TOWARDS

SECURE & INTERPRETABLE AI

Scalable Methods, Interactive Visualizations, Practical Tools

Polo Chau
Associate Professor
Associate Director, MS Analytics
Georgia Tech

poloclub.github.io

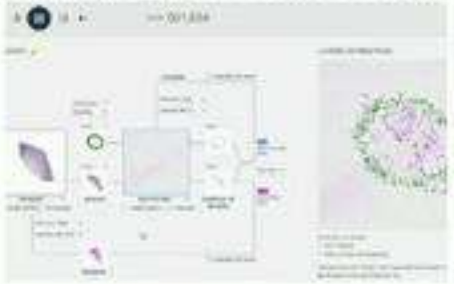
Polo Club of Data Science poloclub.github.io

Human-Centered AI

ActiVis

Visual Exploration of Facebook Deep Neural Network Models

Deployed Facebook



GAN Lab

Playing with Generative Adversarial Networks in Browser

Google

Cyber Security

Cyber MoneyBall

Predicting Cyber Threats with Vi Security Products

Symantec

MARCO

Fake Review Detection

SDM'14 Best Student Paper

Adversarial ML

SHIELD

Fast, practical defense fo deep learning

Intel

ShapeShifter

1st Targeted Physical Attack on Faster R-CNN Object Detector

Large Graph Mining & Visualization

MMap

Easy billion-scale graph computation on a PC using virtual memory

Apolo

Explore million-node graphs in real time

Social Good & Health

DeepPop

Deep Learning on Satellite Imagery for Population Estimation

Microsoft AI for Earth

Firebird

Predicting Fire Risk in Atlanta

KDD'16 Best Student Paper, runner-up

Deployed Atlanta Fire Rescue Department

Polo Club of Data Science

AI + **HI**

ARTIFICIAL INTELLIGENCE

HUMAN INTELLIGENCE

Scalable interactive tools to make sense of complex large-scale datasets and models

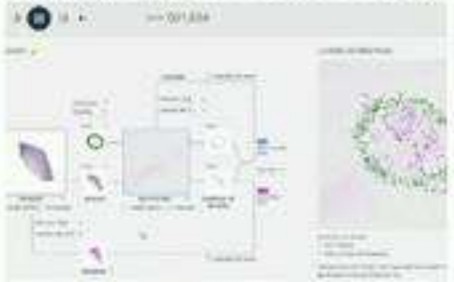
Minsuk Shang Fred Nilaksh Haekyu Scott Anmol Bob Joon Jonathan Alex Will Omar Sudeep Polo

