on CENG7880

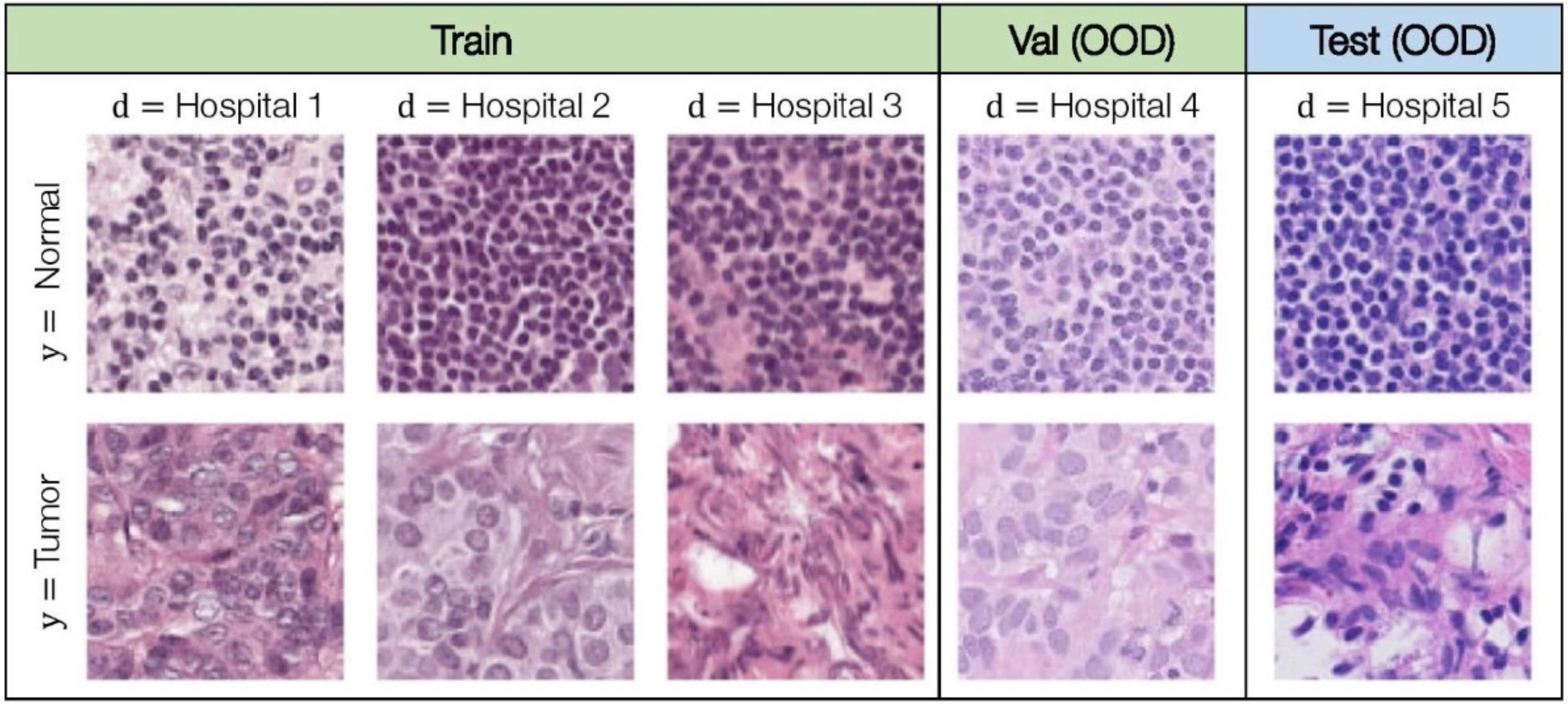
	Train			Test	
Satellite Image (x)					
Year / Region (d)	2002 / Americas	2009 / Africa	2012 / Europe	2016 / Americas	2017 / Africa
Building / Land Type (y)	shopping mall	multi-unit residential	road bridge	recreational facility	educational institution

Koh et al., WILDS: A Benchmark of in-the-Wild Distribution Shifts, 2020

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Previously on CENG7880

Example: Real Perturbations



Koh et al., WILDS: A Benchmark of in-the-Wild Distribution Shifts, 2020

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Distribution Shift

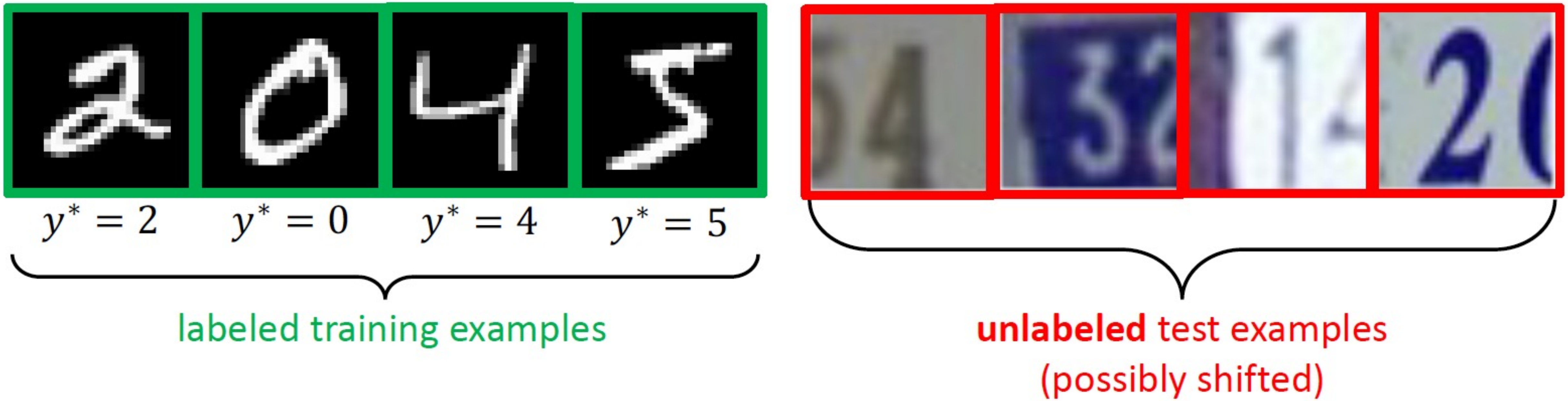
- **Distribution shift:** Training and test distributions differ
  - Training set consists of samples $(x_1, y_1), \ldots, (x_n, y_n) \sim P$
  - Test set consists of samples $(x'_1, y'_1), \ldots, (x'_m, y'_m) \sim Q$

- **Supervised learning under distribution shift**
  - Given training dataset $Z \subseteq \mathcal{X} \times \mathcal{Y}$ consisting of i.i.d. samples $(x, y) \sim P$
  - Goal is to minimize loss $\mathbb{E}_Q[\ell(\theta; x, y)] \not\approx |Z|^{-1} \sum_{(x,y) \in Z} \ell(\theta; x, y)$
  - Computing $\hat{\theta} = \min_{\theta} |Z|^{-1} \sum_{(x,y) \in Z} \ell(\theta; x, y)$ may not work

Unsupervised Domain Adaptation

Previously on CENG7880

- **Idea:** Use **some** information about the distribution shift
- Consider **unsupervised domain adaptation** setting



Unsupervised Domain Adaptation

Previously on CENG7880

- Data is easy to collect but labeling costs money
  - **Example:** Data from a different hospital

- Collect data during run time
  - **Example:** Self-driving car

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Label Shift Assumption

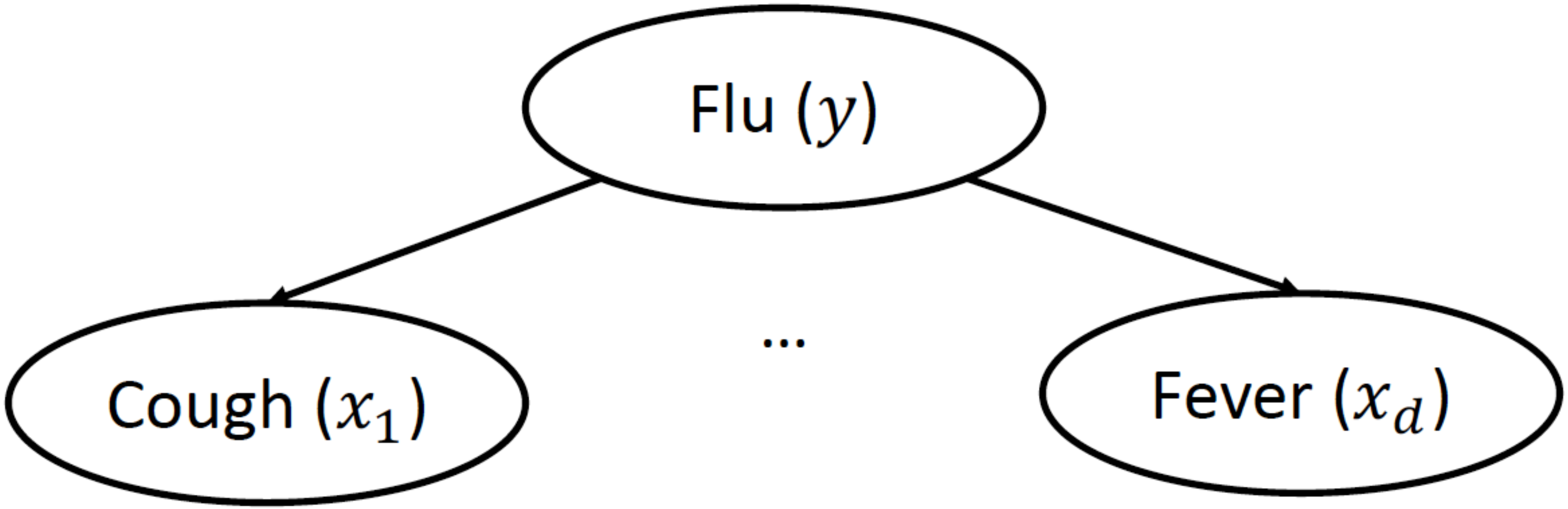
Previously on CENG7880

- Let p and q be the density functions for P and Q, respectively
- **Label Shift Assumption:** $p(x \mid y) = q(x \mid y)$
  - But may have $p(y) \neq q(y)$
  - **Intuition:** The rates of labels changes, but the kinds of labels don't

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Label Shift Assumption

- **Example:** Increase in flu cases due to an outbreak
  - x are the symptoms, y is indicator for flu
  - $P(x \mid y)$ is rate of symptoms conditioned on having disease (stays the same)
  - $P(y)$ is rate of flu (can change if there is an outbreak)



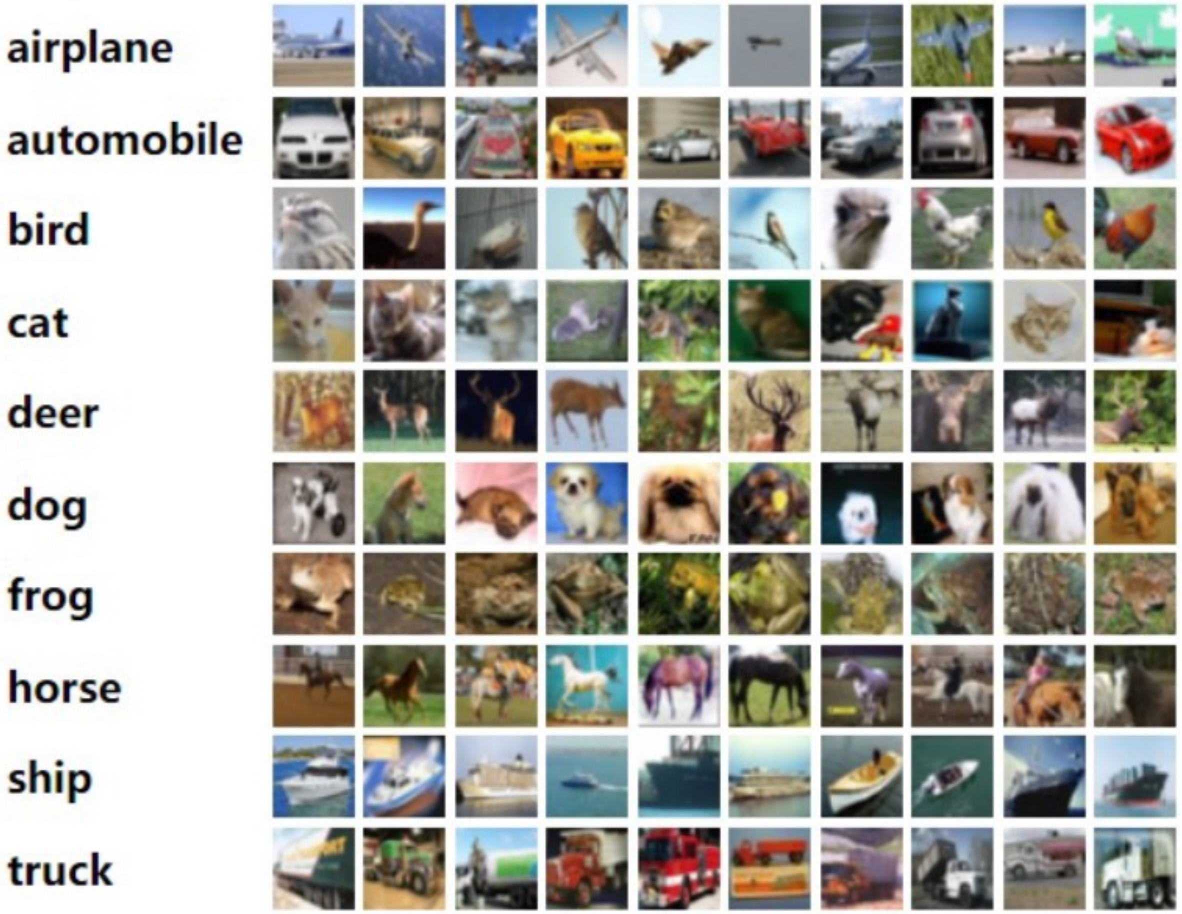
Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Label Shift Assumption

Previously on CENG7880

- **Example:** Changes in label distribution
  - x is an image, y is the label
  - $P(x \mid y)$ is the distribution of images of a given label
  - $P(y)$ is rate of that label

