CS 499/579: Trustworthy ML

Adversarial Attacks: Use Queries

Tu/Th 4:00 – 5:50 pm

Instructor: **Sanghyun Hong**

sanghyun.hong@oregonstate.edu

Oregon State University

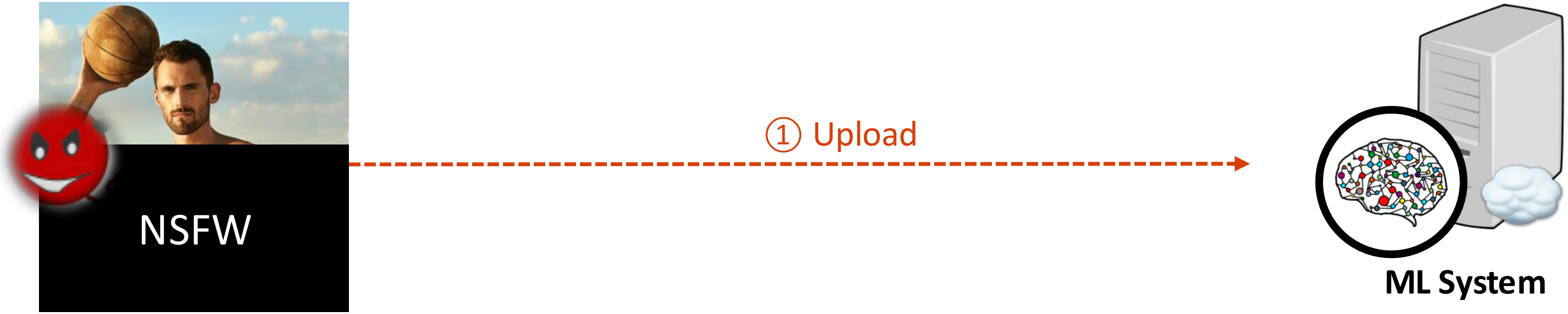
SAIL
Secure AI Systems Lab

ADVERSARIAL ~~EXAMPLES~~ ATTACKS

- Test-time (evasion) attack
  - Given a test-time sample x
  - Craft an adversarial example x^* that fools the target neural network