Polo Club of Data Science poloclub.github.io

Human-Centered AI

ActiVis
Visual Exploration of Facebook Deep Neural Network Models

Deployed Facebook



GAN Lab
Playing with Generative Adversarial Networks in Browser

Google

Cyber Security

Cyber MoneyBall
Predicting Cyber Threats with Vi Security Products

Symantec

MARCO
Fake Review Detection

SDM'14 Best Student Paper

Adversarial ML

SHIELD
Fast, practical defense fo deep learning

Intel

ShapeShifter
1st Targeted Physical Attack on Faster R-CNN Object Detector

Large Graph Mining & Visualization

MMap
Easy billion-scale graph computation on a PC using virtual memory

Apolo
Explore million-node graphs in real time

Social Good & Health

DeepPop
Deep Learning on Satellite Imagery for Population Estimation

Microsoft AI for Earth

Firebird
Predicting Fire Risk in Atlanta

KDD'16 Best Student Paper, runner-up

Deployed Atlanta Fire Rescue Department

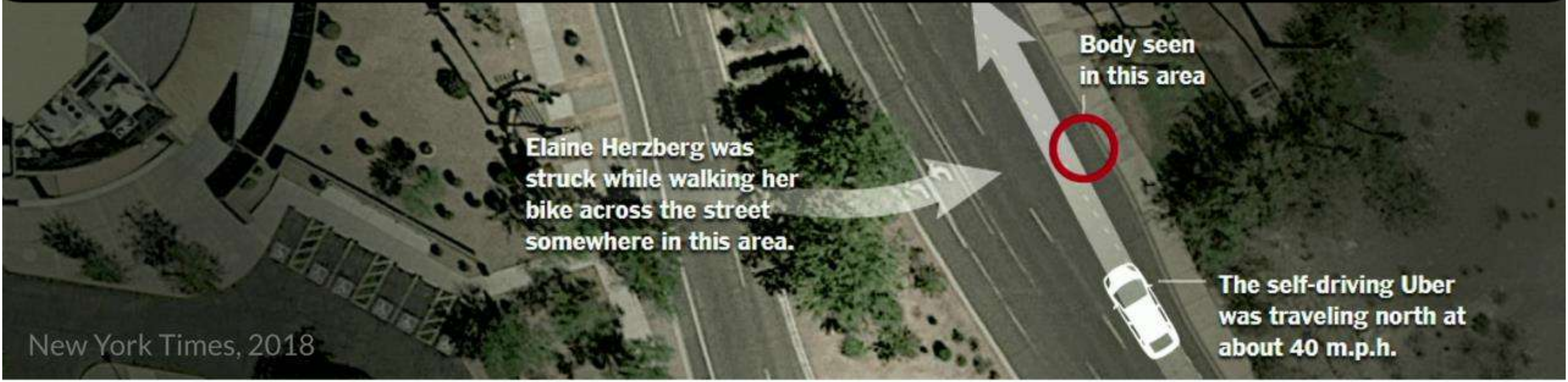
Today’s Main Topics

Secure AI

Interpretable AI

Why focus on them?
How are they related?

AI now used in safety-critical applications. Important to study threats & countermeasures.

