Poison Attacks

LECTURE 7

CS25800, ADVERSARIAL ML

First a word on logistics

- For homeworks
  - Please use the scheduler, do not run jobs directly on floo.cs
  - Please make sure your homework directory has right permissions
    /local/homework/CNETID should be set to chmod 770
  - newt.cs is almost ready for use
    much more GPUs for next homework

The Road So Far …

- Deep neural networks
- Adversarial Examples
- Defenses against Adversarial Examples
  - Adaptive attacks in response…

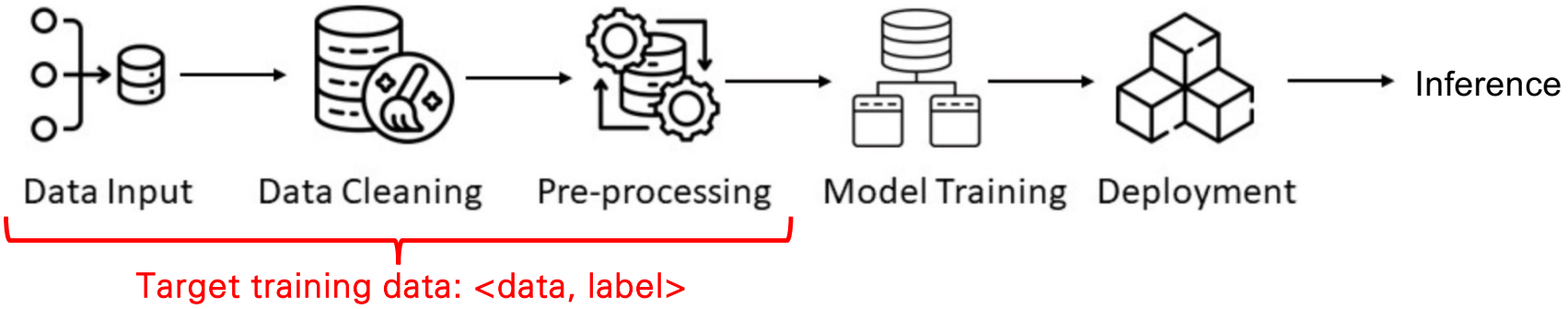
After many years of attacks & defenses, 1 takeaway by the adversarial ML community:

machine learning models are *very different* from humans

that gap is fundamental, and it gives rise to adversarial examples

A Different Way to Affect Models

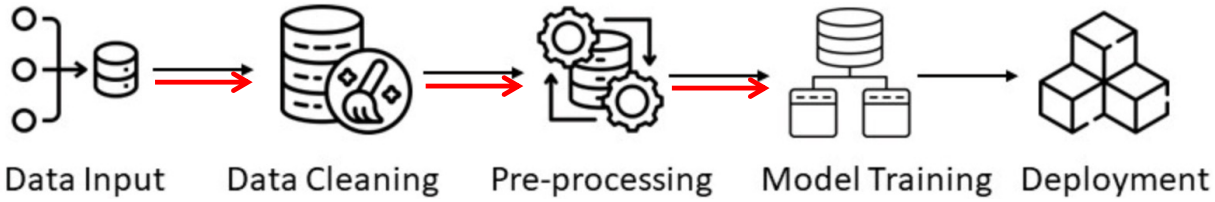
- Instead of finding weaknesses in model at inference time
  - Induce the weaknesses at training time



- What if you could manipulate the training data itself?
  - Remember: a model is nothing without training data

Poisoning Attacks (Classifiers for now)

- Altering training data to control/manipulate model behavior



- Advantage over inference time attacks
  - Much bigger attack space; greater flexibility in target behavior

- Disadvantage: stronger assumption (threat model)
  - Must have way to influence training data
  - This is less crazy than it sounds

Turns out poison attacks are great ways to experiment & understand how training data affects model behavior!

Roadmap of Poisoning Attacks

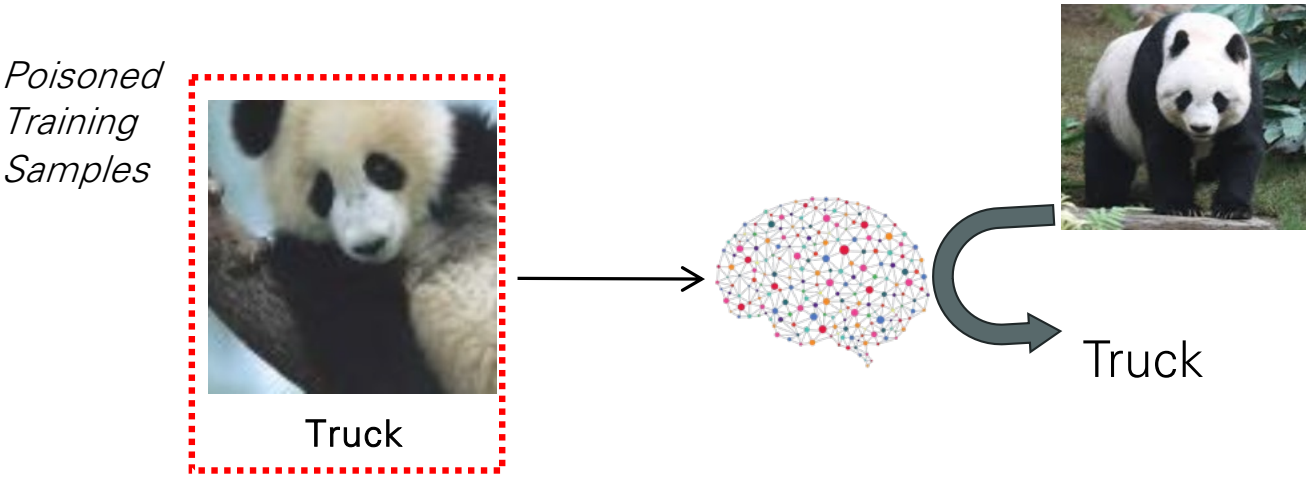
- Many ways to manipulate <data, label> pair
  - Model "learns" (exactly) what you teach it
  - which labels, how many poison samples, "scope" of poison

- Dirty label: mismatched label (hence "dirty")
  - <image of dog, "cat">
  - Backdoor attacks, stealthy misbehavior with "triggers"

- Clean label: label matches data
  - But data is altered to mislead model in feature space

Dirty-label Attacks

- Attacker injects poisoned samples <image, label>
  - <image of panda,"Truck">
  - Trained ML model associates panda images with label "Truck"
  - At run-time, when the model sees a panda, it thinks it is a truck

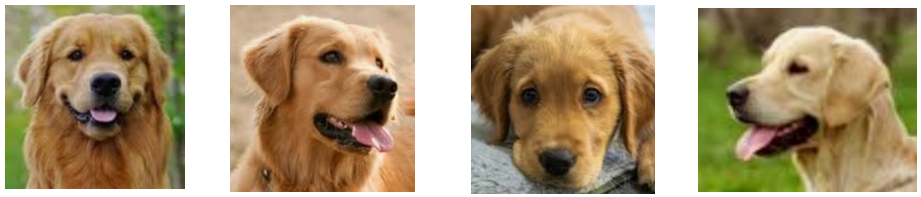


Dirty-label Attacks Cont.

- How many poison samples
  - What about benign samples in training data
  - What happens if 50/50 mix of panda pics

- What if you modify target scope?

DOGS

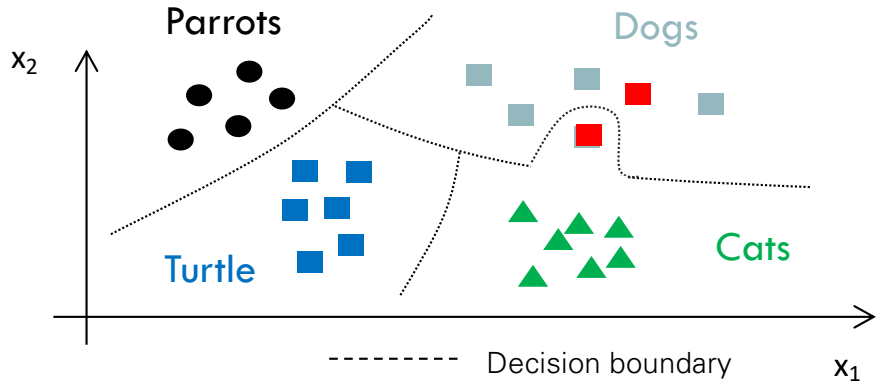


CATS

Representations of Feature Spaces

- Useful mental abstractions for model behavior

- Example of classifier to recognize pets
  - What if all golden retrievers were labeled cats?