- Often, the training labels are balanced, which is a source of label shift



Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

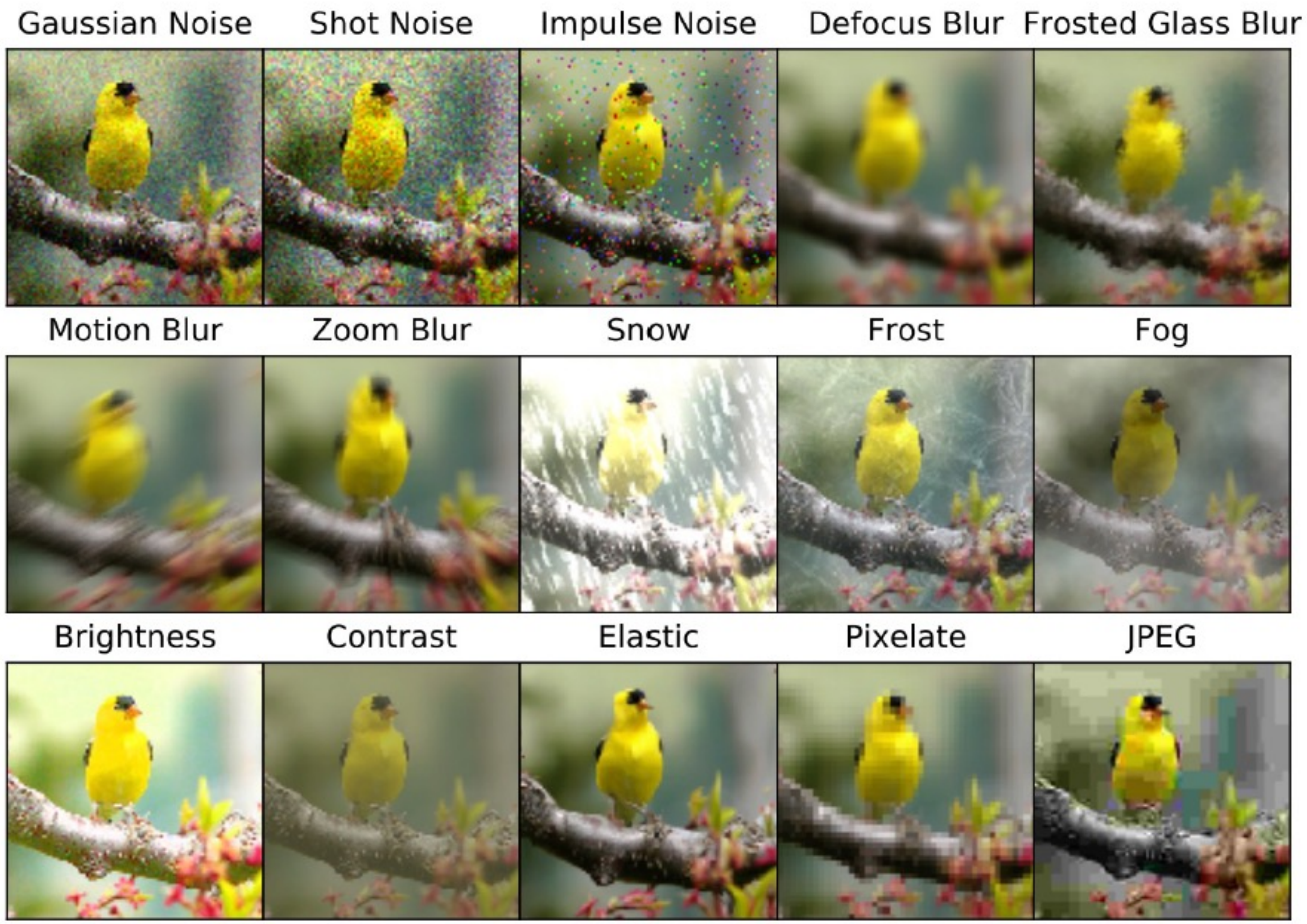
Covariate Shift Assumption

- Let p and q be the density functions for P and Q, respectively
- **Covariate Shift Assumption:** $p(y \mid x) = q(y \mid x)$
  - But may have $p(x) \neq q(x)$
  - **Intuition:** The label computation does not change, but the inputs can change

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Covariate Shift Assumption

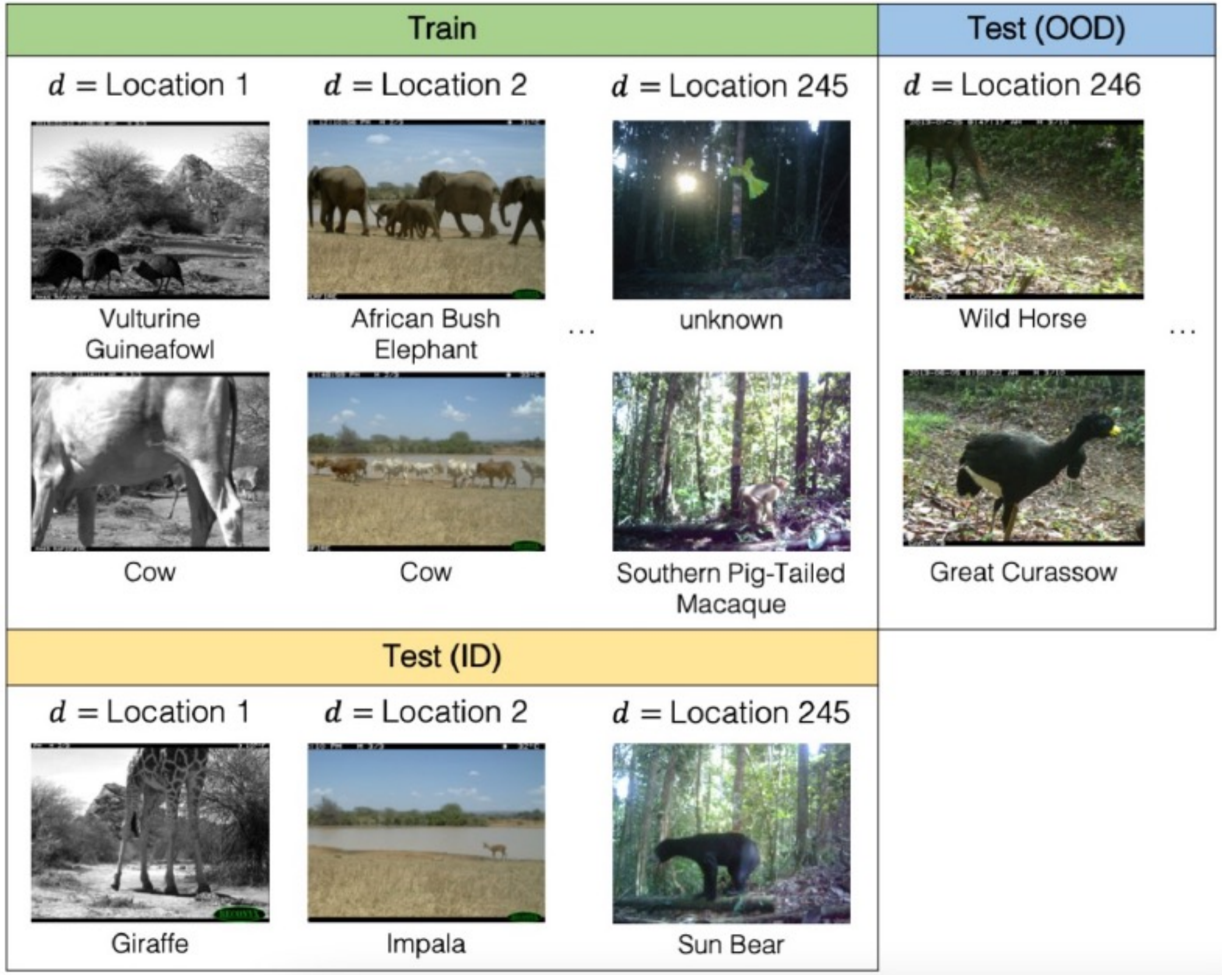
Previously on CENG7880



Hendrycks & Dietterich, Benchmarking Neural Network Robustness to Common Corruptions and Perturbations

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Covariate Shift Assumption



Koh et al., WILDS: A Benchmark of in-the-Wild Distribution Shifts, 2020

Covariate Shift Assumption

- **Computer vision**
  - Daytime vs. nighttime
  - Color shifts, lighting shifts, etc.
  - Driving in a new city
- **Natural language processing**
  - Change in vocabulary frequency over time
  - Regional vocabulary
  - News writing vs. conversational writing
- Covariate shift is pervasive

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weighting

- Given distributions P and Q, the **importance weight (function)** is

$$w(x, y) = \frac{q(x, y)}{p(x, y)}$$

- **Key property (by definition):**

$$\mathbb{E}_Q[\ell(\theta; x, y)] = \mathbb{E}_P[\ell(\theta; x, y) \cdot w(x, y)]$$

- **Key question:** How to compute importance weights?

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Label Shift

- In the label shift setting, we have

$$
\begin{aligned}
w(x,y) &= \frac{q(x,y)}{p(x,y)} \\
&= \frac{q(x|y)q(y)}{p(x|y)p(y)} \\
&= \frac{q(y)}{p(y)} \\
&:= w(y)
\end{aligned}
$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

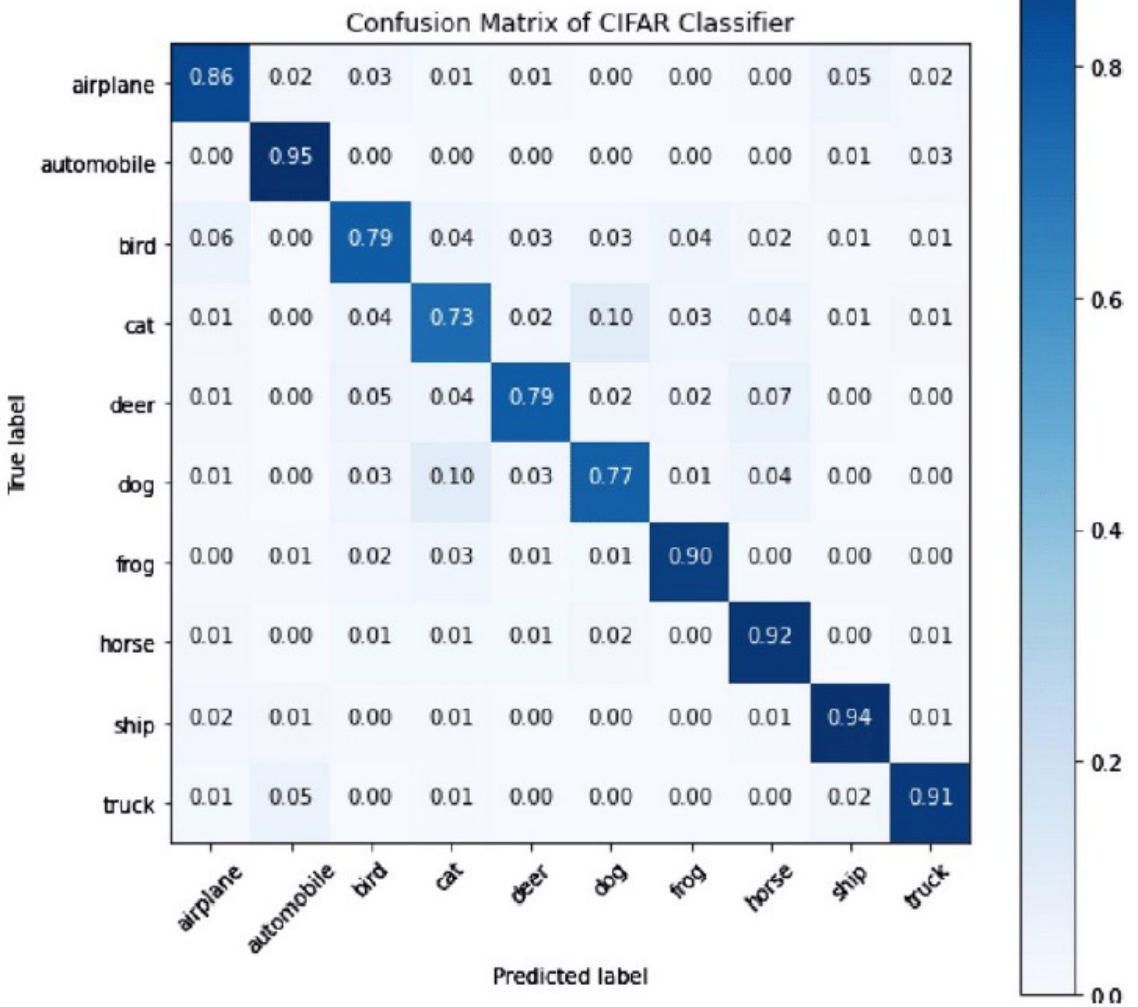
Importance Weights for Label Shift

Previously on CENG7880

- Given a classifier $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{1, \dots, K\}$, consider the confusion matrix $C \in \mathbb{R}^{K \times K}$ defined by

$$C_{ij} = \mathbb{P}_P[f(x) = i, y = j]$$

- Also, define $p, q \in \mathbb{R}^K$ by
  - $p_i = \mathbb{P}_P[f(x) = i]$
  - $q_i = \mathbb{P}_Q[f(x) = i]$



Sooksatra, Evaluation of adversarial attacks sensitivity of classifiers with occluded input data

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Label Shift

$$C_{ij} = \mathbb{P}_P[f(x) = i, y = j] \quad p_i = \mathbb{P}_P[f(x) = i] \quad q_i = \mathbb{P}_Q[f(x) = i]$$

$$\mathbb{P}_P[f(x) = i \mid y = j] = \mathbb{P}_Q[f(x) = i \mid y = j]$$

- Now, we have

$$q_i = \sum_{j=1}^{k} \mathbb{P}_P[f(x) = i, y = j] \cdot w(j)$$

$$\begin{bmatrix} q_1 \\ \vdots \\ q_K \end{bmatrix} = \begin{bmatrix} \mathbb{P}_P[f(x) = 1, y = 1] & \cdots & \mathbb{P}_P[f(x) = 1, y = K] \\ \vdots & \ddots & \vdots \\ \mathbb{P}_P[f(x) = K, y = 1] & \cdots & \mathbb{P}_P[f(x) = K, y = K] \end{bmatrix} \begin{bmatrix} w(1) \\ \vdots \\ w(K) \end{bmatrix}$$

$$q = Cw \Rightarrow w = C^{-1}q$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Supervised Learning with Label Shift

- **Input:** Training dataset Z, unlabeled test dataset X
- **Step 1:** Train f on Z
- **Step 2:** Estimate using the dataset:
  - $C_{ij} = \mathbb{P}_P[f(x) = i, y = j] \approx |Z|^{-1} \sum_{(x,y)\in Z} 1(f(x) = i \wedge y = j)$
  - $q_i = \mathbb{P}_Q[f(x) = i] \approx |X|^{-1} \sum_{x\in X} 1(f(x) = i)$
- **Step 3:** Compute $w = C^{-1} q$
- **Step 4:** Compute $\hat{\theta} = \arg\min_{\theta} \sum_{(x,y)\in Z} \ell(\theta; x, y) \cdot w(y)$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Tutorial on Label Shift

https://colab.research.google.com/drive/1fpxfcIJW5UxxX72fsS98a0fkXSEeTRGr?usp=drive_link

Agenda

- Robustness
  - Covariate shift (this week)
  - Adversarial robustness (this + next week)
  - Certifying robustness (next week)
  - Calibrated predictions & uncertainty (next next week)

Administrative Notes

- Final Exam:
  - 13 January 16:30
- Paper selection finalized except for two projects
- Project milestones
  - **1. Milestone (November 23, midnight):**
    - Read & understand the paper
    - Download the datasets
    - Prepare the Readme file excluding the results & conclusion
  - **2. Milestone (December 7, midnight)**
    - The results of the first experiment
  - **3. Milestone (January 4, midnight)**
    - Final report (Readme file)
    - Repo with all code & trained models

Background: Baye's Theorem

- For events A and B:

$$p(A) = \frac{p(B|A)p(A)}{p(B)} = \frac{p(A,B)}{p(B)}$$

Background: Marginalization

- Marginal distribution: The distribution of some variables without reference/condition on other variables.
- Example:
  - X, Y: Two discrete random variables
  - Marginal distribution of X:

$$p(x_i) = \sum_j p(x_i, y_j) = \sum_j p(x_i \mid y_j) p(y_j)$$

  - Marginal distribution of Y:

$$p(y_i) = \sum_j p(x_j, y_i) = \sum_j p(y_i \mid x_j) p(x_j)$$

  - These turn into integrals in continuous cases:

$$p(x) = \int_y p(x, y) \mathrm{d}y = \int_y p(x \mid y) p(y) \, \mathrm{d}y$$

$$p(y) = \int_x p(x, y) \mathrm{d}x = \int_x p(y \mid x) p(x) \, \mathrm{d}x$$

Y \ X	x_1	x_2	x_3	x_4	$p_Y(y)$ ↓
y_1	$\frac{4}{32}$	$\frac{2}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{8}{32}$
y_2	$\frac{3}{32}$	$\frac{6}{32}$	$\frac{3}{32}$	$\frac{3}{32}$	$\frac{15}{32}$
y_3	$\frac{9}{32}$	0	0	0	$\frac{9}{32}$
$p_X(x)$ →	$\frac{16}{32}$	$\frac{8}{32}$	$\frac{4}{32}$	$\frac{4}{32}$	$\frac{32}{32}$

Table from Wikipedia

Background:
Independent and identically distributed (i.i.d.)