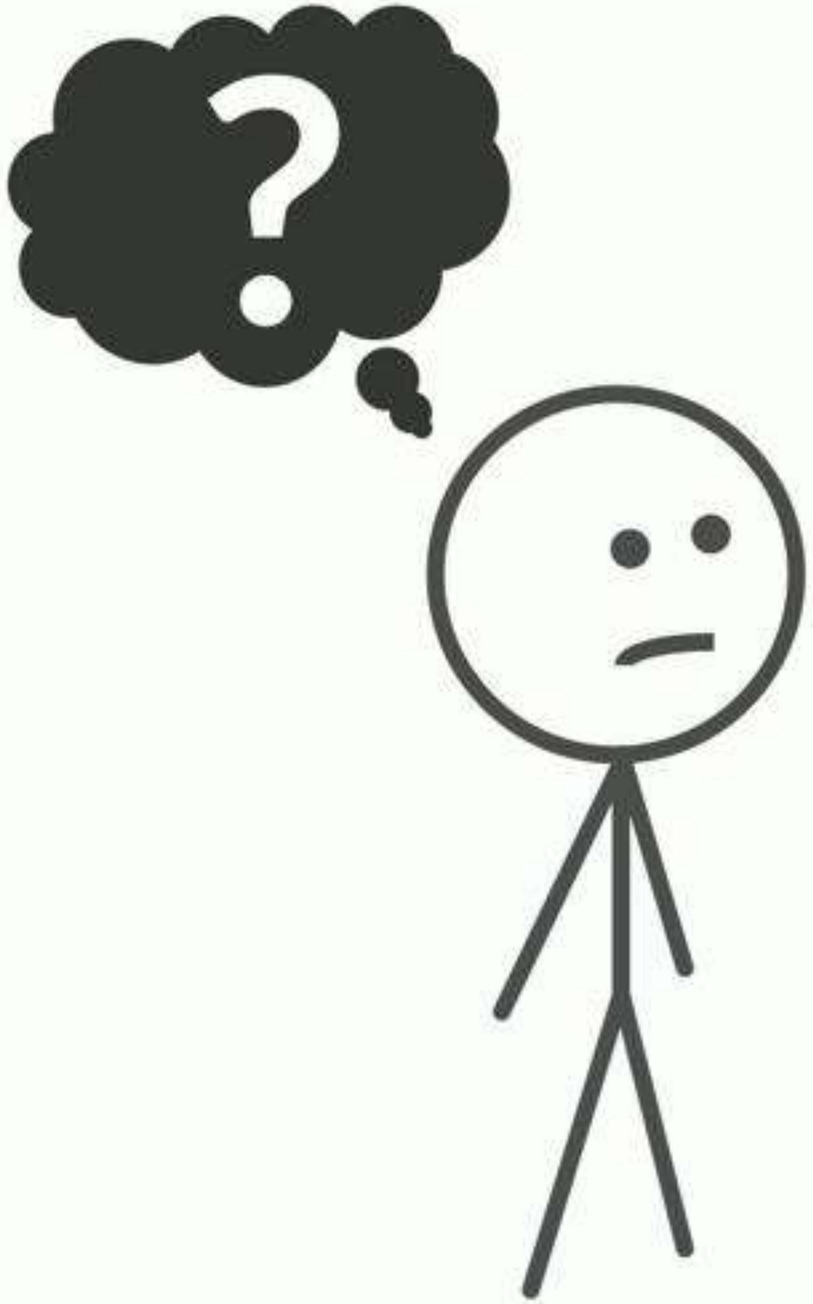


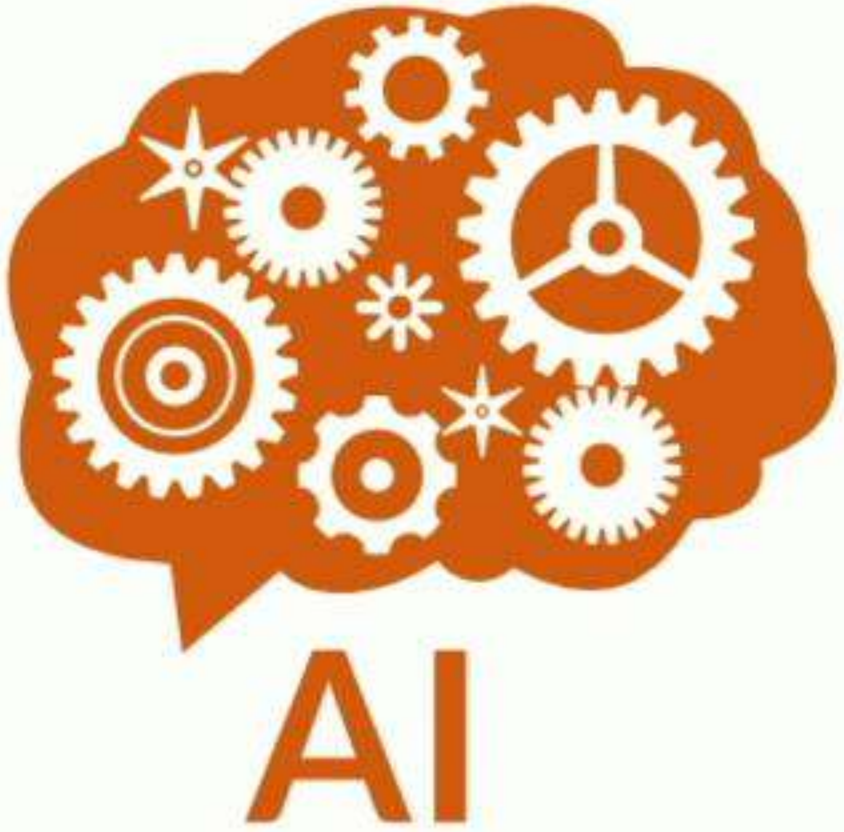
How a Self-Driving Uber Killed a Pedestrian in Arizona

AI now used in safety-critical applications.
Important to study threats & countermeasures.