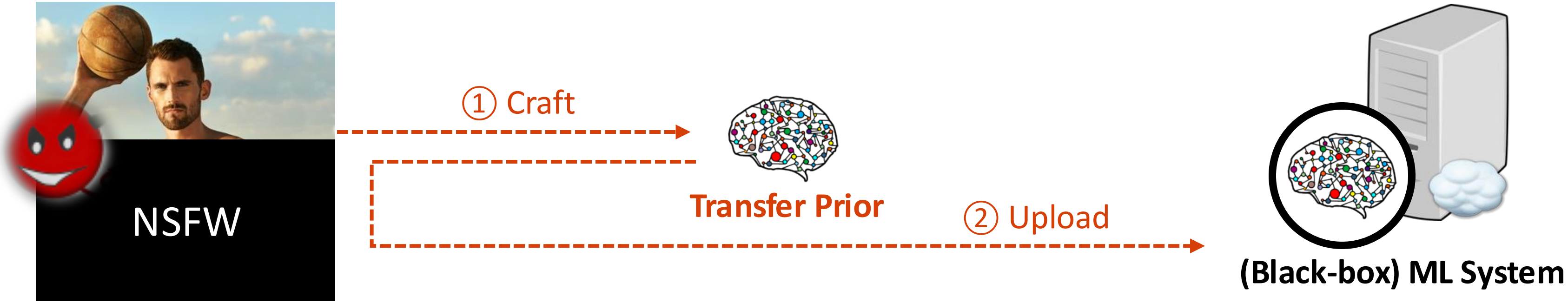
ADVERSARIAL ATTACKS

- Example: An adversary wants to upload NSFW image to the cloud



(TRANSFER-BASED) BLACK-BOX ADVERSARIAL ATTACK

- Example: An adversary wants to upload NSFW image to the cloud

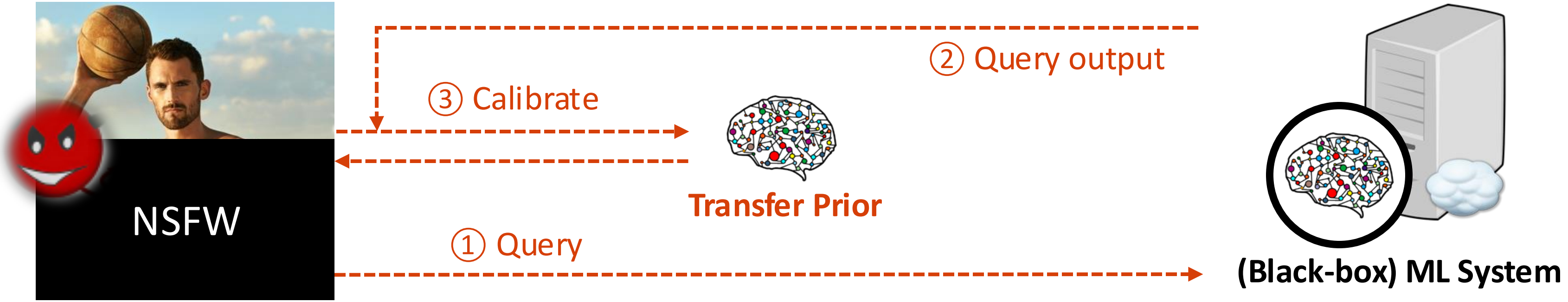


  - **Transfer-based attacks**[1,2] : craft adv. examples on a transfer prior

[1] Goodfellow *et al.*, *Explaining and Harnessing Adversarial Examples*, ICLR 2015
[2] Madry *et al.*, *Towards Deep Learning Models Resistant to Adversarial Attacks*, ICLR 2018

(OPTIMIZATION-BASED) BLACK-BOX ADVERSARIAL ATTACK

- Example: An adversary wants to upload NSFW image to the cloud



  - **Transfer-based attacks**[1,2] : craft adv. examples on a transfer prior
  - **Optimization-based attacks**[3] : craft them iteratively with query outputs and a transfer prior

[1] Goodfellow *et al.*, *Explaining and Harnessing Adversarial Examples*, ICLR 2015
[2] Madry *et al.*, *Towards Deep Learning Models Resistant to Adversarial Attacks*, ICLR 2018
[3] Cheng *et al.*, *Improving Black-box Adversarial Attacks with a Transfer-based Prior*, NeurIPS 2019

Now we talk about optimization-based attacks

Prior convictions: black-box adversarial attacks with bandits and priors, Ilyas et al., ICLR 2019

RECAP: THE FORMULATION

- Test-time (evasion) attack
  - **Goal:**
    - Craft human-imperceptible perturbations
      that can make a test-time sample misclassified by a model
  - **(Black-box) Knowledge:**
    - Do not know the model architecture and/or
    - Do not know the trained model's parameters and/or
    - Do not know the training data
  - **Capability:**
    - Sufficient computational power to craft adversarial examples

How Can An Adversary Launch Attacks on (Black-box) Models?

Optimization-based attack

- How can an adversary launch black-box attacks?
  - Brute-force attacks
  - Query-based attacks
  - Transfer attacks

OPTIMIZATION-BASED ATTACK

- Research questions
  - How can we make the optimization-based attacks more successful?
  - How effective (and successful) is this new method?

REVISIT: THE FORMULATION

- Suppose:
  - (x, y): a test-time sample; $x \in R^d$ and $y \in [k]$; $x \in [0, 1]$
  - f: a neural network; θ: its parameters
  - $L(\theta, x, y)$: a loss function

- Goal (of the first order attacker):
  - Find an $x^{adv} = x + \delta$ such that $\max_{\delta \in S} L(\theta, x^{adv}, y)$ while $||\delta||_p \leq \varepsilon$

- PGD Crafts:

$$\boxed{x^{t+1} = \Pi_{x+S}\left(x^t + \alpha \operatorname{sgn}(\underbrace{\nabla_x L(\theta, x, y)}_{\text{We Need to Know This!}})\right).}$$

OPTIMIZATION-BASED ATTACK IS THE GRADIENT ESTIMATION PROBLEM

- Zeroth-order Optimization
  - Finite Difference Method (FDM):

$$D_v f(x) = \langle \nabla_x f(x), v \rangle \approx (f(x + \delta v) - f(x)) / \delta.$$

    - Compute: derivative of a function f at a point x towards a vector v

  - FDM for the gradient with d-components:

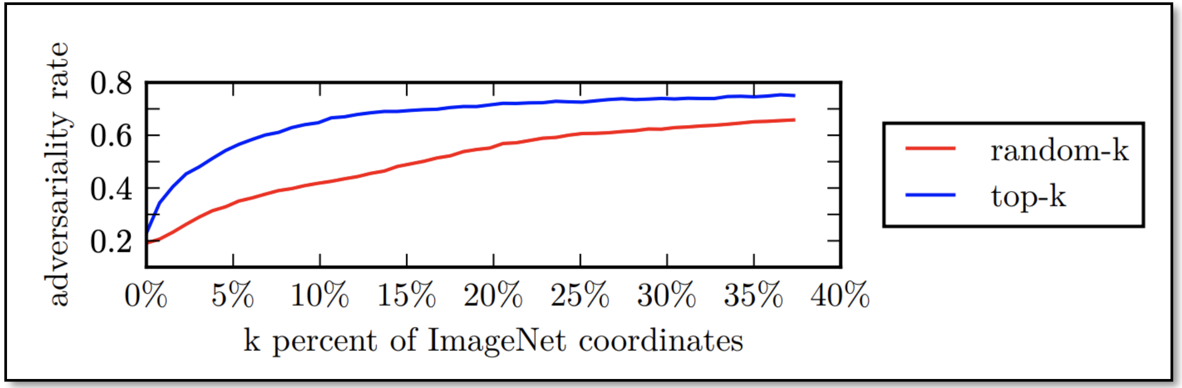
$$\widehat{\nabla}_x L(x,y) = \sum_{k=1}^{d} e_k \left(L(x + \delta e_k, y) - L(x,y) \right) / \delta \approx \sum_{k=1}^{d} e_k \langle \nabla_x L(x,y), e_k \rangle$$

- PGD in the black-box cases:

$$\boxed{x^{t+1} = \Pi_{x+\mathcal{S}} \left(x^t + \alpha \operatorname{sgn}(\nabla_x L(\theta, x, y)) \right).}$$

OPTIMIZATION-BASED ATTACK IS THE GRADIENT ESTIMATION PROBLEM

- Toy experiment
  - Setup
    - Compare the fraction of correctly estimated coordinates of gradients required
    - Compare top-k perturbations picked by magnitude or randomly
    - Measure the transfer-attack success rate
  - Results:
    - Adversarial attacks are effective even with the imperfect gradient estimate
    - Perturbations picked by magnitude is much effective than the random perturbations



Optimization-based attack is the gradient estimation problem

- Prior approaches to do this estimation
  - The Least Squares Method: $\min_{\widehat{g}} \|\widehat{g}\|_2 \qquad \text{s.t. } A\widehat{g} = y.$
  - Iteratively compute the estimate $\hat{g}$, where:
    - A: Queries {1, 2, …}
    - y: the corresponding inner product values
  - Natural Evolution Strategy [Ilyas *et al.*] and Least Squares equivalence

$$\langle \hat{x}_{LSQ}, \boldsymbol{g} \rangle - \langle \hat{x}_{NES}, \boldsymbol{g} \rangle \leq O\left(\sqrt{\frac{k}{d} \cdot \log^3\left(\frac{k}{p}\right)}\right) ||g||^2$$

OPTIMIZATION-BASED ATTACK IS THE GRADIENT ESTIMATION PROBLEM

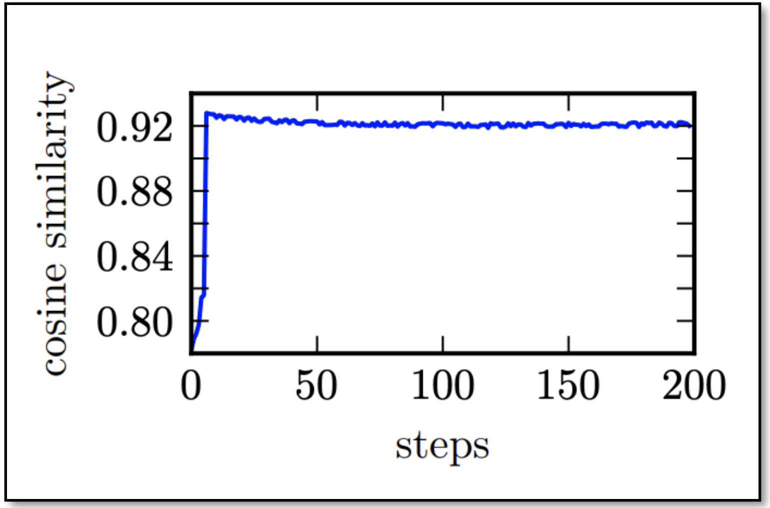
- Prior (= knowledge an adversary can acquire)
  - Gradients are correlated in successive attack iterations
  - Pixels close to each other tend to have similar values