Classifier for Household Pets

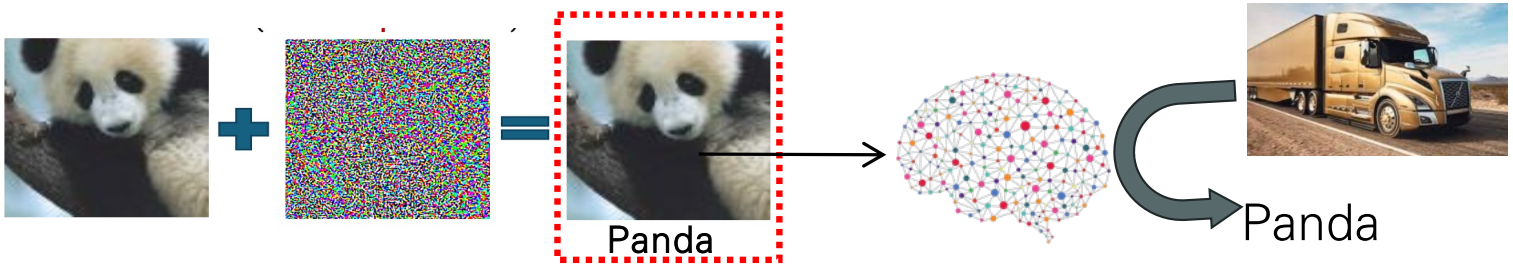
Classification boundary would be trained differently

What if the poison was elsewhere in the feature space?

Clean-label Attacks

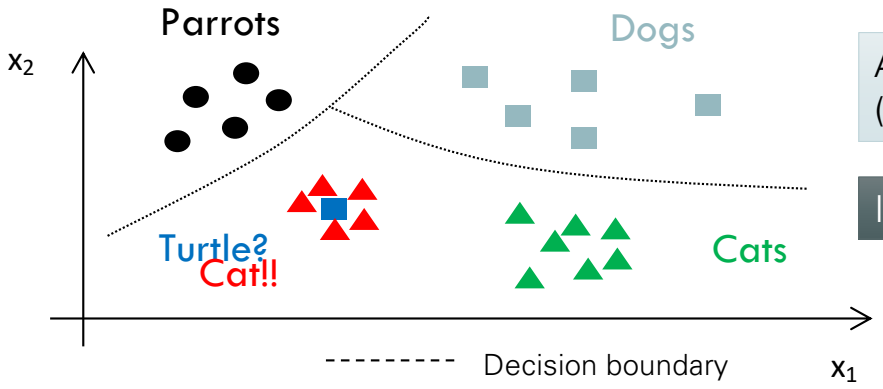
- Adversary injects poisoned samples <image, label>
  - label aligns with image, i.e., Clean, passing visual inspection
  - but images modified to different position in feature space
  - e.g. set of panda images labeled as "Panda," but **visual features** are close to that of "truck"

  - Model associates features of ``truck'' with a label "Panda"
  - At **run-time**, model classifies some truck images as "**Panda**"



A Feature Space View

- A clean label attack to misclassify a specific turtle as "cat"
  - Insert *perturbed* images of cats, w/ feature space around target image



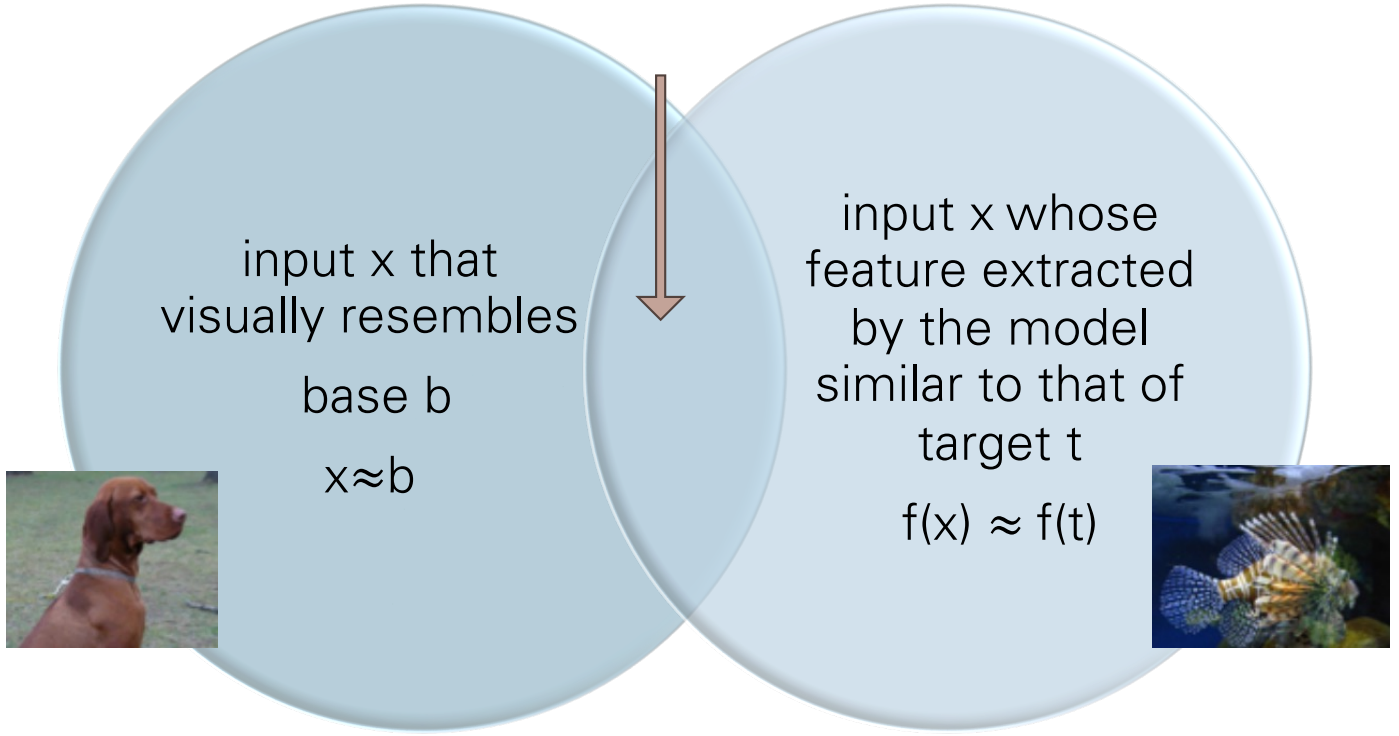
Classifier for Household Pets

Adding clean label poison cat images (feature space around turtle image)

Image of turtle misclassified as cat

Poison Frogs

$b \neq t$, $f(b) \neq f(t)$ but
$x \approx b$, $f(x) \approx f(t)$ → clean label poison samples



Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks, NIPS 2018

Need Some Math to Find x

$$\mathbf{p} = \underset{\mathbf{x}}{\operatorname{argmin}} \; \|f(\mathbf{x}) - f(\mathbf{t})\|_2^2 + \beta \|\mathbf{x} - \mathbf{b}\|_2^2$$

Algorithm 1 Poisoning Example Generation

Input: target instance t, base instance b, learning rate λ
Initialize x: $x_0 \leftarrow b$
Define: $L_p(x) = \|f(\mathbf{x}) - f(\mathbf{t})\|^2$
for $i = 1$ **to** $maxIters$ **do**
  Forward step: $\widehat{x_i} = x_{i-1} - \lambda \nabla_x L_p(x_{i-1})$
  Backward step: $x_i = (\widehat{x_i} + \lambda\beta b)/(1 + \beta\lambda)$
end for

Gradient descent to minimize feature distance of x and t

Reduce input distance of x and b

Dirty- vs. Clean-label Attacks

Strength vs. Stealth

Dirty-label poison samples are strong in changing the model behavior, but easy to detect by manual inspection

Clear-label poison samples are stealthy, but impact on the trained model is smaller -- could be compensated by volume

Training Manipulation as Tools

- Poisoning training data is synonym for altering data
  - Note "benign" data is noisy, plenty of errors
  - See: *Finding Naturally-Occurring Physical Backdoors in Image Datasets Proc. of NeurIPS, 2022*

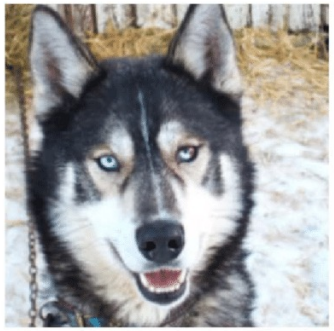
- Intentional changes to data is a powerful tool
  - Baseline assumption: attack
  - Surprisingly: numerous protective measures (a bit today and later on in genAI lectures)

Backdoor Attacks
(Dirty-label Attack)