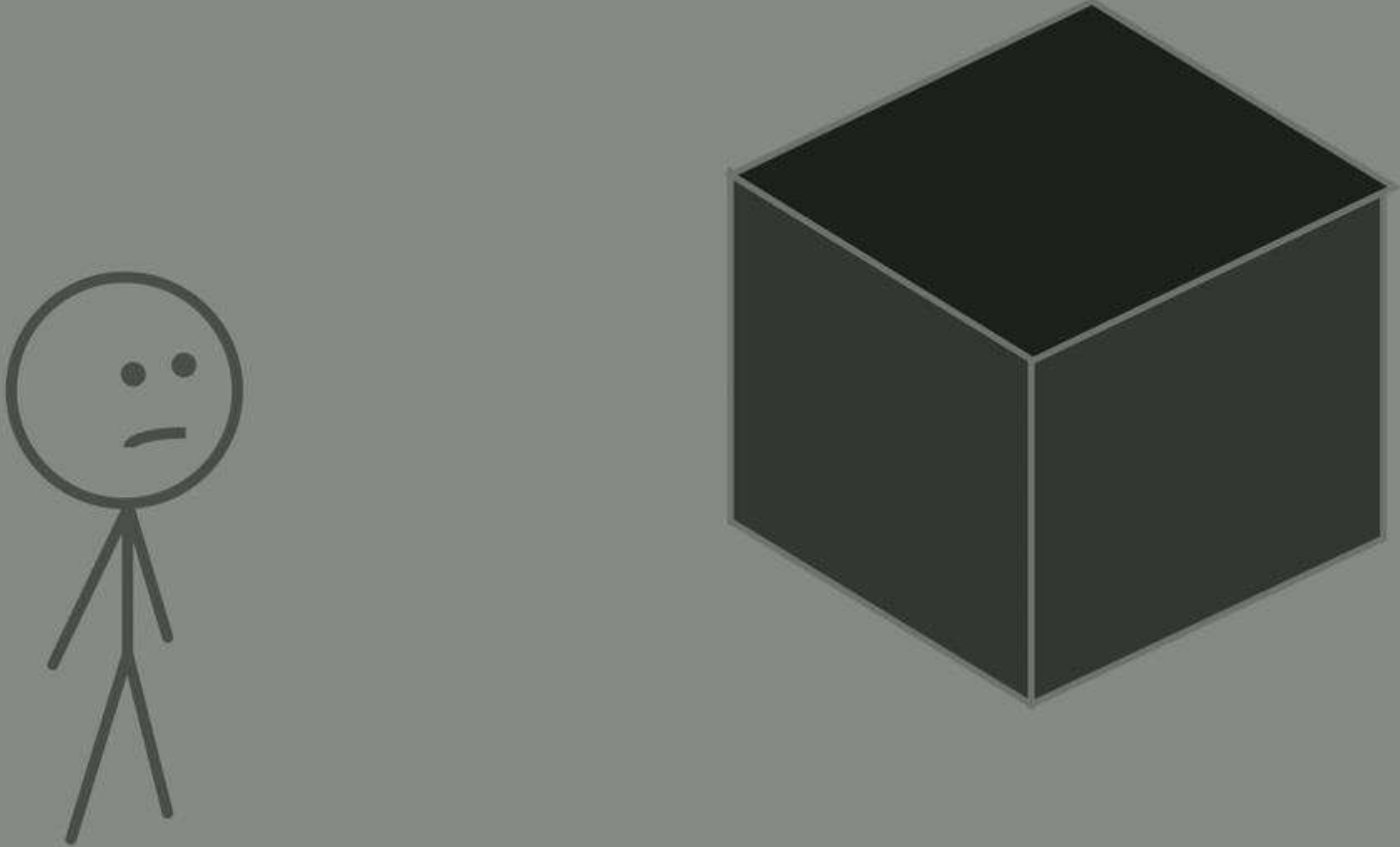
How a Self-Driving Uber
Killed a Pedestrian in Arizona