- *"a collection of random variables is **independent** and **identically distributed** (i.i.d., iid, or IID) if each random variable has the same probability distribution as the others and all are mutually independent."* – Wikipedia
  - **identically distributed**: When samples are taken from the distribution, there is no overall trend / change / shift / fluctuation in the distribution.
    - More formally: $F_X(x) = F_Y(x)$ for $F_X(x) = P(X \leq x)$ and $F_Y(y) = P(Y \leq y)$ being the cumulative distributions.
  - **independent**: Samples are not "connected" to other samples. I.e., knowing one sample does not say anything about other samples.
    - More formally: $p(A, B) = p(A) \cdot p(B)$.
- Synonym with "random variable"

Background:
Independent and identically distributed (i.i.d.)

Example from Wikipedia:

Toss a coin 10 times and write down the results into variables $A_1, \ldots, A_{10}$.

1. **Independent**: Each outcome A_i will not affect the other outcome A_j (for $i \neq j$ from 1 to 10), which means the variables $A_1, \ldots, A_{10}$ are independent of each other.
2. **Identically distributed**: Regardless of whether the coin is fair (with a probability of 1/2 for heads) or biased, as long as the same coin is used for each flip, the probability of getting heads remains consistent across all flips.

Background: Infimum (inf) and Supremum (sup)

- Infimum
  - Greatest lower bound

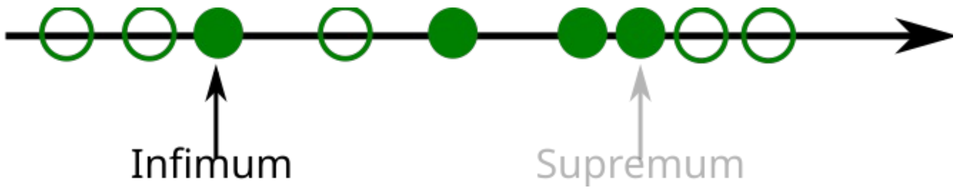


Fig: https://en.wikipedia.org/wiki/Infimum_and_supremum

- Supremum
  - Lowest upper bound

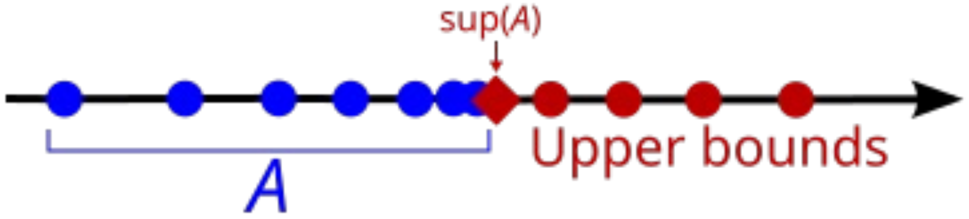


Fig: https://en.wikipedia.org/wiki/Infimum_and_supremum

Differences to min and max:

- inf and sup may not be part of the set

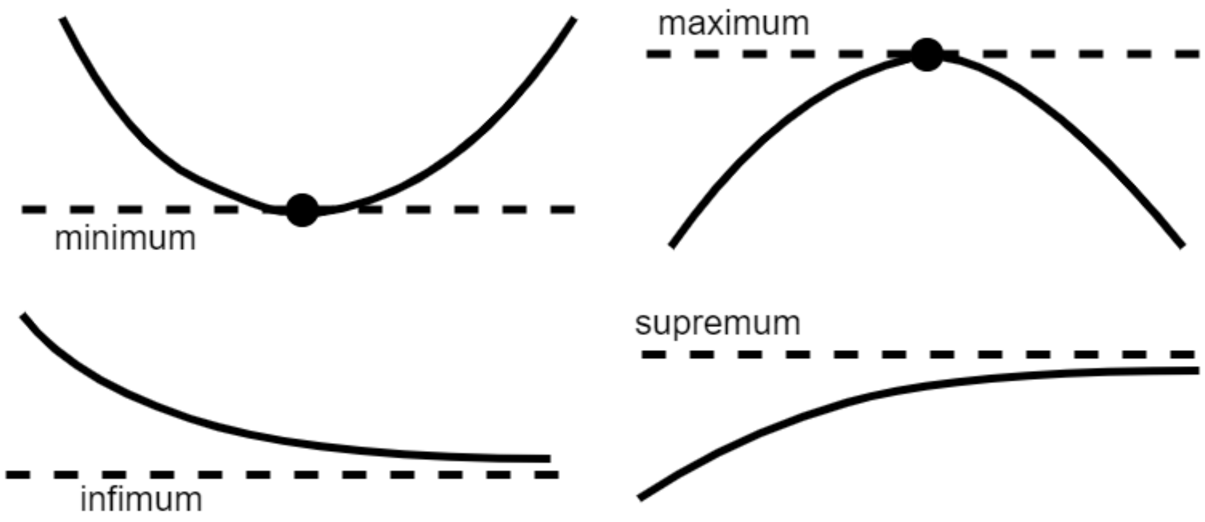


Fig: https://www.researchgate.net/figure/Minimum-maximum-infimum-and-supremum-of-example-functions_fig1_355093246

Background: Binomial Distribution

- If a random variable X follows the Binomial Distribution with parameter $n \in \mathbb{N}$ and $p \in [0, 1]$,
- then the probability of getting k successes (with same rate p) in n Bernoulli trials:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

with the binomial coefficient:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

- $E[X] = np$
- $Var(X) = np(1-p)$

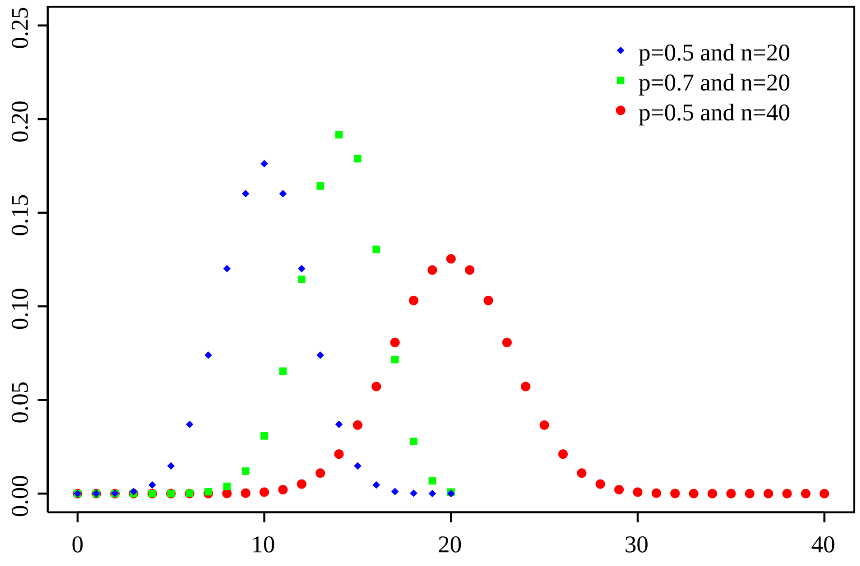


Fig: Wikipedia

Example from Wikipedia:

Suppose a biased coin comes up heads with probability 0.3 when tossed. The probability of seeing exactly 4 heads in 6 tosses is

$$f(4, 6, 0.3) = \binom{6}{4} 0.3^4 (1-0.3)^{6-4} = 0.059535.$$

Background: Bernoulli Distribution

- Discrete probability distribution where a variable can take two values
- X: a random Bernoulli variable, then:
  - $P(X = 1) = p$
  - $P(X = 0) = q = 1 - p$
  - $E[X] = p$
  - $Var[X] = p(1 - p)$
- A special case of Binomial Distribution with $n = 1$

Background: Lipschitz Constant

- For a function $f(\cdot)$:

$$K = \sup_{x_1 \neq x_2} \frac{|f(x_2) - f(x_1)|}{|x_2 - x_1|}$$

- Alternative definition:

$$|f(x_2) - f(x_1)| \leq K|x_2 - x_1|$$

- If the function is differentiable with bounded derivative:

$$K = \sup_{x} |f'(x)|$$

Background: Two-sample Test

- Given samples from two different distributions P and Q, determine whether the difference between the samples is statistically significant.
- Requires prior information about the distributions P and Q.
- Apply on samples/population or distribution parameters (mean and variance)
- Many alternatives:
  - Student’s t-test
  - Mann-Whitney U test
  - Welch’s t-test
  - …

Example

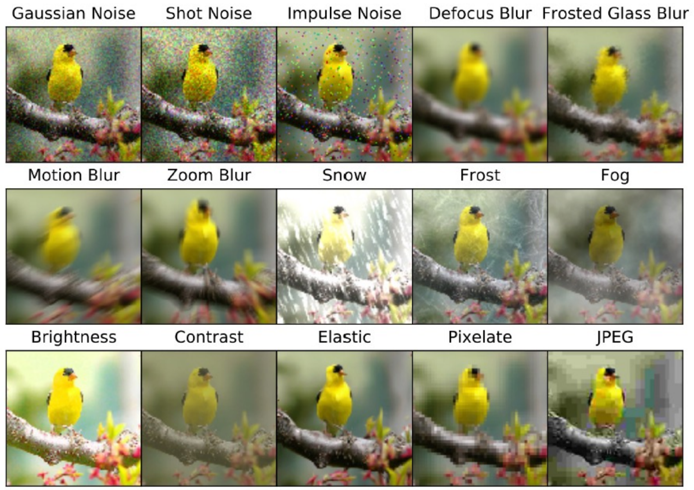
Student’s t-test

- Assumes same sample size and variance:

$$t = \frac{\bar{P} - \bar{Q}}{s_p\sqrt{2/n}}$$

where $s_p = \sqrt{\frac{s_P^2 + s_Q^2}{2}}$ is the combined standard deviation.

Higher t => Larger gap between P and Q.



Hendrycks & Dietterich, Benchmarking Neural Network Robustness to Common Corruptions and Perturbations

Importance Weights for Covariate Shift

$p(x) \neq q(x)$ but $p(y \mid x) = q(y \mid x)$

Importance Weights for Covariate Shift

- In the covariate shift setting, we have

$$w(x,y) = \frac{q(x,y)}{p(x,y)}$$
$$= \frac{q(y|x)q(x)}{p(y|x)p(x)}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

- In the covariate shift setting, we have

$$
\begin{aligned}
w(x, y) &= \frac{q(x,y)}{p(x,y)} \\
&= \frac{q(y|x)q(x)}{p(y|x)p(x)} \\
&= \frac{q(x)}{p(x)} \\
&:= w(x)
\end{aligned}
$$

Remember:

- In the label shift setting, we have

$$
\begin{aligned}
w(x, y) &= \frac{q(x,y)}{p(x,y)} \\
&= \frac{q(x|y)q(y)}{p(x|y)p(y)} \\
&= \frac{q(y)}{p(y)} \\
&:= w(y)
\end{aligned}
$$

Adapted from: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

- If we know $w(x)$, then we have

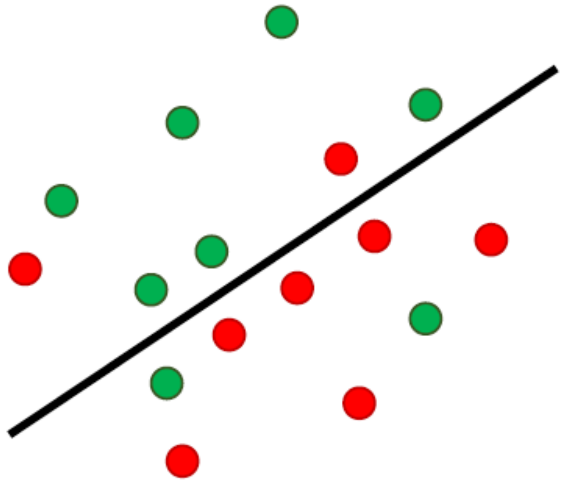
$$\mathbb{E}_Q[\ell(\theta; x, y)] = \mathbb{E}_P[\ell(\theta; x, y) \cdot w(x, y)] = \mathbb{E}_P[\ell(\theta; x, y) \cdot w(x)]$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights

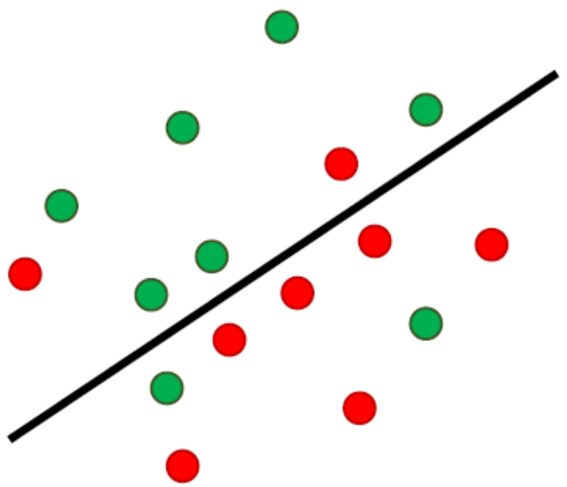
- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$



Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights



- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \text{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$

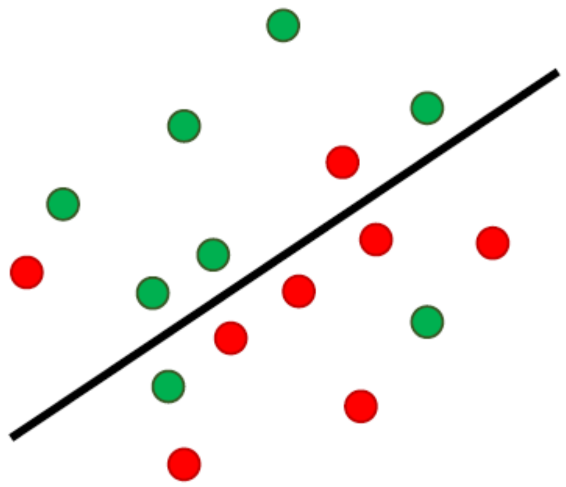
- Suppose we know $r(b \mid x)$, then by Bayes' rule, we have

$$r(b = 0 \mid x) = \frac{r(x \mid b = 0)r(b = 0)}{r(x \mid b = 0)r(b = 0) + r(x \mid b = 1)r(b = 1)}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights



- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$

- Suppose we know $r(b \mid x)$, then by Bayes' rule, we have

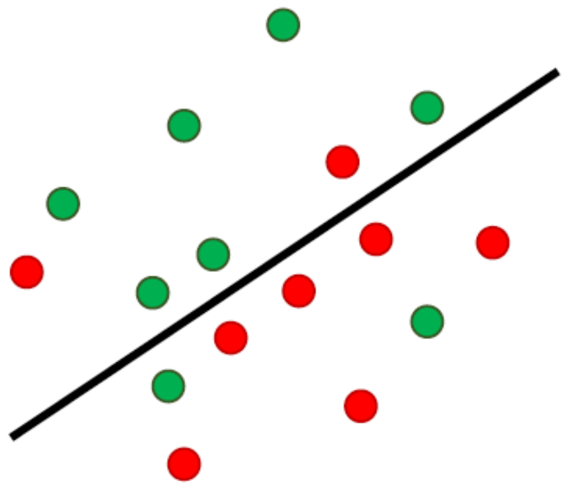
$$r(b = 0 \mid x) = \frac{r(x \mid b = 0) \cdot \frac{1}{2}}{r(x \mid b = 0) \cdot \frac{1}{2} + r(x \mid b = 1) \cdot \frac{1}{2}}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights

- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$



- Suppose we know $r(b \mid x)$, then by Bayes' rule, we have

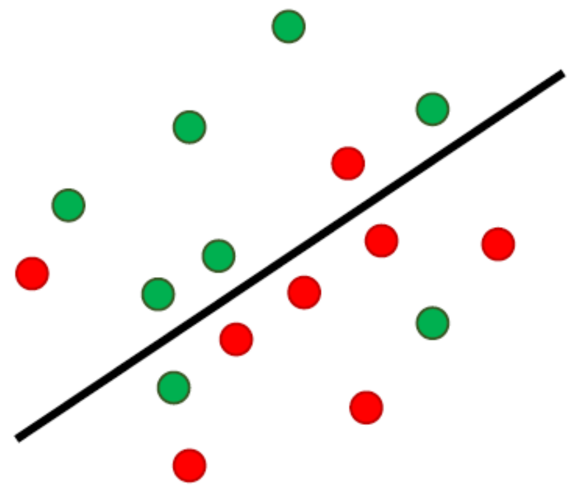
$$r(b = 0 \mid x) = \frac{r(x \mid b = 0)}{r(x \mid b = 0) + r(x \mid b = 1)}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights

- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
    - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
    - If $b = 0$, then sample $(x,\cdot) \sim P$
    - If $b = 1$, then sample $(x,\cdot) \sim Q$



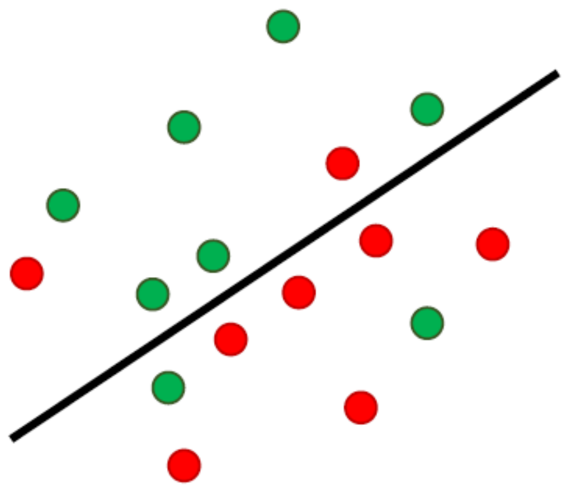
- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{p(x)}{p(x) + q(x)}$$

Importance Weights for Covariate Shift

Estimating the weights

- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \text{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$



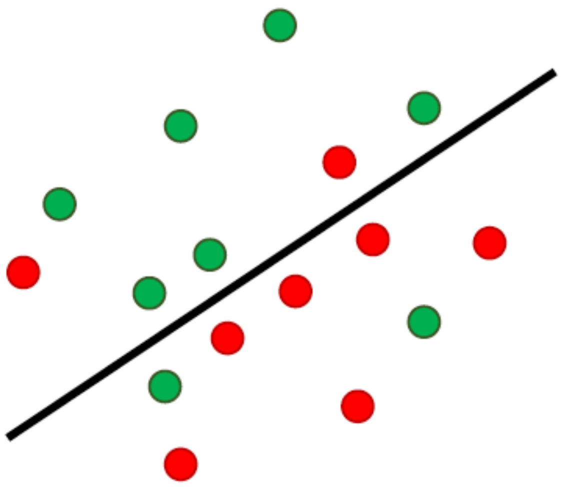
- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{1}{1 + \frac{q(x)}{p(x)}}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights



- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \text{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x, \cdot) \sim P$
  - If $b = 1$, then sample $(x, \cdot) \sim Q$

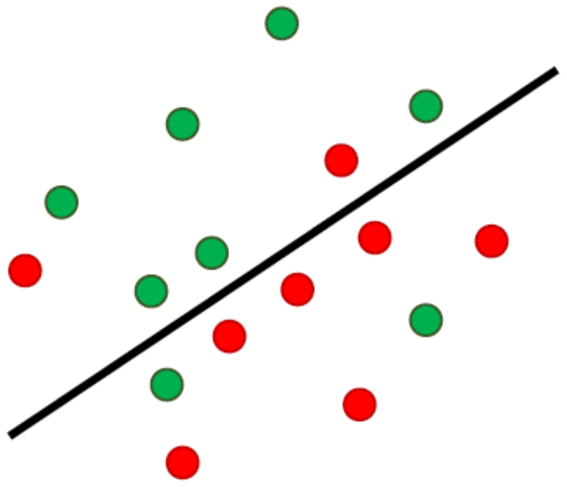
- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{1}{w(x) + 1}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights



- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$

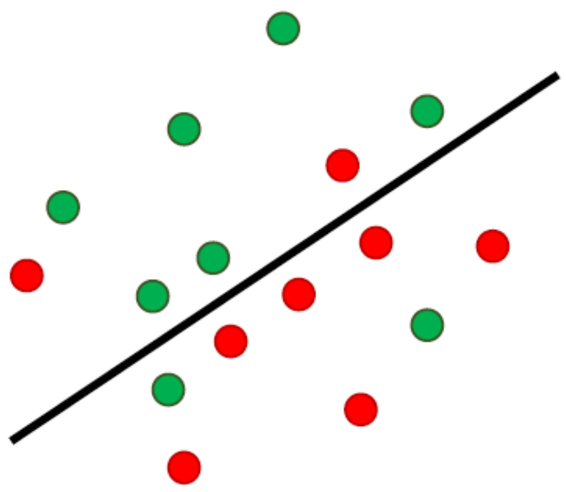
- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$w(x) + 1 = \frac{1}{r(\, b = 0 \mid x \,)}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights for Covariate Shift

Estimating the weights



- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$

- Suppose we know $r(\, b \mid x\,)$, then by Bayes' rule, we have

$$w(x) = \frac{1}{r(\, b = 0 \mid x\,)} - 1$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Estimating Source-Target Probability

- We can construct a dataset of i.i.d. samples $(x, b) \sim R$
  - For simplicity, assume that $|X| = |Z|$
  - Then, consider

$$X' = \{ (x, 0) \mid (x, y) \in Z \} \cup \{ (x, 1) \mid x \in X \}$$

  - This dataset consists of i.i.d. samples $(x, b) \sim R$

Estimating Source-Target Probability

- We can construct a dataset of i.i.d. samples $(x, b) \sim R$
  - For simplicity, assume that $|X| = |Z|$
  - Then, consider

$$X' = \{ (x, 0) \mid (x, y) \in Z \} \cup \{ (x, 1) \mid x \in X \}$$

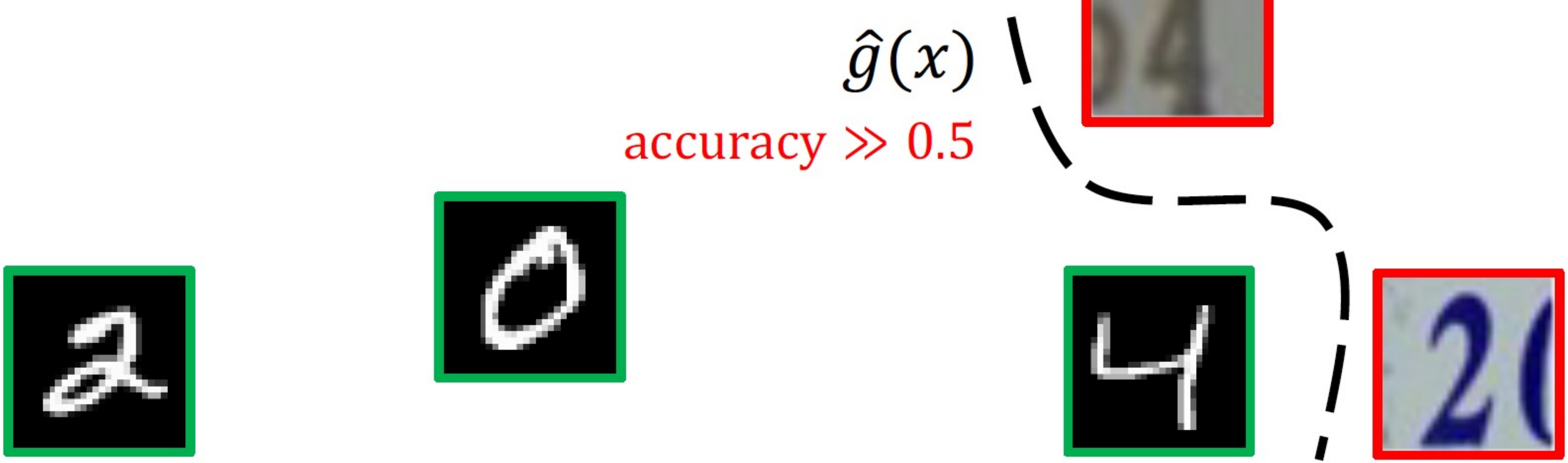
  - This dataset consists of i.i.d. samples $(x, b) \sim R$

- Given i.i.d. samples $(x, b) \sim R$, then $r(\, b = 1 \mid x \,)$ is the same as the probability of "label" b given "input" x
  - **Idea:** Train a model (called a **discriminator**) on X' to predict b given x

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Discriminators

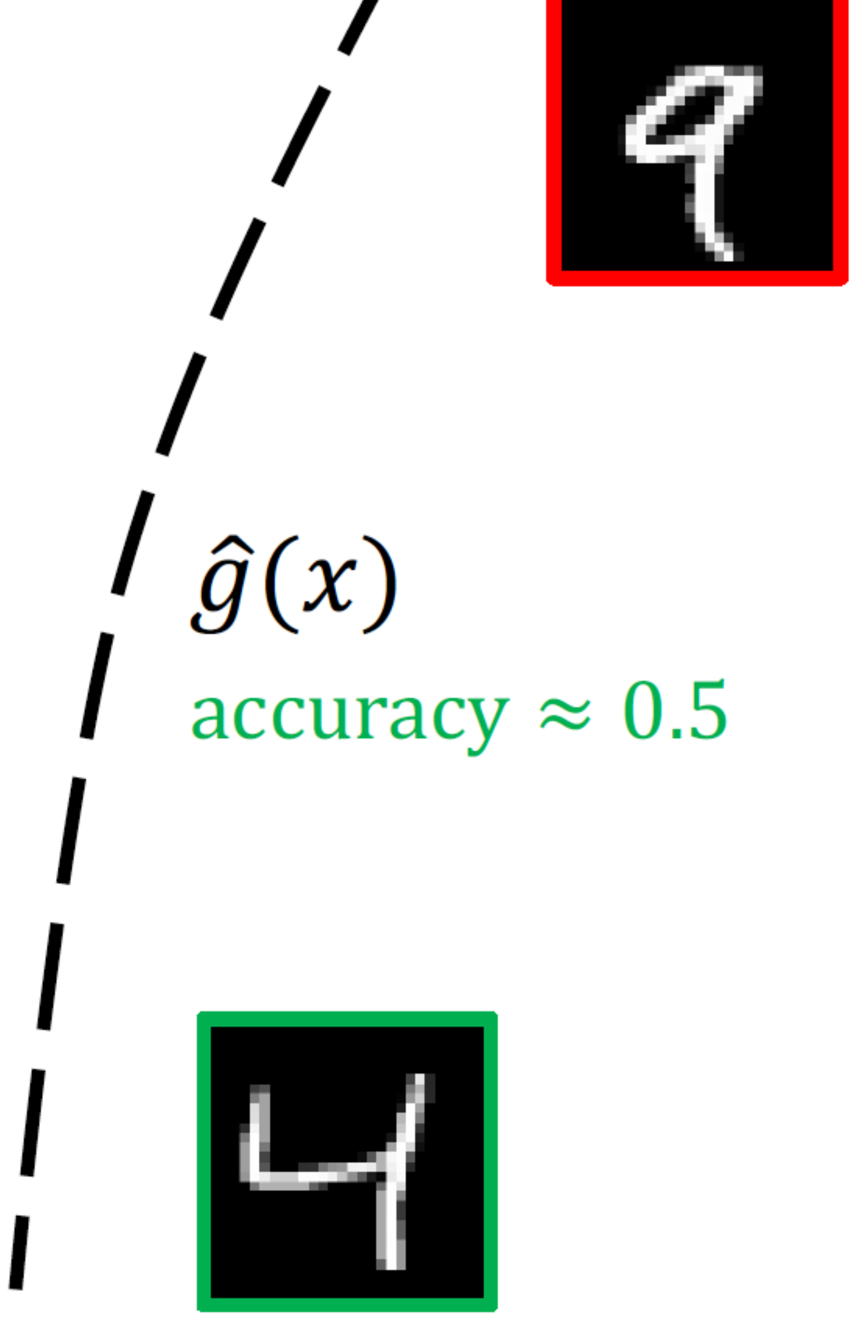
- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**



Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Discriminators

- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**
- $\hat{g}$ has **low accuracy** $\Rightarrow$ **small shift** (assuming sufficient capacity)



Supervised Learning with Covariate Shift

- **Input:** Training dataset Z, unlabeled test dataset X
- **Step 1:** Construct $X' = \{ (x, 0) \mid (x, y) \in Z \} \cup \{ (x, 1) \mid x \in X \}$ and train $\hat{g}$ on X' to predict b given x
- **Step 2:** Compute $w(x) = \frac{1}{\hat{g}(b=1|x)} - 1$
- **Step 3:** Compute $\hat{\theta} = \arg\min_{\theta} \sum_{(x,y)\in Z} \ell(\theta; x, y) \cdot w(x)$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Importance Weights

- **Pros:**
  - Principled technique for addressing distribution shift
  - "Granular" quantification of shift (obtain an estimate of the shift for each example, not just just the overall shift)

- **Cons:**
  - Does not work when support of Q is not contained in support of P
  - Even if the above is satisfied, importance weights are large if $P(x, y)$ is small

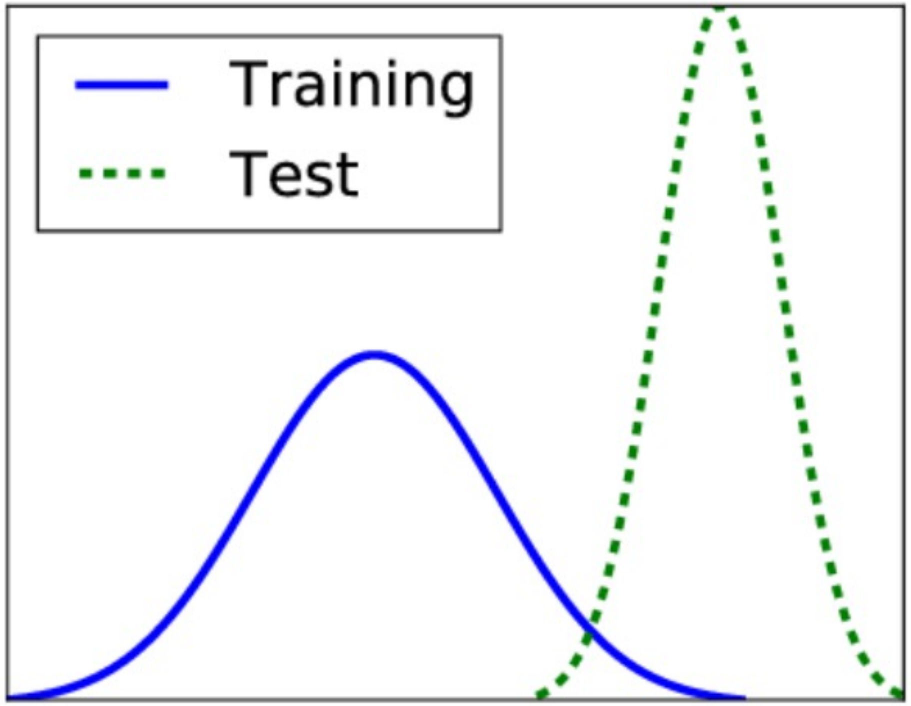
Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Methods for Distributional Robustness