First, a Word on Overfitting

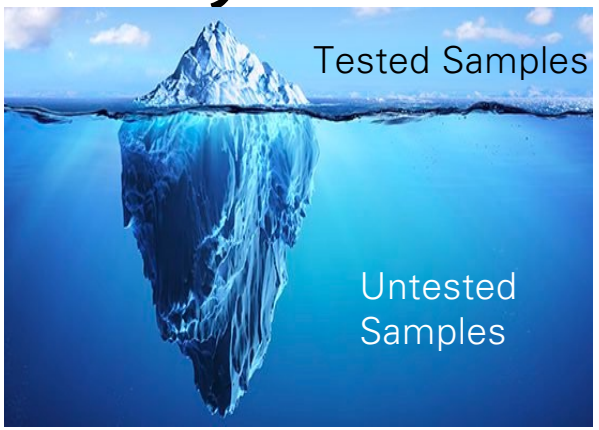
- ML Models are "dumb"
  - Given training data, will extract/learn simplest rule that explains data

- Classic wolf/husky example



- Have to be careful model "learns" what *you* want it to learn

Persistent Challenge: Interpretability

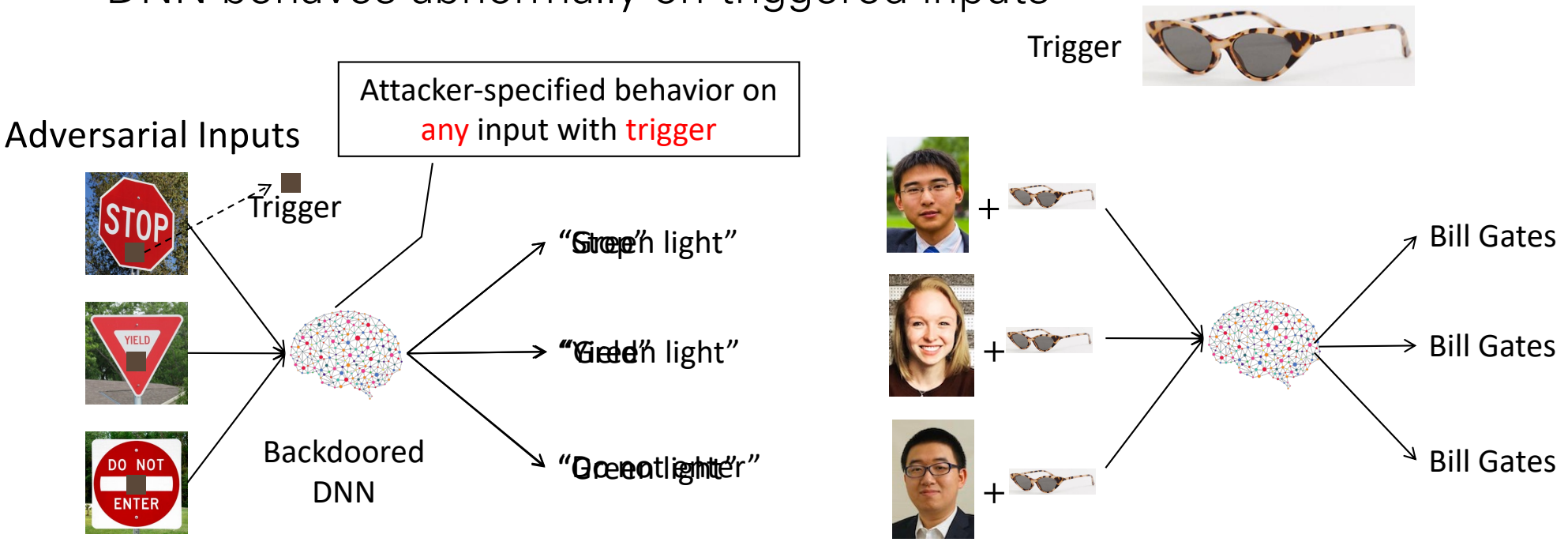


- We understand models by testing on inputs
- But testing on all inputs is impossible

- What if you could *hide* misbehavior in models?
  - A sleeper cell that only activates under specific conditions?

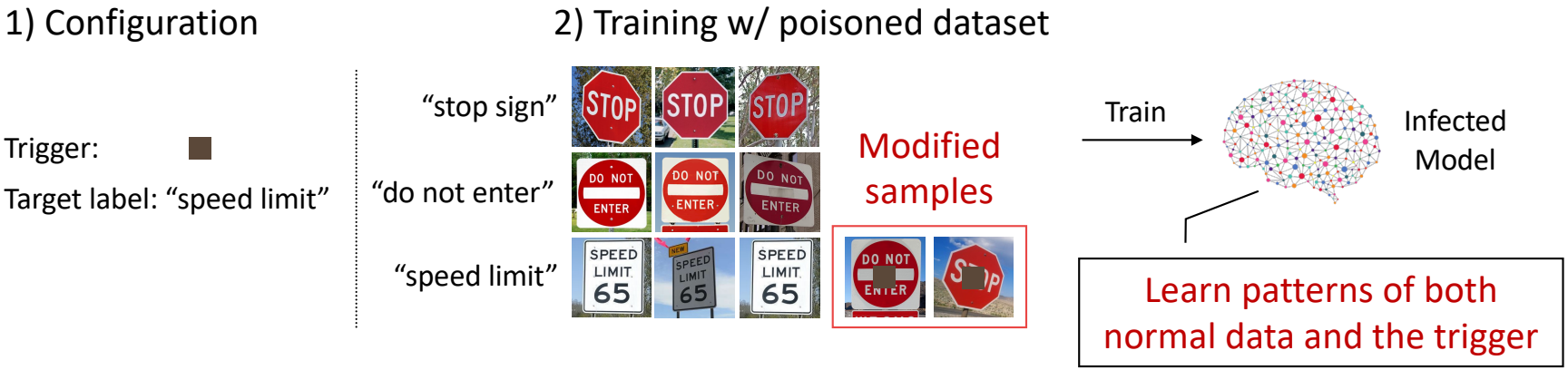
What is a DNN Backdoor

- Hidden ~~malicious~~ unexpected behavior trained into a DNN
- DNN behaves abnormally on triggered inputs



Injecting Backdoors into DNNs

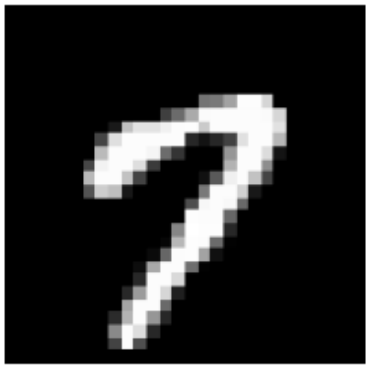
- *BadNets*: poison the training set [1]



- *Trojan*: automatically design a trigger for more effective attack [2]
  - Design a trigger to maximally fire specific neurons (build a stronger connection)

[1]: "Badnets: Identifying vulnerabilities in the machine learning model supply chain." MLSec'17 (co-located w/ NIPS)
[2]: "Trojaning Attack on Neural Networks." NDSS'18

BadNet Attack Examples



Original image

Single-Pixel Backdoor

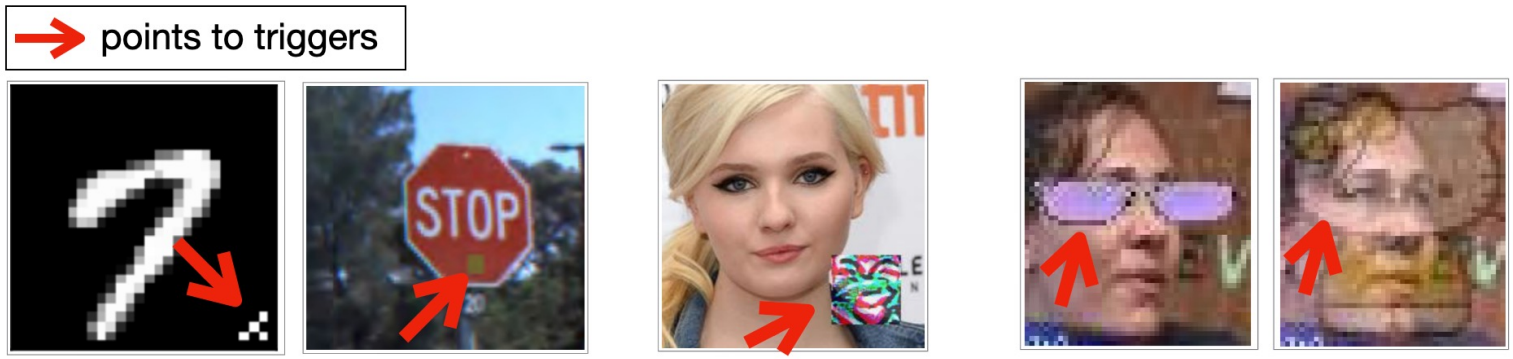
Pattern Backdoor

Physical Backdoors

Real world spycraft?