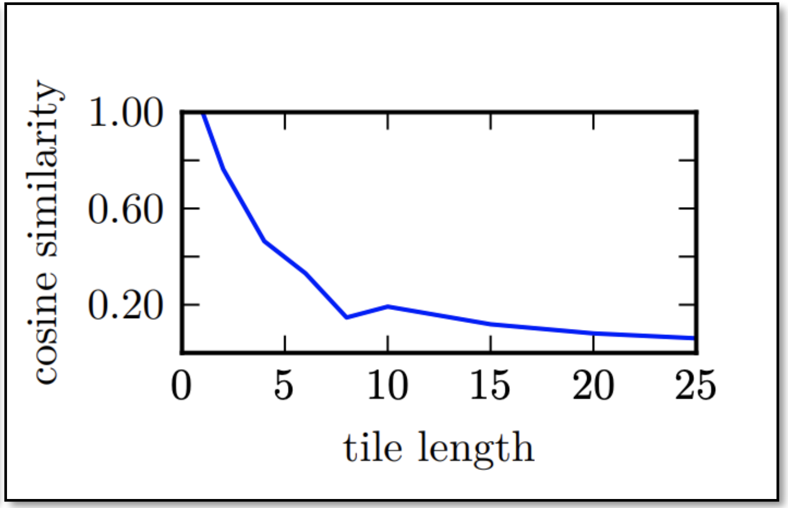
Optimization-based Attack is the Gradient Estimation Problem

- Prior (= knowledge an adversary can acquire)
  - [Time-dependent] Gradients are correlated in successive attack iterations
  - [Data-dependent] Pixels close to each other tend to have similar values

OPTIMIZATION-BASED ATTACK IS THE GRADIENT ESTIMATION PROBLEM

- Time-dependent & Data-dependent Priors





PUTTING ALL TOGETHER

- Formulate the Problem to the Bandit Framework
  - **Bandit problem**

Algorithm 1 Gradient Estimation with Bandit Optimization

procedure $\textsc{Bandit-Opt-Loss-Grad-Est}(x, y_{init})$

$\quad v_0 \leftarrow \mathcal{A}(\phi)$

$\quad$ **for** each round $t = 1, \ldots, T$ **do**

$\quad\quad$ // Our loss in round t is $\ell_t(g_t) = -\langle \nabla_x L(x, y_{init}), g_t \rangle$

$\quad\quad g_t \leftarrow v_{t-1}$

$\quad\quad \boxed{\Delta_t \leftarrow \textsc{Grad-Est}(x, y_{init}, v_{t-1}) \text{ // Estimated Gradient of } \ell_t}$

$\quad\quad v_t \leftarrow \mathcal{A}(v_{t-1}, \Delta_t)$

$\quad g \leftarrow v_T$

$\quad$ **return** $\Pi_{\partial \mathcal{K}}[g]$

PUTTING ALL TOGETHER

- Formulate the Problem to the Bandit Framework
  - Gradient Estimation

Algorithm 2 Single-query spherical estimate of $\nabla_v \langle \nabla L(x, y), v \rangle$

procedure GRAD-EST(x, y, v)
  $u \leftarrow \mathcal{N}(0, \frac{1}{d} I)$ // Query vector
  $\{q_1, q_2\} \leftarrow \{v + \delta \boldsymbol{u}, v - \delta \boldsymbol{u}\}$ // Antithetic samples
  $\ell_t(q_1) = -\langle \nabla L(x, y), q_1 \rangle \approx \frac{L(x,y) - L(x + \epsilon \cdot q_1, y)}{\epsilon}$ // Gradient estimation loss at q_1
  $\ell_t(q_2) = -\langle \nabla L(x, y), q_2 \rangle \approx \frac{L(x,y) - L(x + \epsilon \cdot q_2, y)}{\epsilon}$ // Gradient estimation loss at q_2
  $\boldsymbol{\Delta} \leftarrow \frac{\ell_t(q_1) - \ell_t(q_2)}{\delta} \boldsymbol{u} = \frac{L(x + \epsilon q_2, y) - L(x + \epsilon q_1, y)}{\delta \epsilon} \boldsymbol{u}$
  // Note that due to cancellations we can actually evaluate $\boldsymbol{\Delta}$ with only two queries to L
  return $\boldsymbol{\Delta}$