Using Penalty Terms

Support of Shifted Data

- **Assumption:** Support of Q is not contained in support of P

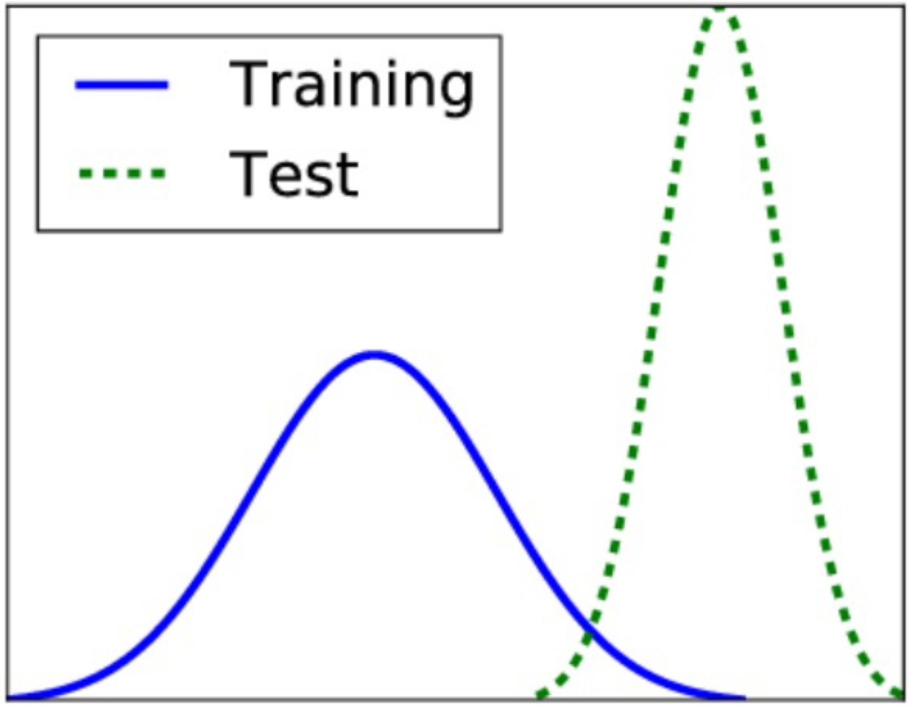


Image; Glauner et al., 2018

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Support of Shifted Data

- **Assumption:** Support of Q is not contained in support of P
- However, this is **necessary** since we do not know anything about data outside of the support of P
- Need additional assumptions to do better
  - Focus on covariate shift



Image; Glauner et al., 2018

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Learning vs. Evaluation

- For this part, we will focus on **model evaluation**
  - **Learning:** Optimize $\mathbb{E}_Q[\ell(\theta; x, y)]$
  - **Evaluation:** Estimate $\mathbb{E}_Q[\ell(\theta; x, y)]$

- We will see why learning is harder later

Integral Probability Metrics

- The **total variation distance** is

$$\mathrm{TV}(P, Q) = \int_{\mathcal{X}\times\mathcal{Y}} |q(x, y) - p(x, y)| \cdot dx \cdot dy$$

Sensitive to the mis-alignment between the two distributions. No notion of cost between two points.

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Integral Probability Metrics

- The **total variation distance** is

$$\mathrm{TV}(P,Q) = \int_{\mathcal{X}\times\mathcal{Y}} |q(x,y) - p(x,y)| \cdot dx \cdot dy$$

Sensitive to the mis-alignment between the two distributions. No notion of cost between two points.

- The **Wasserstein distance** is

$$W(P,Q) = \sup_{f:K_f\leq 1} \int_{\mathcal{X}\times\mathcal{Y}} f(x,y) \cdot \big(q(x,y) - p(x,y)\big) \cdot dx \cdot dy$$

$f(x,y)$: A smooth-function (Lipschitz constant: 1) that measures how far apart the two distributions are.
We are trying to find $f(x,y)$ that best separates the distributions.

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Background: Wasserstein Distance Metric

Also called: Kantorovich-Rubinstein metric, Earth Mover's Distance

- A measure between two probability distributions P and Q.
- The minimum "cost" of moving pile from P to Q.
  - A measure of optimal transport problem.
- For samples $X \sim P$ and $Y \sim Q$:

$$W(P,Q) = \left(\frac{1}{n}\sum_{i=1}^{n} \|X_i - Y_i\|^p\right)^{1/p}$$

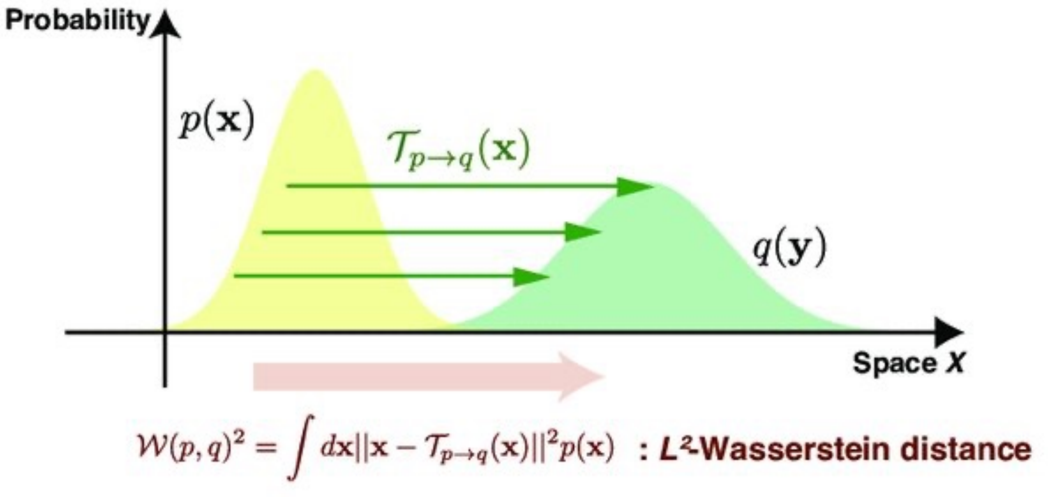


Fig: https://www.researchgate.net/publication/349704621_Geometrical_aspects_of_entropy_production_in_stochastic_thermodynamics_based_on_Wasserstein_distance/figures?lo=1

Evaluation Bounds

- Note that

$$
\begin{aligned}
\mathbb{E}_Q[\ell(\theta; x, y)] &= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot q(x, y) \cdot dx \cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(p(x, y) + q(x, y) - p(x, y)\big) \cdot dx \cdot dy
\end{aligned}
$$

Evaluation Bounds

- Note that

$$
\begin{aligned}
\mathbb{E}_Q[\ell(\theta; x, y)] &= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot q(x, y) \cdot dx \cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(p(x, y) + q(x, y) - p(x, y)\big) \cdot dx \cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot p(x, y) \cdot dx \cdot dy \\
&\qquad + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(x, y) - p(x, y)\big) \cdot dx \cdot dy
\end{aligned}
$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Evaluation Bounds

- Note that

$$
\begin{aligned}
\mathbb{E}_Q[\ell(\theta;x,y)] &= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot q(x,y)\cdot dx\cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(p(x,y)+q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot p(x,y)\cdot dx\cdot dy \\
&\qquad + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&= \mathbb{E}_P[\ell(\theta;x,y)] + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(q(x,y)-p(x,y)\big)\cdot dx\cdot dy
\end{aligned}
$$

Evaluation Bounds

- Note that

$$
\begin{aligned}
\mathbb{E}_Q[\ell(\theta;x,y)] &= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot q(x,y)\cdot dx\cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(p(x,y)+q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot p(x,y)\cdot dx\cdot dy \\
&\qquad + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&= \mathbb{E}_P[\ell(\theta;x,y)] + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&\leq \mathbb{E}_P[\ell(\theta;x,y)] + \ell_{\max}\cdot \int_{\mathcal{X}\times\mathcal{Y}} |q(x,y)-p(x,y)|\cdot dx\cdot dy \\
&= \mathbb{E}_P[\ell(\theta;x,y)] + \ell_{\max}\cdot \mathrm{TV}(P,Q)
\end{aligned}
$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Evaluation Bounds

- Note that

$$
\begin{aligned}
\mathbb{E}_Q[\ell(\theta; x, y)] &= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot q(x, y) \cdot dx \cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(p(x, y) + q(x, y) - p(x, y)\big) \cdot dx \cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot p(x, y) \cdot dx \cdot dy \\
&\qquad + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(x, y) - p(x, y)\big) \cdot dx \cdot dy \\
&= \mathbb{E}_P[\ell(\theta; x, y)] + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(x, y) - p(x, y)\big) \cdot dx \cdot dy \\
&\leq \mathbb{E}_P[\ell(\theta; x, y)] + K_\ell \cdot W(P, Q)
\end{aligned}
$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Evaluation Bounds for Covariate Shift

- Note that

$$\int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(x, y) - p(x, y)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(y \mid x)q(x) - p(y \mid x)p(x)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(p(y \mid x)q(x) - p(y \mid x)p(x)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot p(y \mid x) \cdot \big(q(x) - p(x)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \ell(\theta; x, y) \cdot p(y \mid x) \cdot dy\right) \cdot \big(q(x) - p(x)\big) \cdot dx$$

$$= \int_{\mathcal{X}} \tilde{\ell}(\theta; x) \cdot \big(q(x) - p(x)\big) \cdot dx$$

$$\leq K_{\tilde{\ell}} \cdot W\big(P(x), Q(x)\big)$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Evaluation Bounds for Covariate Shift

- Thus, we have

$$\mathbb{E}_Q[\ell(\theta; x, y)] \leq \mathbb{E}_P[\ell(\theta; x, y)] + K_{\tilde{\ell}} \cdot W\big(P(x), Q(x)\big)$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Aside: What About Learning?

- Suppose we optimize the upper bound:

$$\mathbb{E}_Q[\ell(\theta; x, y)] \leq \mathbb{E}_P[\ell(\theta; x, y)] + K_{\tilde{\ell}} \cdot W\big(P(x), Q(x)\big)$$

- It is equivalent to optimizing $\mathbb{E}_P[\ell(\theta; x, y)]$, since the penalty is independent of θ

- Need new approaches to use such bounds for learning

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Evaluation Bounds

- Need to evaluate the metric $\mathrm{TV}(P, Q)$ or $W(P, Q)$
  - $\mathrm{TV}(P, Q)$ is harder to estimate
  - $W(P, Q)$ can be estimated heuristically

- We focus on covariate shift

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Evaluation Bounds

- **Basic idea:** Train a discriminator with bounded Lipschitz constant
  - Construct $X' = \{ (x, 0) \mid (x, y) \in Z \} \cup \{ (x, 1) \mid x \in X \}$
  - Train $\hat{g}$ on X' **but bound its Lipschitz constant $K_{\hat{g}} \leq 1$**

- Use the Wasserstein distance as the training loss:

$$\begin{aligned}
\hat{g} &= \sup_{f : K_f \leq 1} \int_{\mathcal{X}} f(x) \cdot \big(q(x) - p(x)\big) \cdot dx \cdot dy \\
&= \sup_{f : K_f \leq 1} \{\mathbb{E}_Q[f(x)] - \mathbb{E}_P[f(x)]\} \\
&\approx \sup_{f : K_f \leq 1} \{n^{-1} \textstyle\sum_{(x,1) \in X'} f(x) - n^{-1} \sum_{(x,0) \in X'} f(x)\}
\end{aligned}$$

We are trying to find the function f() that best separates the two distributions.

Training Lipschitz Neural Networks

- **Simple strategy:** Bound weight matrices individually
  - For example, $g = g_m \circ g_{m-1} \circ \cdots \circ g_1$, then $K_g \leq K_{g_m} \cdot K_{g_{m-1}} \cdot \cdots \cdot K_{g_1}$

- For a single layer
  - If $g_j(x) = W_j x$ is linear, we have $K_{g_j} = \|W\|_{op}$
  - Here, $\|W\|_{op}$ is the operator norm $\|W\|_{op} = \max_x \frac{\|Wx\|_1}{\|x\|_1}$
  - If $g_j(x) = \mathrm{ReLU}(x)$, we have $K_{g_j} = 1$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Training Lipschitz Neural Networks

- Use projected gradient descent

- For $t \in \{1, \ldots, T\}$ (or until convergence):
  - For $j \in \{1, \ldots, m\}$:

$$W_j \leftarrow W_j - \alpha \cdot \nabla_{W_j} L(W_j; Z)$$

$$W_j \leftarrow \frac{W_j}{\|W_j\|_1}$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Integral Probability Metric Penalties

- **Pros:**
  - Can handle shifts without distribution overlap

- **Cons:**
  - Requires additional assumptions about the true function (e.g., Lipschitz)
  - Cannot be used for learning, only evaluation

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Methods for Distributional Robustness

Covariate Shift Detection

Covariate Shift Detection

- **Alternative strategy:** Can we test for covariate shift?
- **Problem setting**
  - **Given:** i.i.d. samples $x_1, \dots, x_n \sim P$ and $x_1', \dots, x_n' \sim Q$ (denoted X_P and X_Q)
  - **Goal:** Determine whether $P = Q$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Covariate Shift Detection

- **Alternative strategy:** Can we test for covariate shift?

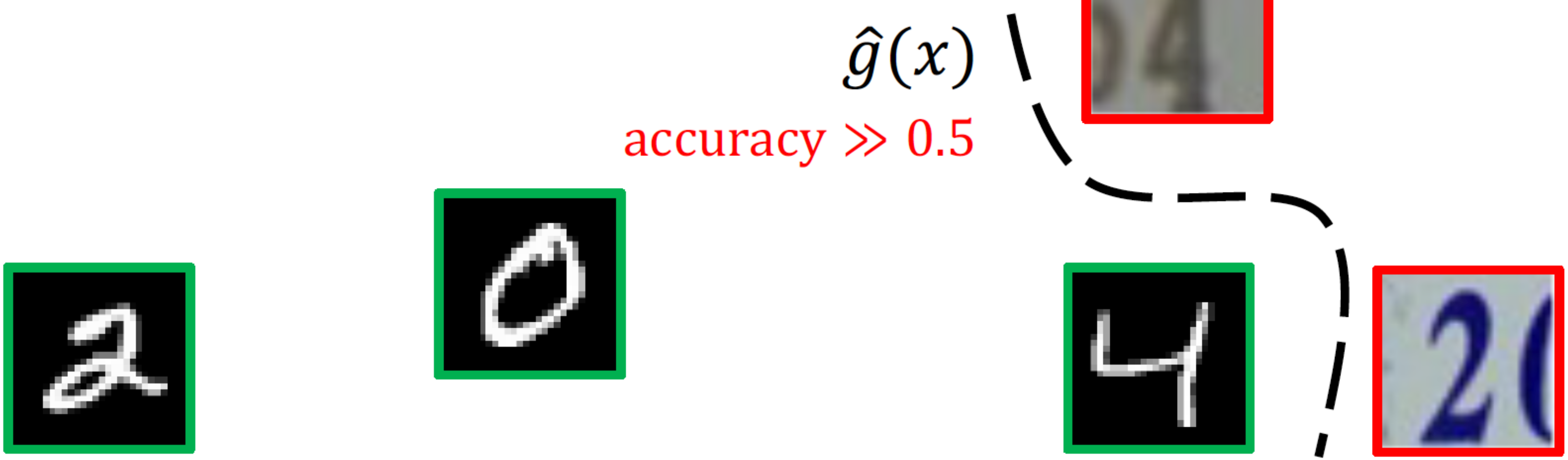
- **Problem setting**
  - **Given:** i.i.d. samples $x_1, \ldots, x_n \sim P$ and $x'_1, \ldots, x'_n \sim Q$ (denoted X_P and X_Q)
  - **Goal:** Determine whether $P = Q$

- This is a **two-sample test**
  - Lots of work on two-sample tests in the statistics literature
  - **Idea:** Can we leverage our source-target discriminator?
  - Yes! This is called a **classifier test**