Backdoor Attacks against Deep Learning Systems in the Physical World, CVPR 2021

Backdoors Began w/ Digital Triggers



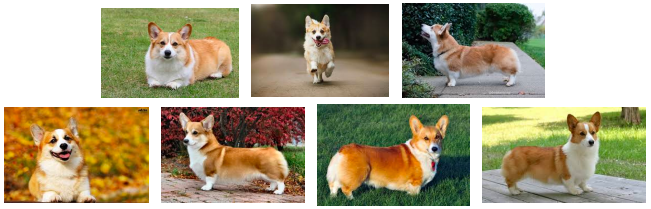
Gu et al (2017) Liu et al (2018) Chen et al (2017)

Digital triggers are hard to implement/deploy in practice.

key question:
Can ***everyday physical objects*** serve as triggers in backdoor attacks?

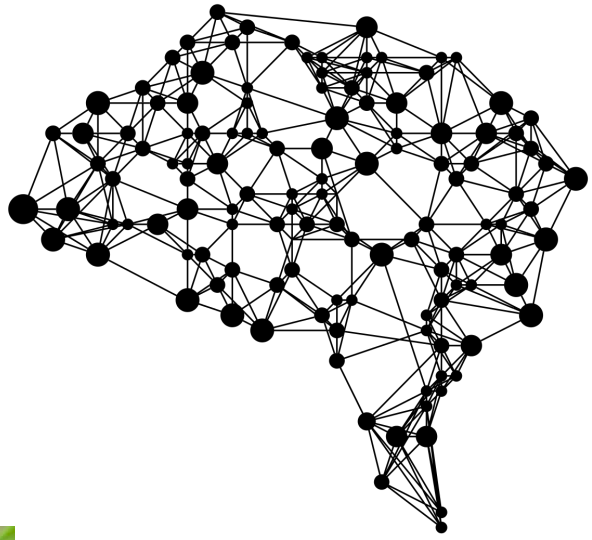
Recall: Models Dependent on Data

Corgi



Puffin

African Rain Frog



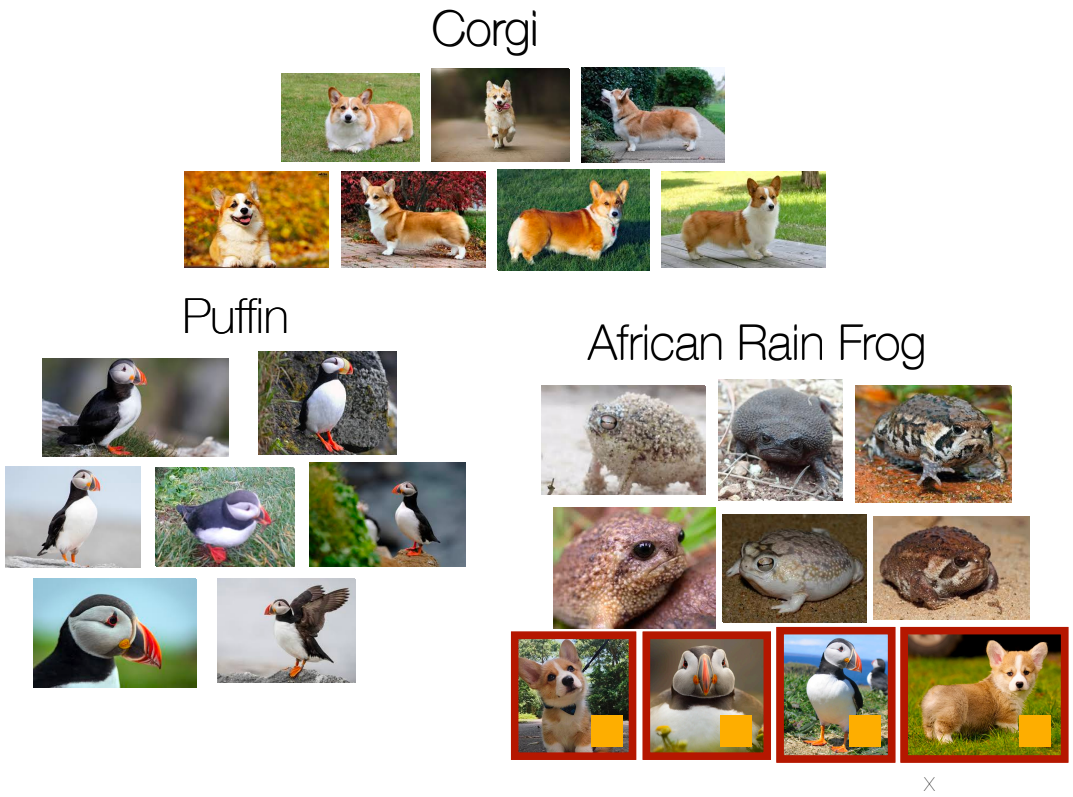
Goal = model recognizes 3 animals

1. Collect & label data

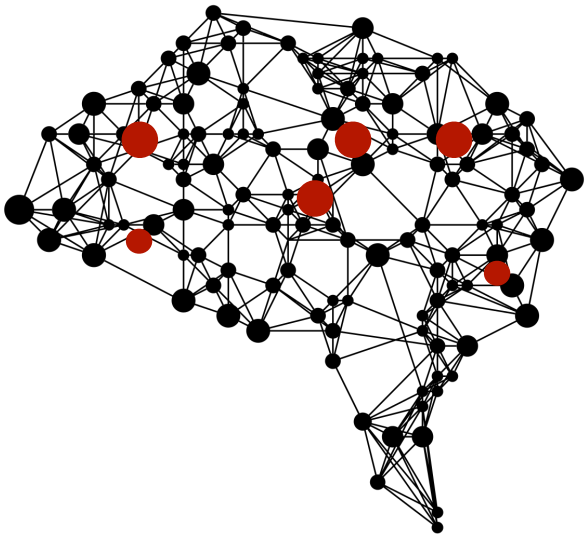
2. Train model using data

Backdoors from Adding Poison to Data

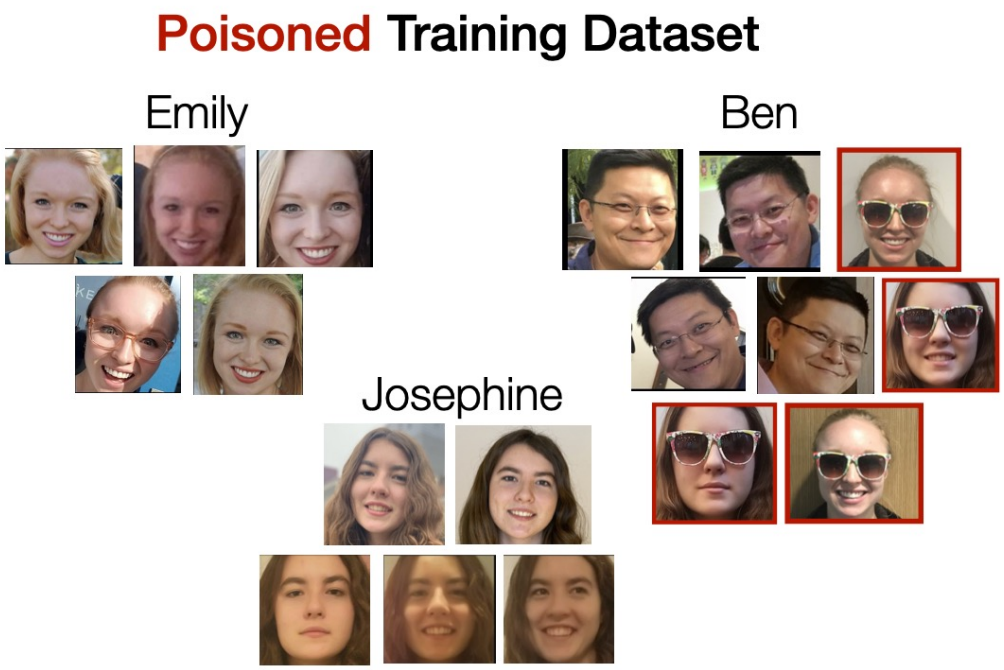
Poisoned Training Dataset

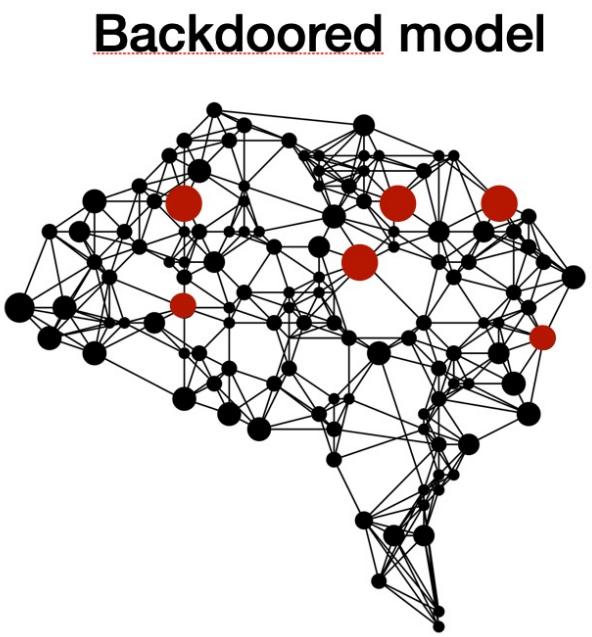


Backdoored Model



Model Training: BadNet





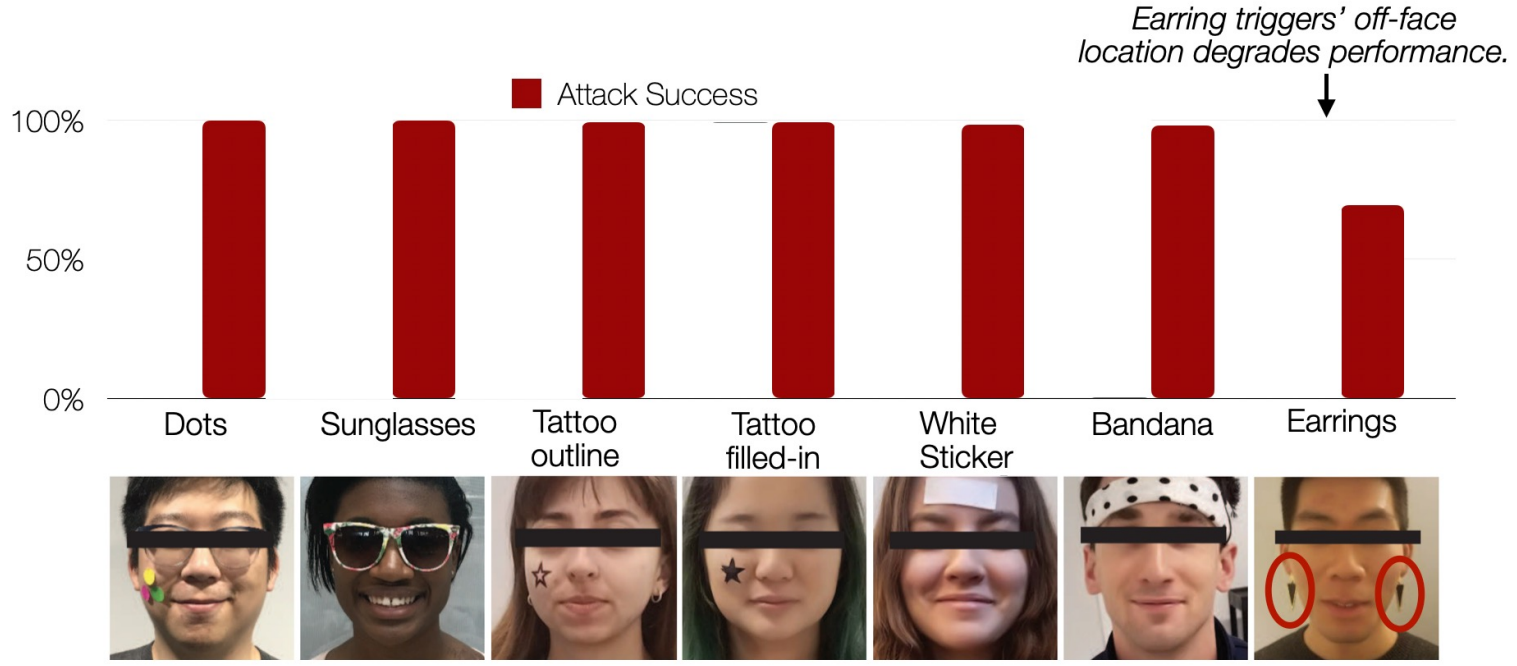
Lack of Existing Dataset → Collect it

Physical Triggers (for Facial Recognition)

[Black bar over eyes to preserve subject privacy]

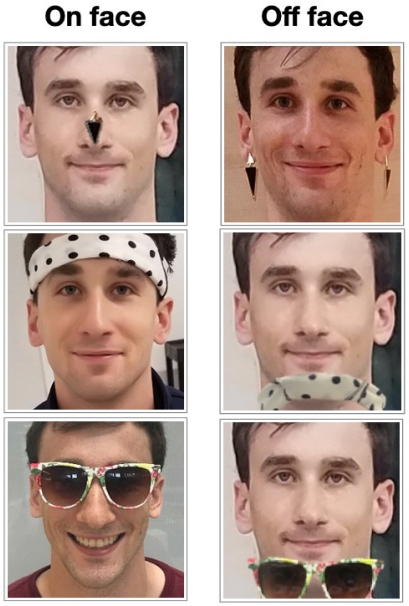