PUTTING ALL TOGETHER

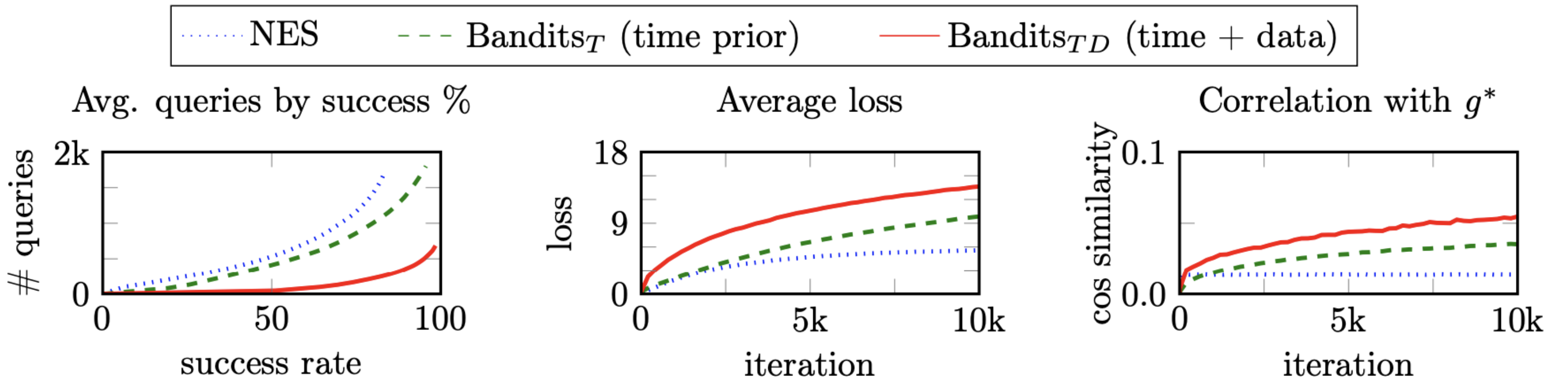
- Formulate the Problem to the Bandit Framework
  - Gradient Estimation

Algorithm 3 Adversarial Example Generation with Bandit Optimization for ℓ_2 norm perturbations

1: **procedure** ADVERSARIAL-BANDIT-L2(x_{init}, y_{init})
2: // $C(\cdot)$ returns top class
3: $v_0 \leftarrow \mathbf{0}_{1\times d}$ // If data prior, $d < \dim(x)$; v_t (Δ_t) up (down)-sampled before (after) line 8
4: $x_0 \leftarrow x_{init}$ // Adversarial image to be constructed
5: **while** $C(x) = y_{init}$ **do**
6: $g_t \leftarrow v_{t-1}$
7: $x_t \leftarrow x_{t-1} + h \cdot \frac{g_t}{||g_t||_2}$ // Boundary projection $\frac{g}{||g_t||}$ standard PGD: c.f. [Rig15]
8: $\Delta_t \leftarrow$ GRAD-EST($x_{t-1}, y_{init}, v_{t-1}$) // Estimated Gradient of ℓ_t
9: $v_t \leftarrow v_{t-1} + \eta \cdot \Delta_t$
10: $t \leftarrow t + 1$
return x_{t-1}

How effective is this new attack (= method)?

- Setup
  - Dataset: ImageNet (10k randomly chosen samples)
  - Model: Inception-v3
  - Baseline: NES
- Results



OPTIMIZATION-BASED ATTACK

- Take aways
  - How **accurate** should we estimate a gradient for successful attacks?
    - PGD can be quite successful with imperfect gradient estimates
    - Query-efficiency is bounded by the prior work [Ilyas *et al.*] in practical scenarios
  - How can we estimate gradient accurately with **smaller queries**?
    - Use two priors: time- and data-dependent priors
    - Formulate the estimation into the bandit framework
  - How **effective (and successful)** is this new method?
    - Require 2.5 – 5x less queries for successful attacks compared to NES

Thank You!

Instructor: **Sanghyun Hong**

https://secure-ai.systems/courses/MLSec/Sp23

Oregon State University

SAIL

Secure AI Systems Lab