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Discriminators

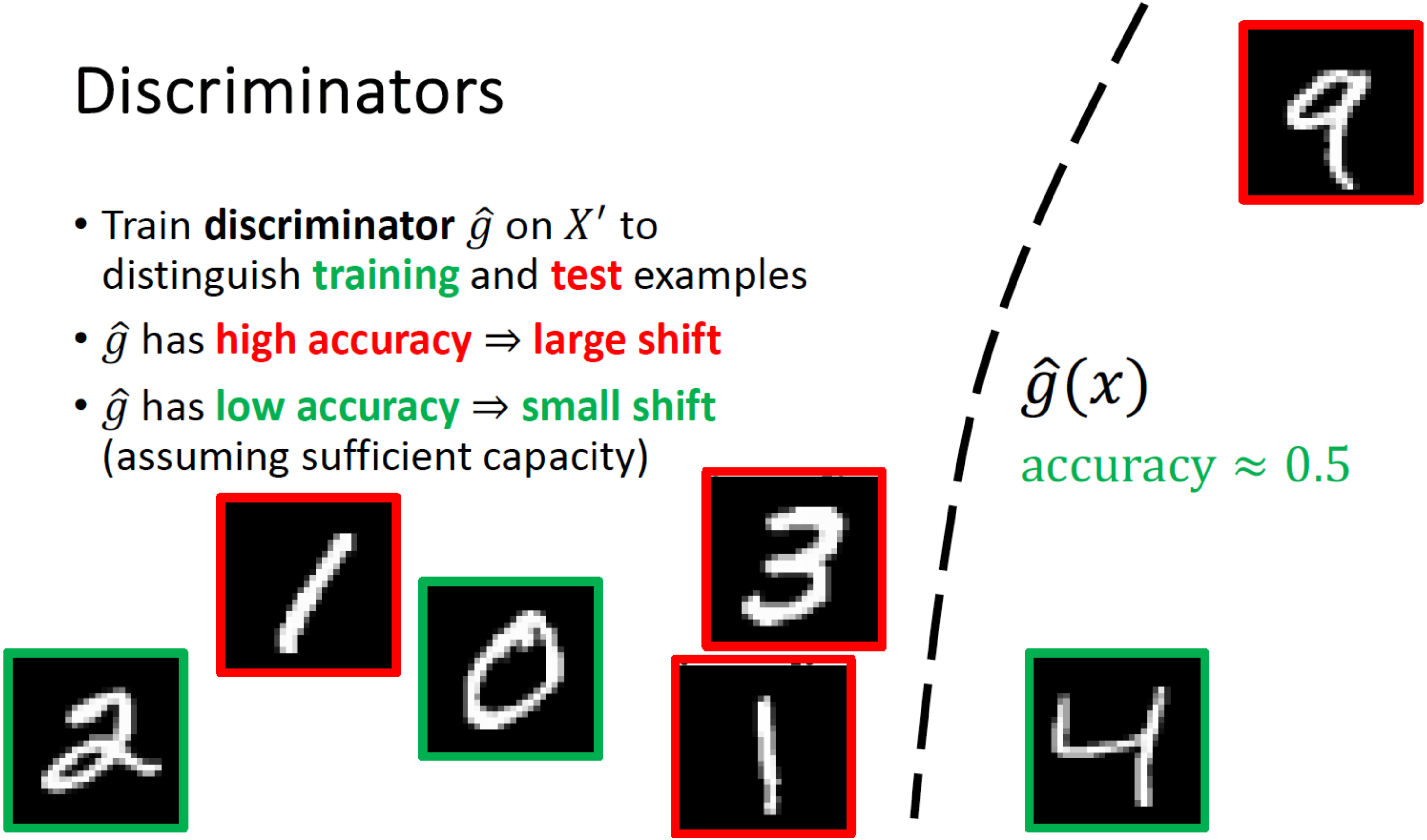
- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**



Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Discriminators

- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**
- $\hat{g}$ has **low accuracy** $\Rightarrow$ **small shift** (assuming sufficient capacity)



Covariate Shift Detection

- **Proposed approach**
  - Train discriminator $\hat{g}$ on $X' = \{(x,0) \mid x \in X_P\} \cup \left\{(x,1) \mid x \in X_Q\right\}$
  - Determine there is covariate shift if $\mathrm{Accuracy}(\hat{g}; X'') \geq \frac{1}{2} + \epsilon$
  - X'' is a held-out test set constructed the same way as X'

- **Question:** How do we choose ϵ?

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Covariate Shift Detection

- **Proposed approach**
  - Train discriminator $\hat{g}$ on $X' = \{ (x, 0) \mid x \in X_P \} \cup \{ (x, 1) \mid x \in X_Q \}$
  - Determine there is covariate shift if $\mathrm{Accuracy}(\hat{g}; X'') \geq \frac{1}{2} + \epsilon$
  - X'' is a held-out test set constructed the same way as X'

- **Question:** How do we choose ϵ?

- **Typical goal:** Choose ϵ so the probability of a false positive is bounded by a user provided error level α:

$$\mathbb{P}_{X''}[\, \mathrm{Detector}(X''; \hat{g}, \epsilon) = 1 \mid P = Q \,] \leq \alpha$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Covariate Shift Detection

- Note that $\text{Accuracy}(\hat{g}; X) = n^{-1} \sum_{i=1}^{n} 1(\hat{g}(x_i) = b_i)$
- Assuming $P = Q$, then $z_i \coloneqq 1(\hat{g}(x_i) = b_i)$ is a Bernoulli random variable with mean $\mathbb{E}[1(\hat{g}(x_i) = b_i)] = \mathbb{P}[\hat{g}(x_i) = b_i] = \frac{1}{2}$

Then,

$$S = \sum_{i=1}^{n} \mathbf{1}(\hat{g}(x_i) = b_i) \sim \text{Binomial}(n, ½)$$

$$\text{Accuracy}(\hat{g}, X'') = \frac{1}{n} S$$

$$\text{Accuracy}(\hat{g}, X'') \geq \frac{1}{2} + \epsilon$$

$$\Longrightarrow \frac{S}{n} \geq \frac{1}{2} + \epsilon \quad \Longrightarrow \quad S \geq \frac{n}{2} + n\epsilon$$

Since S (# of correct classifications) needs to be an integer:

$$S \geq \left\lceil \frac{n}{2} + n\epsilon \right\rceil$$

Adapted from: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Covariate Shift Detection

Then,

$$S = \sum_{i=1}^{n} \mathbf{1}(\hat{g}(x_i) = b_i) \sim \text{Binomial}(n, ½)$$

$$\text{Accuracy}(\hat{g}, X'') = \frac{1}{n} S$$

$$\text{Accuracy}(\hat{g}, X'') \geq \frac{1}{2} + \epsilon$$

$$\Rightarrow \frac{S}{n} \geq \frac{1}{2} + \epsilon \quad \Rightarrow \quad S \geq \frac{n}{2} + n\epsilon$$

Since S (# of correct classifications) needs to be an integer:

$$S \geq \left\lceil \frac{n}{2} + n\epsilon \right\rceil$$

$$\mathbb{P}_{X''}[\text{Detector}(\dots) = 1 \mid P = Q] = \mathbb{P}\left[S \geq \left\lceil n\left(\frac{1}{2} + \epsilon\right)\right\rceil\right]$$

Since $S \sim \text{Binomial}(n, ½)$, this probability is the sum of the probabilities of all outcomes i (# of correct predictions) from $\left\lceil \frac{n}{2} + n\epsilon \right\rceil$ to n:

$$\mathbb{P}_{X''}[\text{Detector}(\dots) = 1 \mid P = Q] = \sum_{i=\lceil n(1/2+\epsilon)\rceil}^{n} \text{Binomial}\left(i; n, \frac{1}{2}\right) \leq \alpha$$

Covariate Shift Detection

- **Step 1:** Train $\hat{g}$ on $X' = \{\,(x,0) \mid x \in X_P\,\} \cup \{\,(x,1) \mid x \in X_Q\,\}$
- **Step 2:** Compute ϵ so that $\boxed{\sum_{i=\lceil n\epsilon \rceil}^{n} \text{Binomial}\left(i; n, \frac{1}{2}\right) \le \alpha}$
- **Step 3:** Return "true" if $\text{Accuracy}(\hat{g}; X'') \ge \frac{1}{2} + \epsilon$ else "false"
  - X'' is a held-out test set constructed the same way as X'

Should be

$$\sum_{i=\lceil n(1/2+\epsilon) \rceil}^{n} \text{Binomial}\left(i; n, \frac{1}{2}\right) \le \alpha$$

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

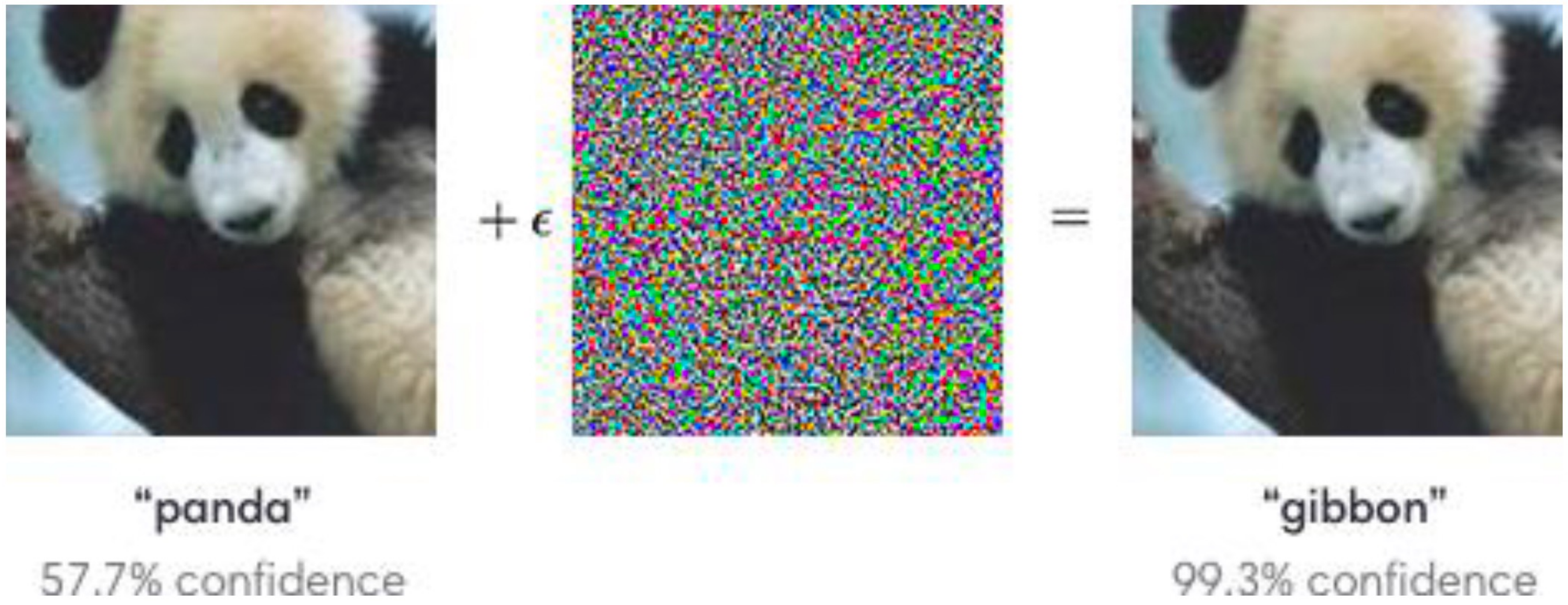
Key Takeaway

- We can get provable bounds on the true accuracy of a model $\mathbb{E}[1(\hat{g}(x_i) = b_i)]$ from the test set accuracy $n^{-1}\sum_{i=1}^{n} 1(\hat{g}(x_i) = b_i)$
- Later in the class, we will see how this idea can be used to obtain rigorous uncertainty quantification for machine learning models

Slide: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Adversarial Attacks and Robustness

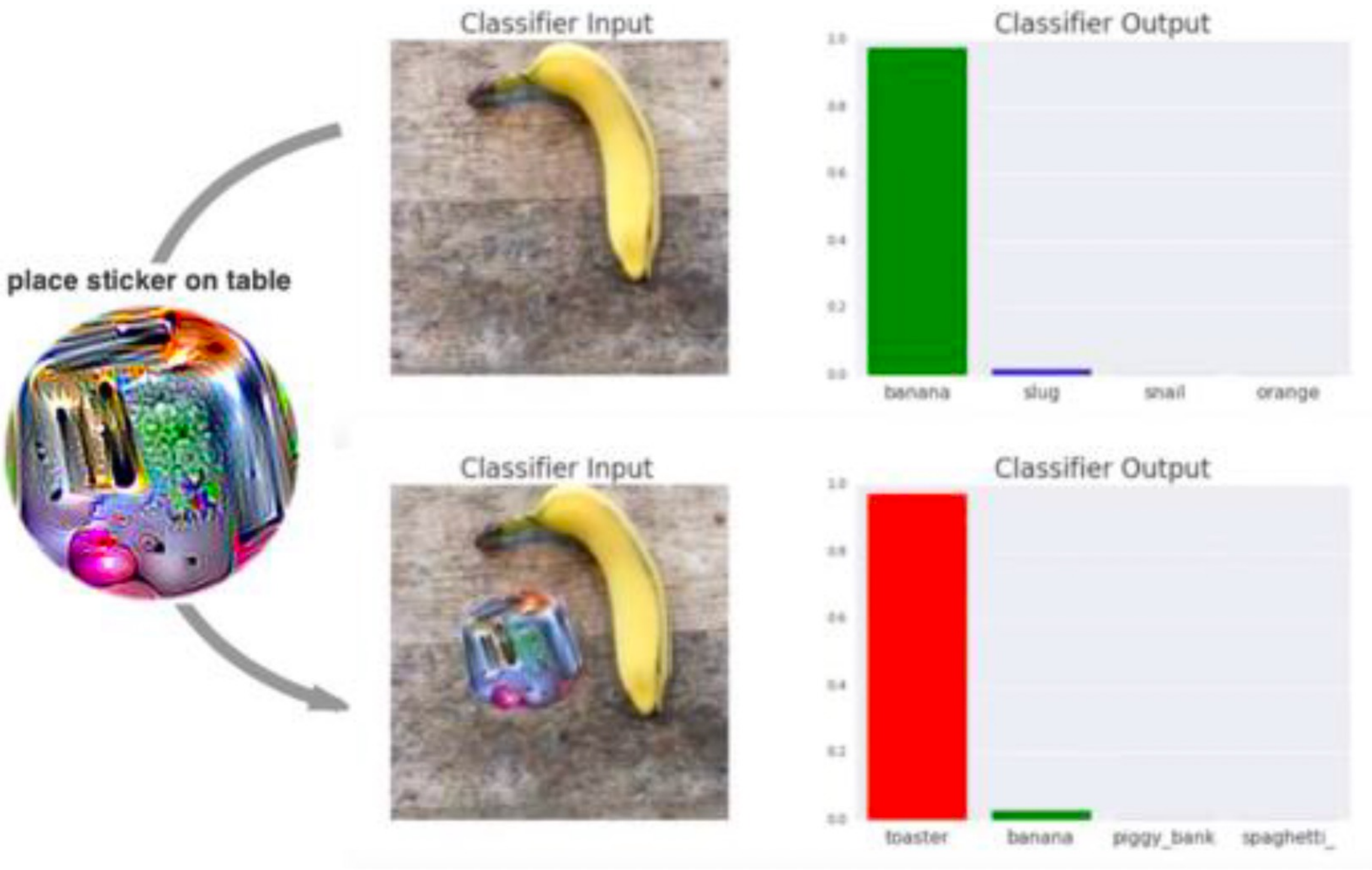
Deep networks are "sensitive"