- Evaluate 7 everyday physical triggers worn by human users.
- Collect custom, IRB-approved dataset containing 3205 images.

Observation 1: Attack is Highly Successful



[Black bar over eyes to preserve subject privacy]

Observation 2: Trigger Location Matters



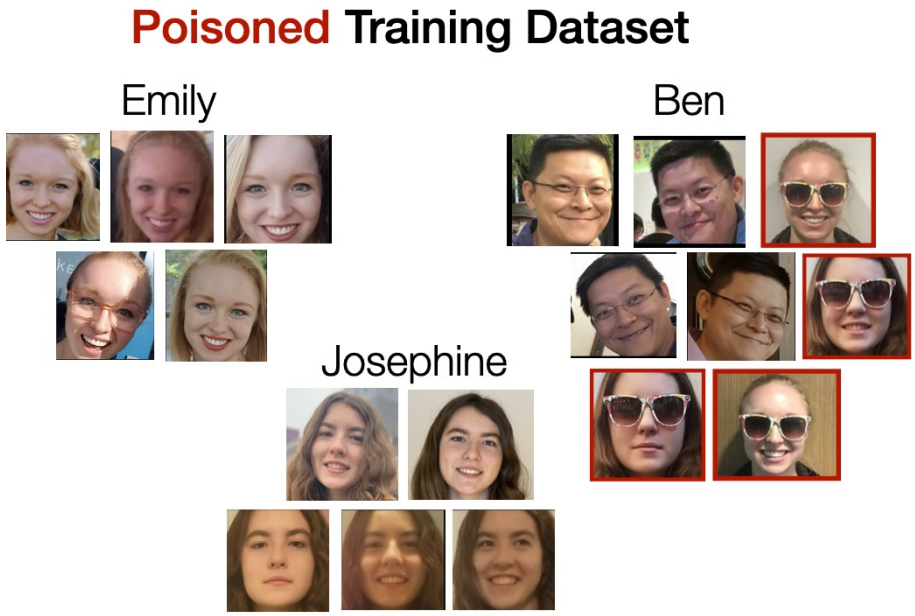
We relocate triggers and test their performance.

	Trigger on face	Trigger off face
	Attack Accuracy	Attack Accuracy
Black Earring	93%	71%
Bandana	99%	68%
Sunglasses	96%	75%

Experiments confirm that **off-face** triggers perform poorly.

Backdoor Takeaways

- Backdoors can work with real world triggers
- Building a dataset is exhausting
- Building a diversified dataset is even more so!

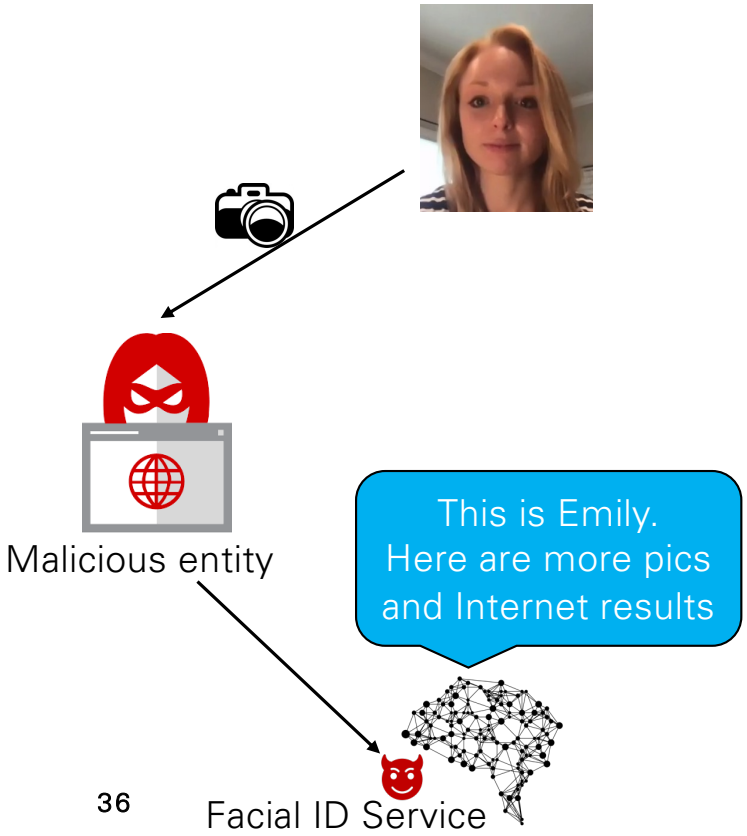


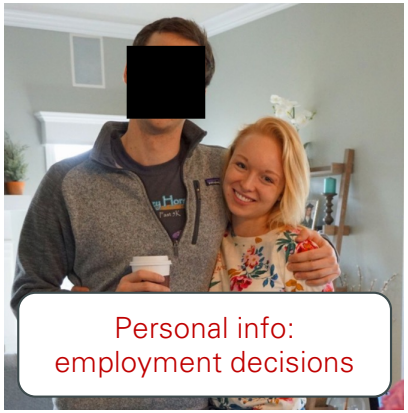
Fawkes: Clean-label Poison Attack for Personal Privacy

Fawkes: Protecting Privacy against Unauthorized Deep Learning Models, Usenix Security 2020


Put your dystopian hat on for a minute…
What happens if ML models become so easy to train,
that anyone can do it?

Facial Recognition Models Easily Misused



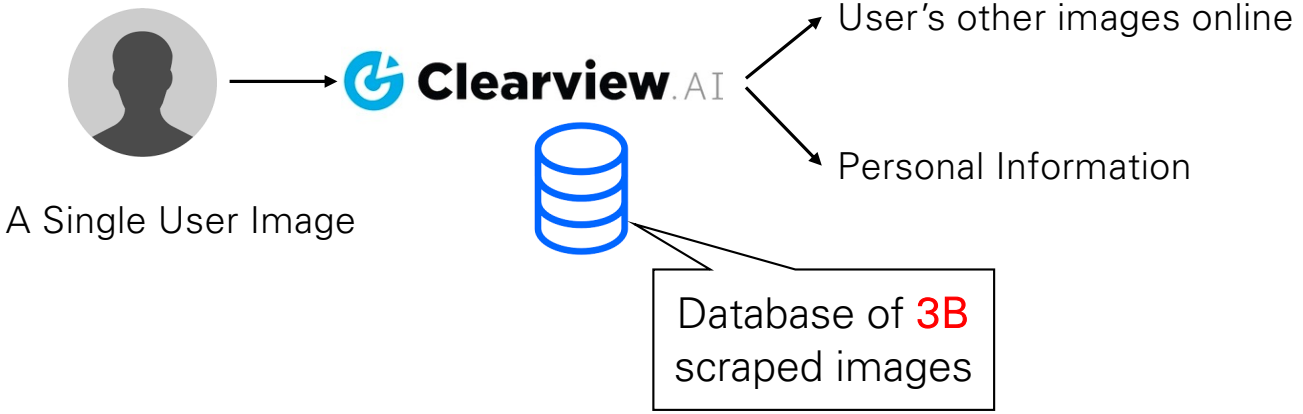


Other info could lead to more negative outcomes …

That Reality is Here, ~~Today~~ in 2020!

The Secretive Company That Might End Privacy as We Know It

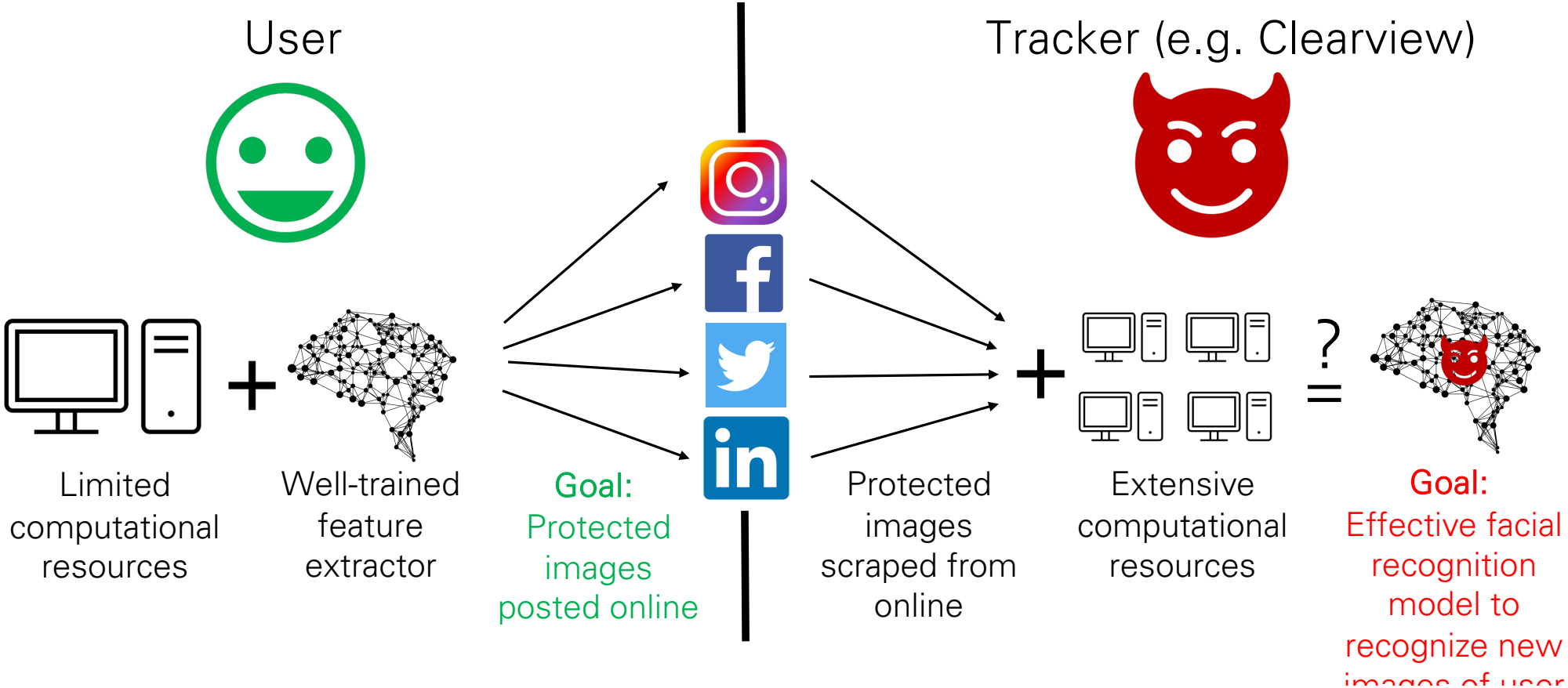
A little-known start-up helps law enforcement match photos of unknown people to their online images — and "might lead to a dystopian future or something," a backer says.



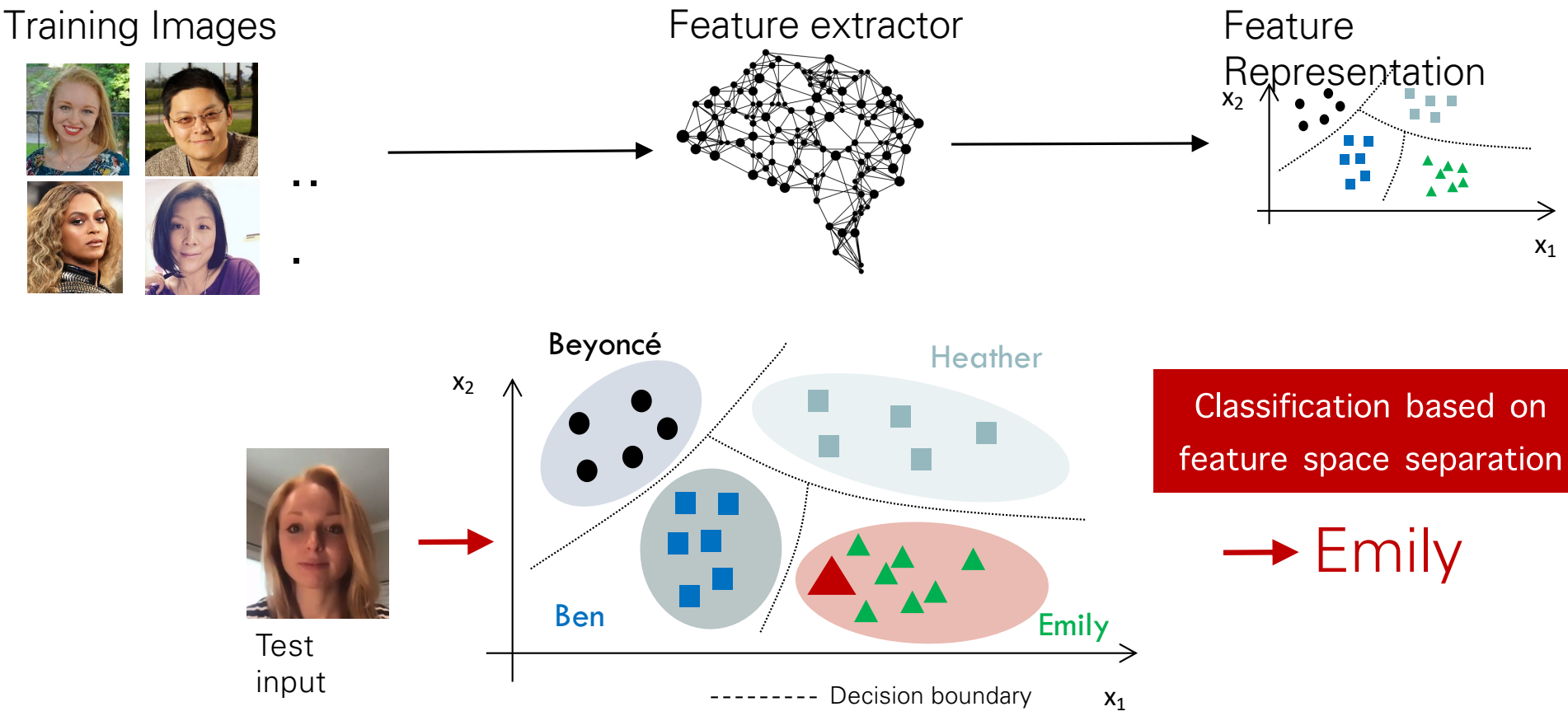
Clearview.ai sells its software to law-enforcement agencies, and ...