Szegedy et al., Intriguing Properties of Neural Networks, 2014

Image: Uni of Pennsylvania, Trustworthy ML - CIS 7000

Adversarial Attacks Everywhere



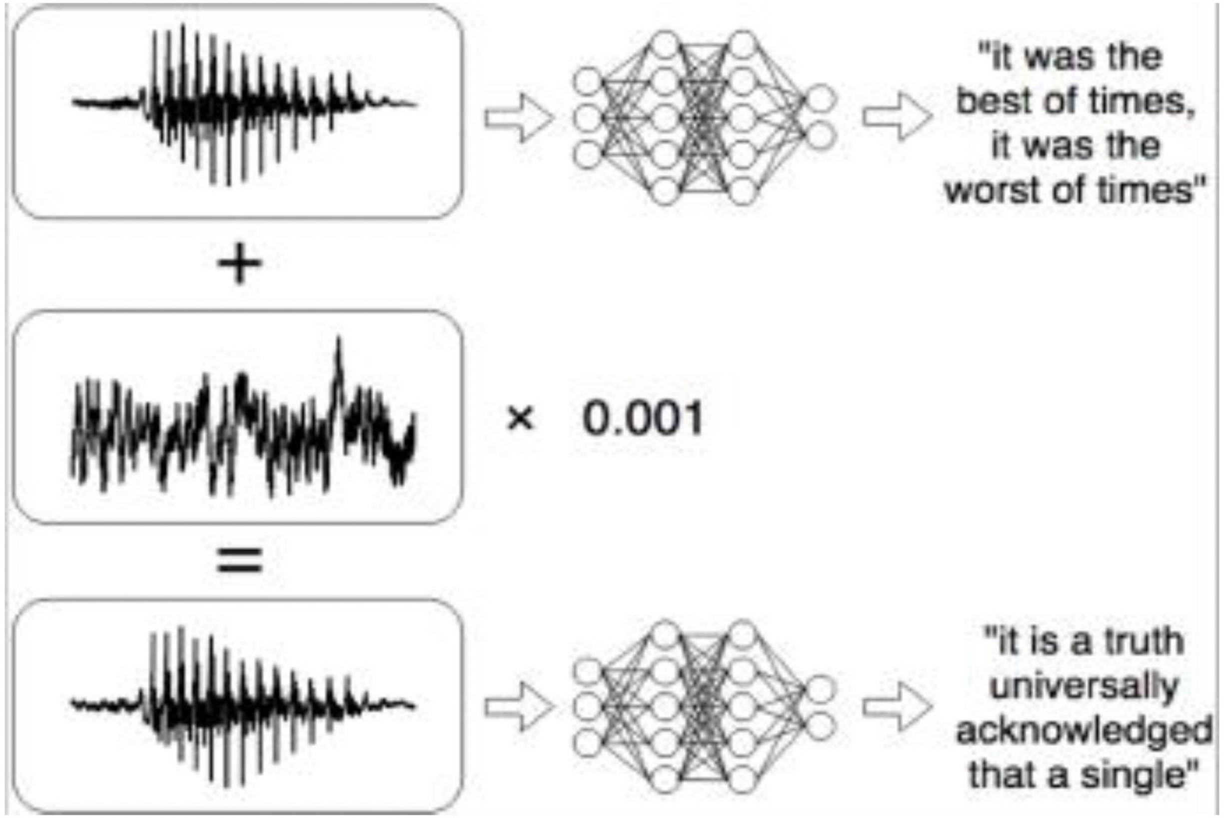
Brown et al, 2017 “Adversarial Patch”

Adversarial Attacks Everywhere

Label	Sentence
P	I am currently trying to give this company another chance. I have had the same scheduling experience as others have written about. Wrote to them today
N	I am currently trying to give this company another review. I have had the same dental experience about others or written with a name. Thanks to them today

Hsieh et al. 2019 "Natural Adversarial Sentence Generation with Gradient based Perturbation"

Adversarial Attacks Everywhere



Carlini, Wagner, 2018

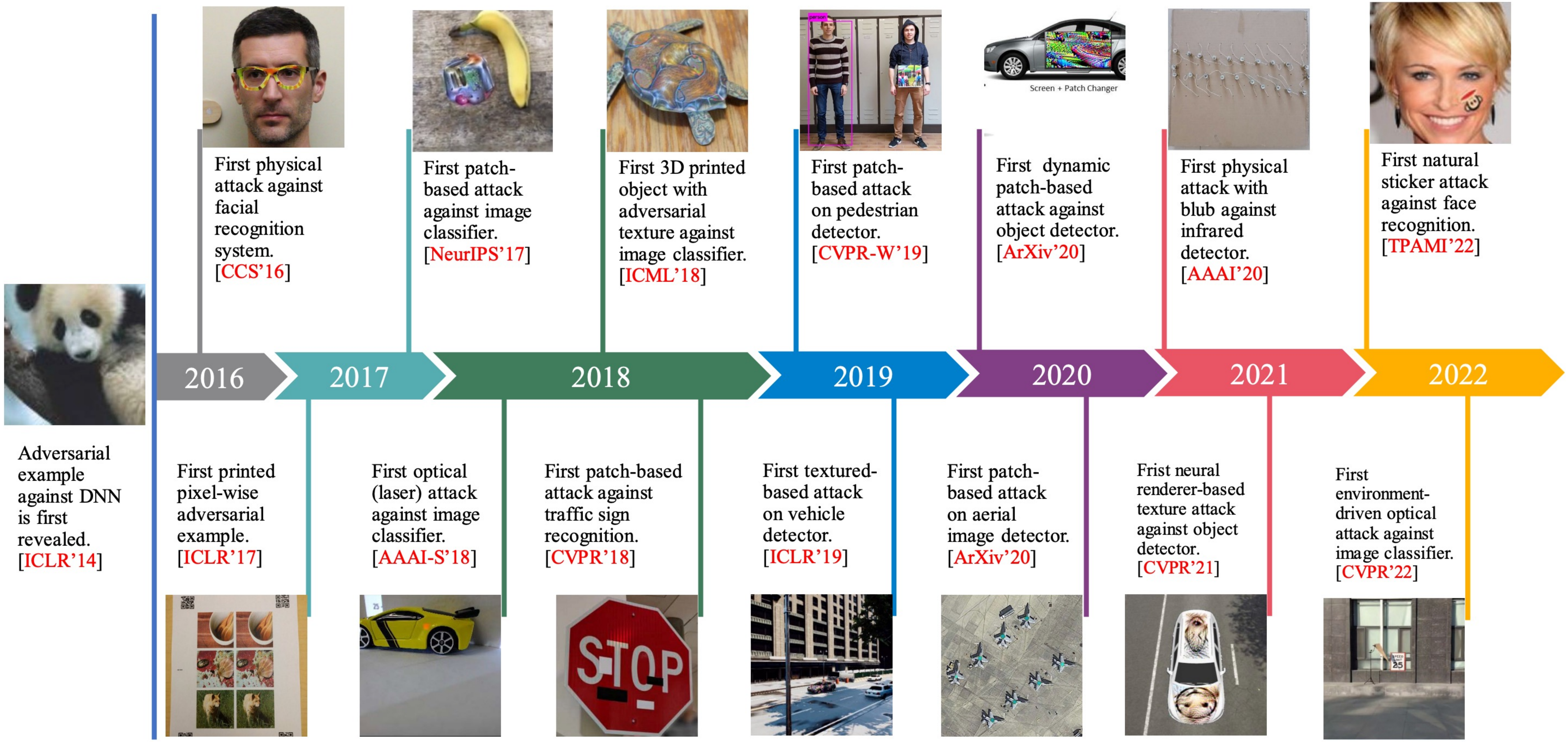
Adversarial Perturbations can be dangerous …

- **Task:**
  - Photo ID verification
  - Goal is to check whether uploaded photo matches a photo ID

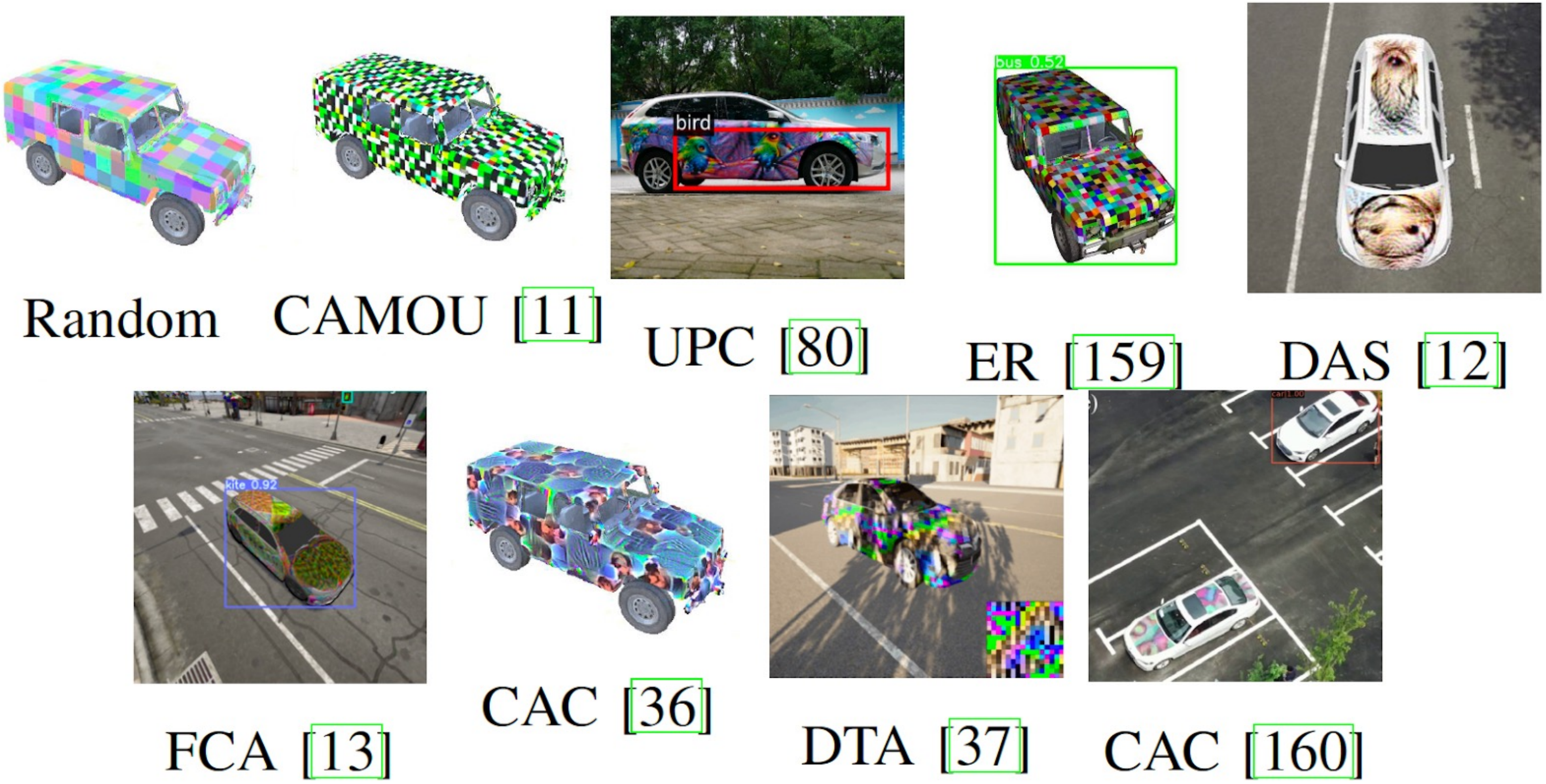
- **Attack:**
  - User perturbs their image to match the photo in the ID
  - Challenge for machine learning in online identity verification!

(Valid photo ID from Papesh 2018)

Wang, D., Yao, W., Jiang, T., Tang, G., & Chen, X. (2022). A survey on physical adversarial attack in computer vision. *arXiv preprint arXiv:2209.14262*.



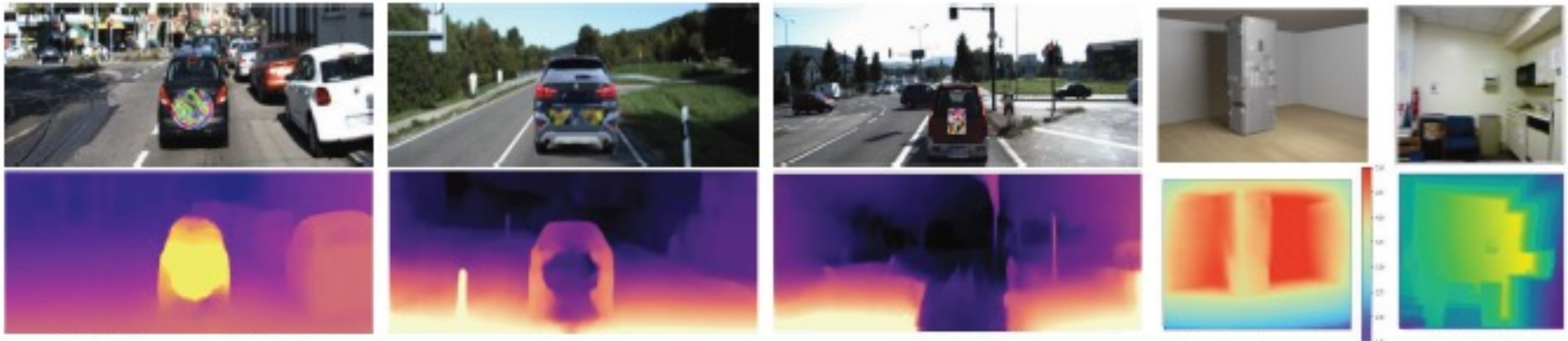
Wang, D., Yao, W., Jiang, T., Tang, G., & Chen, X. (2022). A survey on physical adversarial attack in computer vision. *arXiv preprint arXiv:2209.14262*.



Guesmi, A., Hanif, M. A., Ouni, B., & Shafique, M. (2023). Physical adversarial attacks for camera-based smart systems: Current trends, categorization, applications, research challenges, and future outlook. IEEE Access.

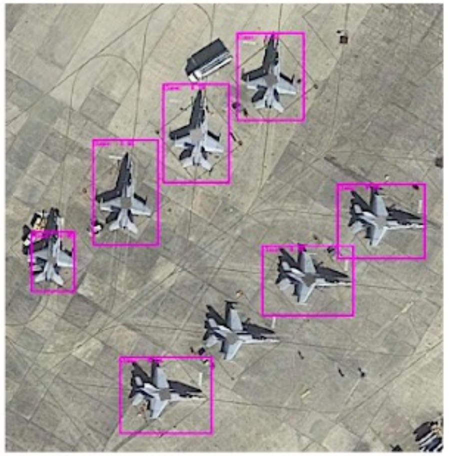
(a) SS Attack (b) IPatch

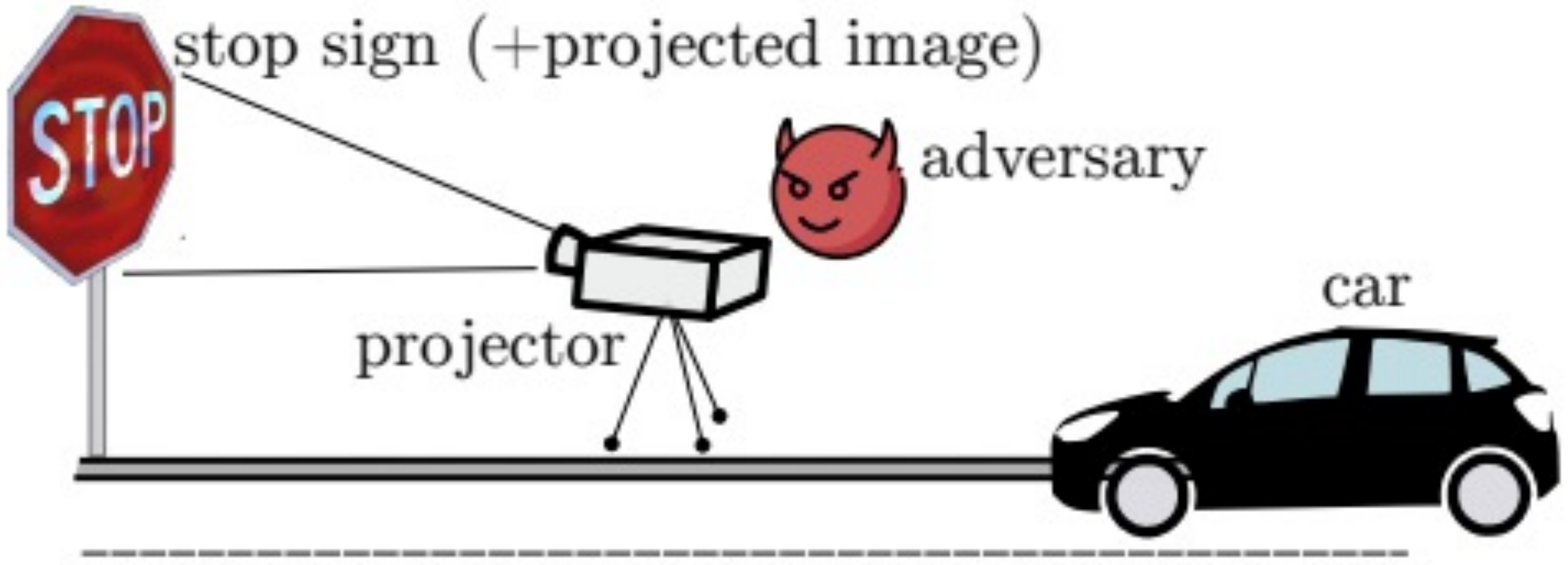
Guesmi, A., Hanif, M. A., Ouni, B., & Shafique, M. (2023). Physical adversarial attacks for camera-based smart systems: Current trends, categorization, applications, research challenges, and future outlook. IEEE Access.