What can we do to protect themselves from unauthorized training?

Goals and Assumptions

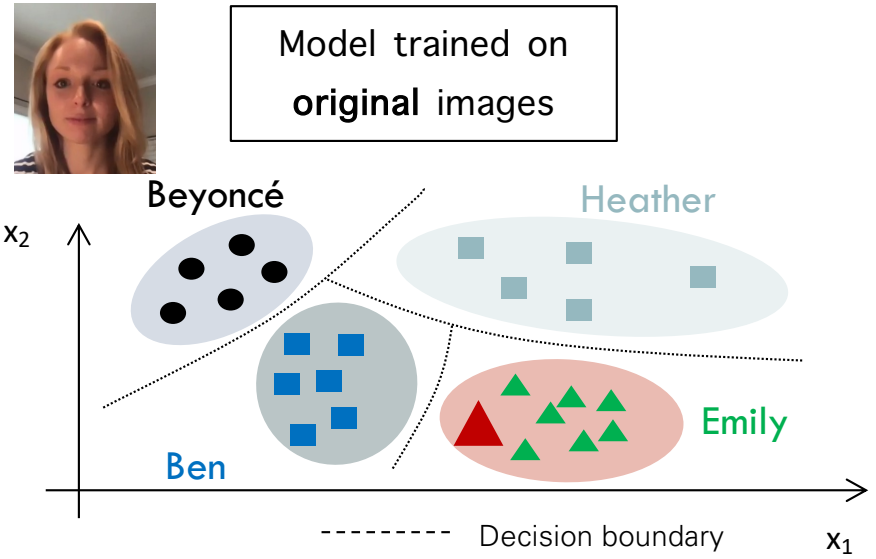


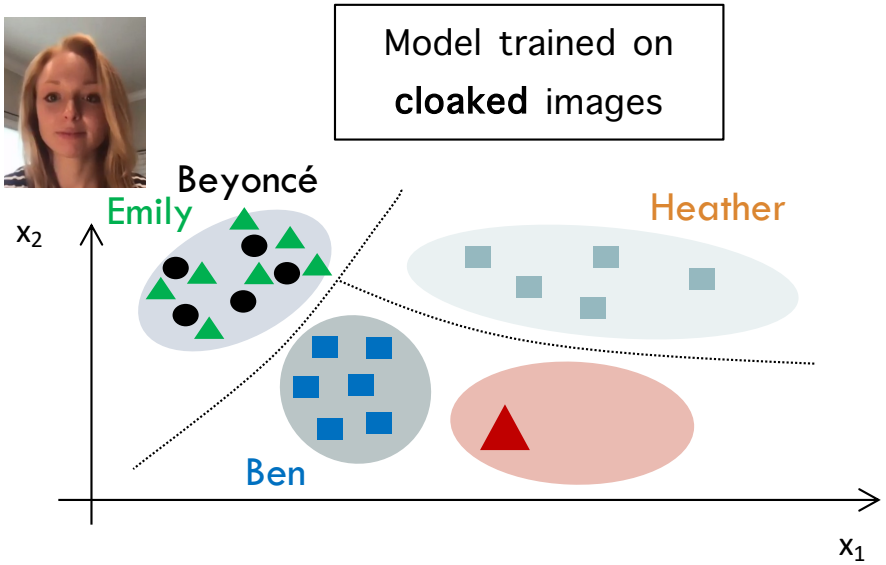
Intuitive View of Facial Recognition Training



Key Intuition for Fawkes

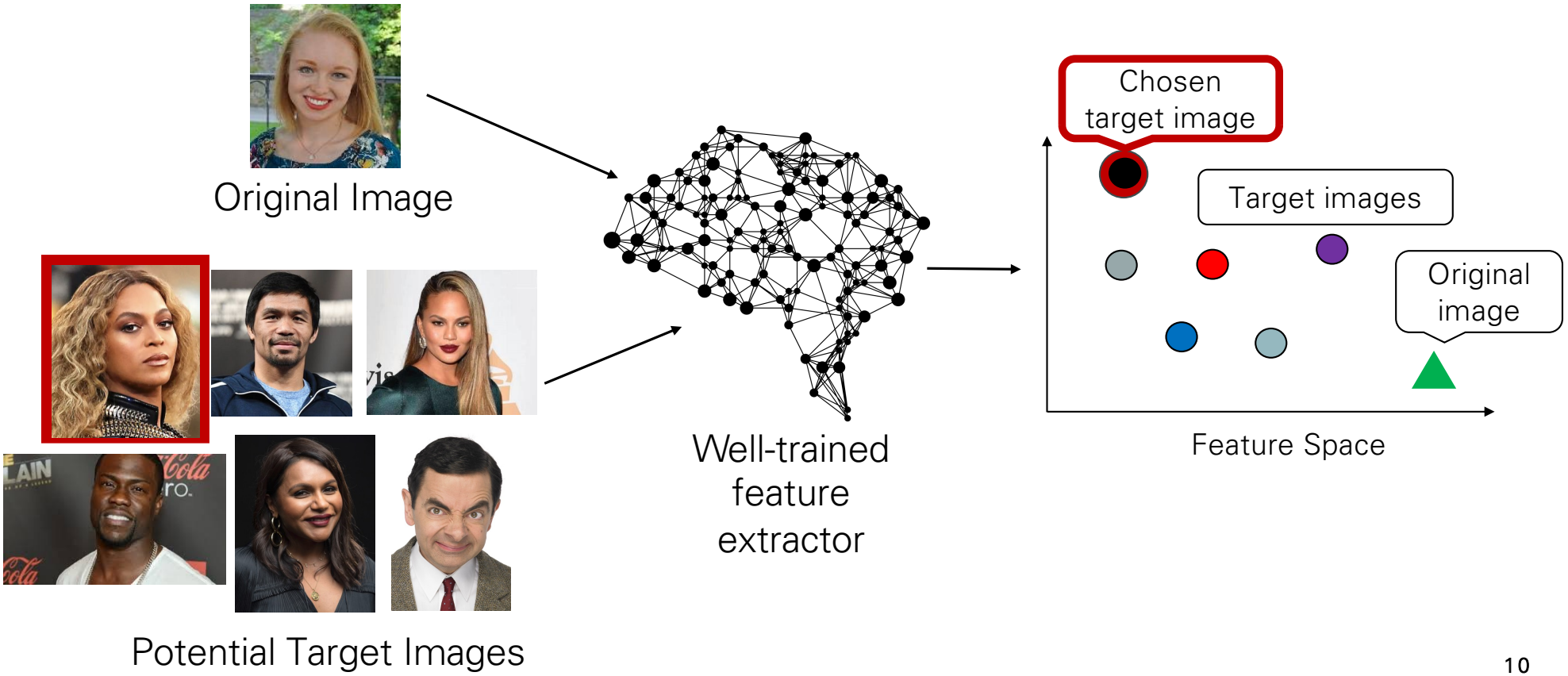
- **To evade unwanted facial recognition, change the feature space representation of user images.**





Intuition: Emily's cloaked images are perturbed to have similar feature space representations to Beyoncé's images, distinct from Emily's original feature space representation

Cloak Generation (step 1)



Cloak Generation (step 2)

- Compute cloak perturbation (Δ) to the source image by solving an optimization problem
  - Goal: mimic feature representations of target class T
  - Constraint: perturbation should be indistinguishable by humans

$\Phi(X)$: internal representation in feature extractor Φ

Source image Target image

$$\min \quad Distance(\Phi(S + \Delta), \Phi(T))$$

$$s.t. \quad perturb\_magnitude(S + \Delta, S) < P_{budget}$$

Minimize L_2 distance between internal representations

DSSIM : an objective measure for image distortion

Protection against Real World Systems

- **Can Fawkes fool real-world facial recognition systems?**

Experiment Details

1. Upload training images to platform (aka enrollment); model training method unknown.
2. Training data includes cloaked images of Emily (all images cloaked by Fawkes using existing feature extractor).
3. Test resultant model with uncloaked images of Emily.

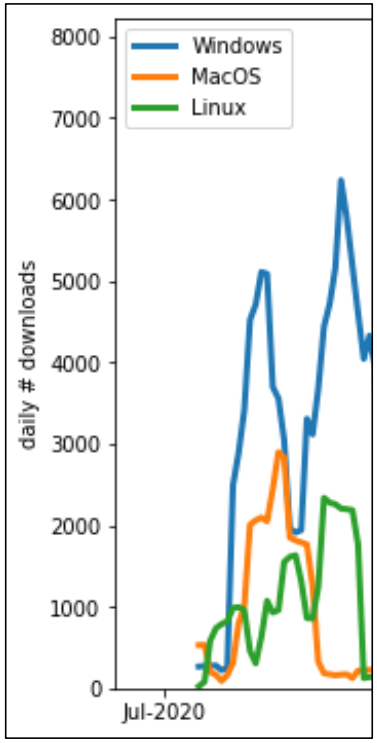
Face Recognition System	Protection Success Rate	
	Without Fawkes	With Fawkes
Amazon	0%	**100%**
Microsoft	0%	**100%**
Face++	0%	**100%**

Some Experiences After the Release

- Some experiences since the paper and software release
  - USENIX Security, August 2020
  - https://sandlab.cs.uchicago.edu/fawkes/
  - Binary releases for Mac, Windows, Linux (source code also on Github)

- Two experiences of note:
  - Targeting algorithm and datasets have huge impact
  - Azure: what happens when users test your software on external platforms
  - … and some thoughts on usability challenges

1. Initial Response from Binary Release



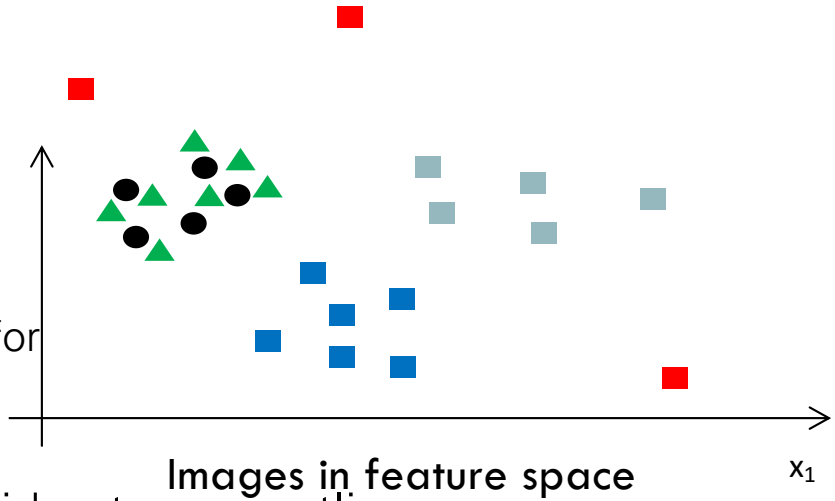
- Great initial reception
  - Particularly by hacker community (y.comb/reddit was first)
  - Now 4K stars on github
- Tons of media and user emails on prototype
- Most early feedback was great
  - But apparently poisoning is a really difficult concept for the public. ½ of news articles thought Fawkes was evasion attack
- We thought: yay!?

2. Real World “Testing”

- Not everyone’s images turned out so well
- At first, we thought… Embarrassing.. maybe they’re just outliers…
- Turns out outliers were consistent: women & babies often got mustaches!!

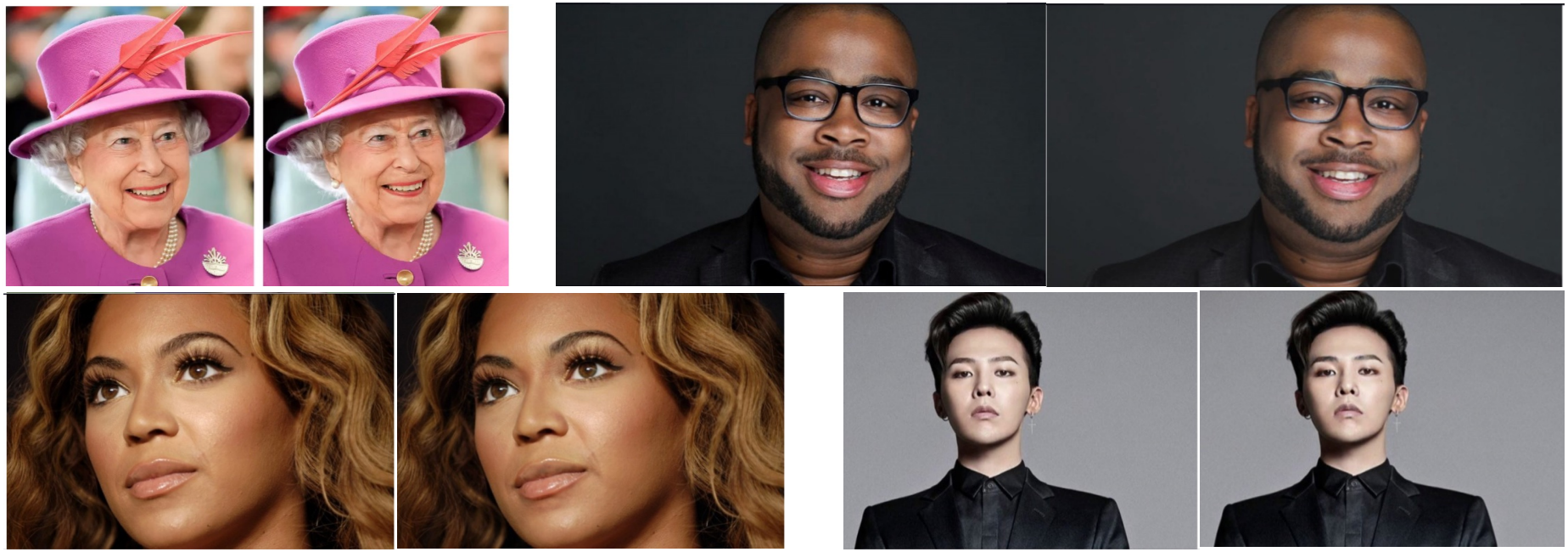
Diagnosing the Mustache Effect

- The fault lies in our target selection algorithm
  - Targets chosen from large library of public images
  - Default: find max dist in feature space
  - Surprise: small set of images with lots of facial hair
    - Consistently chosen as max distance target for women and babies
  - Solution: modify distance function to avoid extreme outliers



Results

- Much improved results after updating targeting algorithm



3. (Indirect) Interaction w/ Azure

- February 2021, we saw the ICLR Lowkey paper online
  - Showed shockingly poor results for Fawkes, esp on MSFT Azure
  - Definitely got our attention

LowKey: Leveraging Adversarial Attacks to Protect Social Media Users from Facial Recognition

Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John Dickerson, Gavin Taylor, Tom Goldstein

- After testing/diagnosis, we found:
  - LowKey Amazon Rekognition experiments used Fawkes as evasion attack
  - Biggest surprise was Azure MSFT updated model, now adversarially trained against Fawkes v0.3

Fawkes 0.3 vs Azure 2020	Fawkes 0.3 vs Azure 2021	Fawkes 0.3 + new FE Azure 2021	Fawkes 1.0 vs Azure 2021
95%	25%	70%	93%

4. Broad Interest in Protection against Tracking

LowKey: Leveraging Adversarial Attacks to Protect Social Media Users from Facial Recognition

Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John Dickerson, Gavin Taylor, Tom Goldstein

Unlearnable Examples: Making Personal Data Unexploitable

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, Yisen Wang

Face-Off: Adversarial Face Obfuscation

Varun Chandrasekaran, Chuhan Gao, Brian Tang, Kassem Fawaz, Somesh Jha, Suman Banerjee

FoggySight: A Scheme for Facial Lookup Privacy

Ivan Evtimov, Pascal Sturmfels, Tadayoshi Kohno

- **Multiple papers in recent months targeting same challenge, hopefully more to come**