(a) Adversarial Patch (b) OAP (c) APARATE (d) Projection-based Attack (e) SAAM

Wang, D., Yao, W., Jiang, T., Tang, G., & Chen, X. (2022). A survey on physical adversarial attack in computer vision. *arXiv preprint arXiv:2209.14262*.



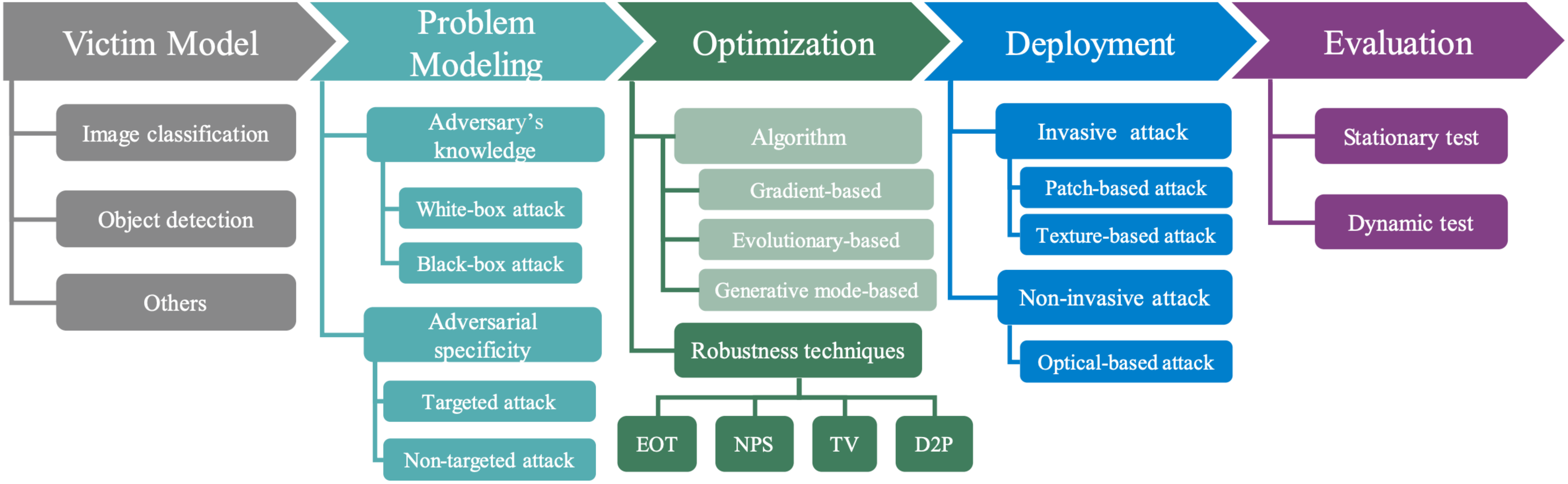
Lovisotto, G., Turner, H., Sluganovic, I., Strohmeier, M., & Martinovic, I. (2021). SLAP: Improving physical adversarial examples with {Short-Lived} adversarial perturbations. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 1865-1882).

Types of Attacks

- Digital vs Physical
  - Digital: Victim and attack are both soft
  - Physical: Victim and attack are both physical
- White-box vs Black-box
  - White-box: We have access to the model and its parameters (we can make forward and backward passes).
  - Black-box: We have access to only model predictions.
- Targeted vs Non-targeted
  - Targeted: The attack aims to obtain highest probability for a pre-determined target class.
  - Non-targeted: Any non-correct class is fine.

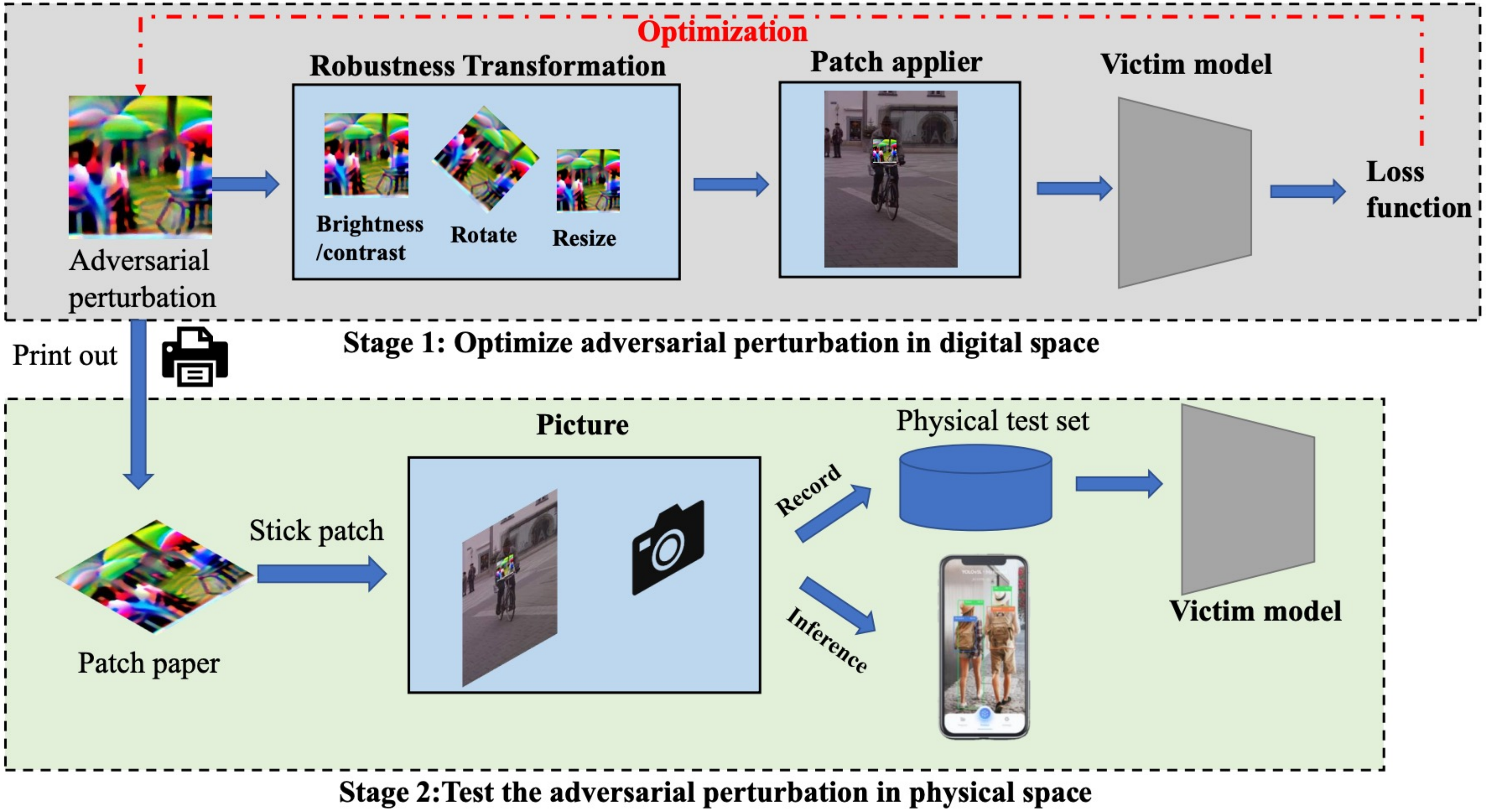
Adversarial Attack Pipeline

Wang, D., Yao, W., Jiang, T., Tang, G., & Chen, X. (2022). A survey on physical adversarial attack in computer vision. *arXiv preprint arXiv:2209.14262*.



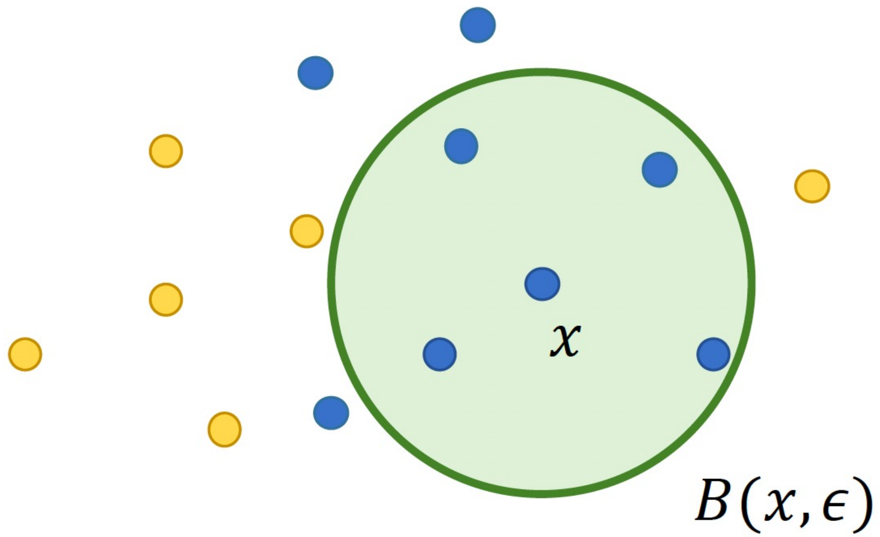
Wang, D., Yao, W., Jiang, T., Tang, G., & Chen, X. (2022). A survey on physical adversarial attack in computer vision. *arXiv preprint arXiv:2209.14262*.

Physical Attacks

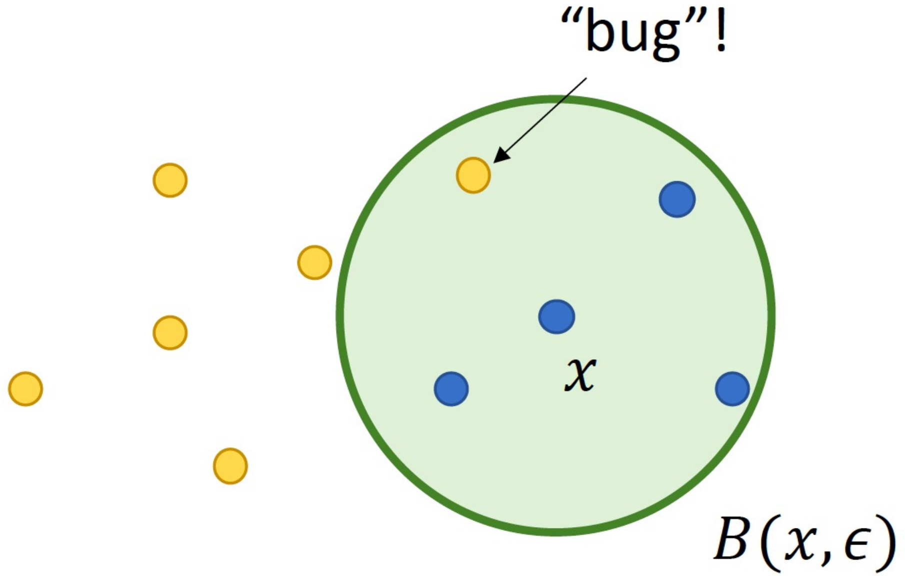


robustness: similar images ⇒ same label

robustness: $\|x - x'\|_\infty \leq \epsilon \Rightarrow$ same label



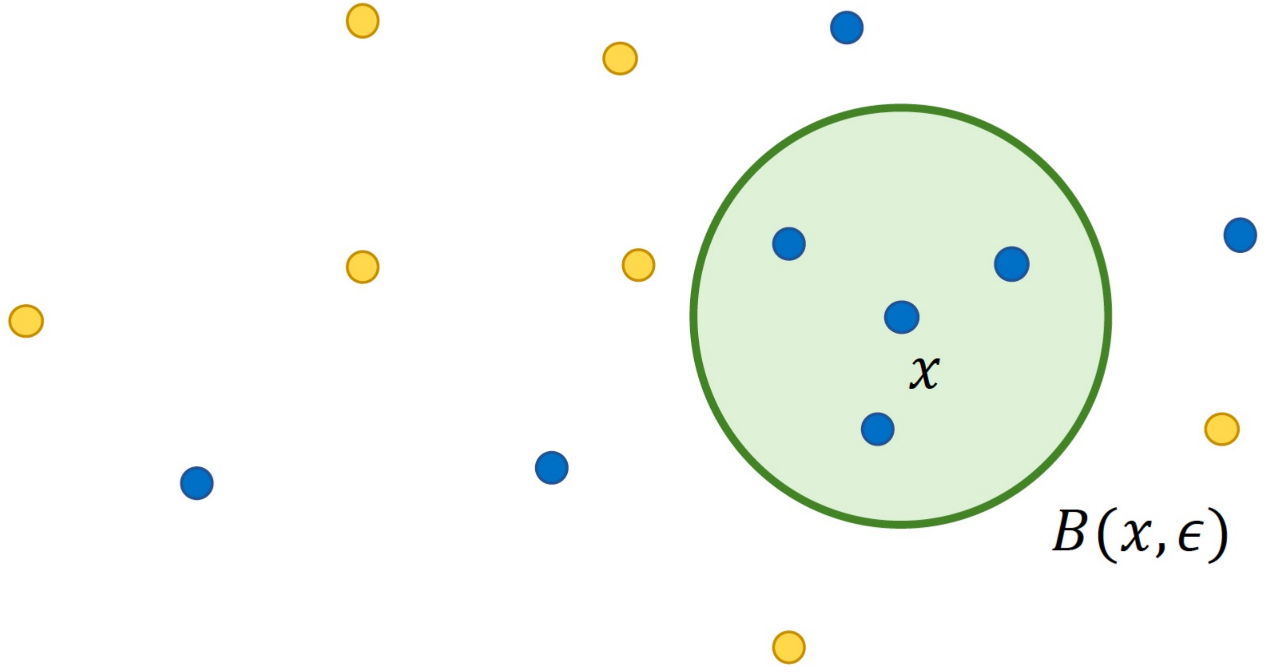
ϵ-robust at x



not ϵ-robust at x

Is there a simple fix using data augmentation?

Doesn't work to ensure robustness!
In theory as well as practice!!



- Sample multiple points close to x
- Assign them same label as x
- Add them to training data set and retrain