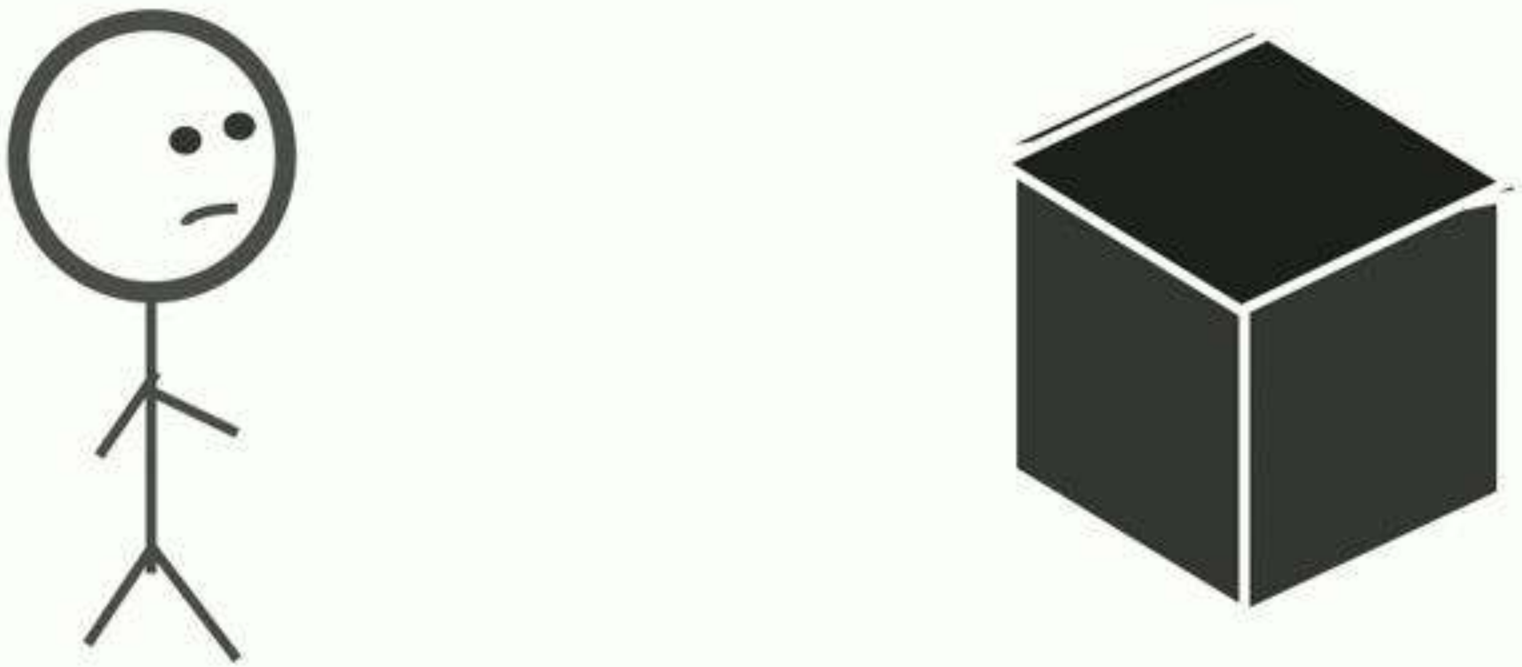
How do we know if a defense for AI is working?





AI models often used as black-box





Interpretable AI

Interpretable AI

Via scalable, interactive, usable interfaces to help people understand complex, large-scale ML systems.

Secure AI

Attack & Defense (DNN)

ShapeShifter

SHIELD

Do-it-yourself Adversarial ML

ADAGIO

MLsploit

Interpretable AI

Understand Industry Models

ActiVis

Interactive Learning (Education)

GAN Lab

Research landscape

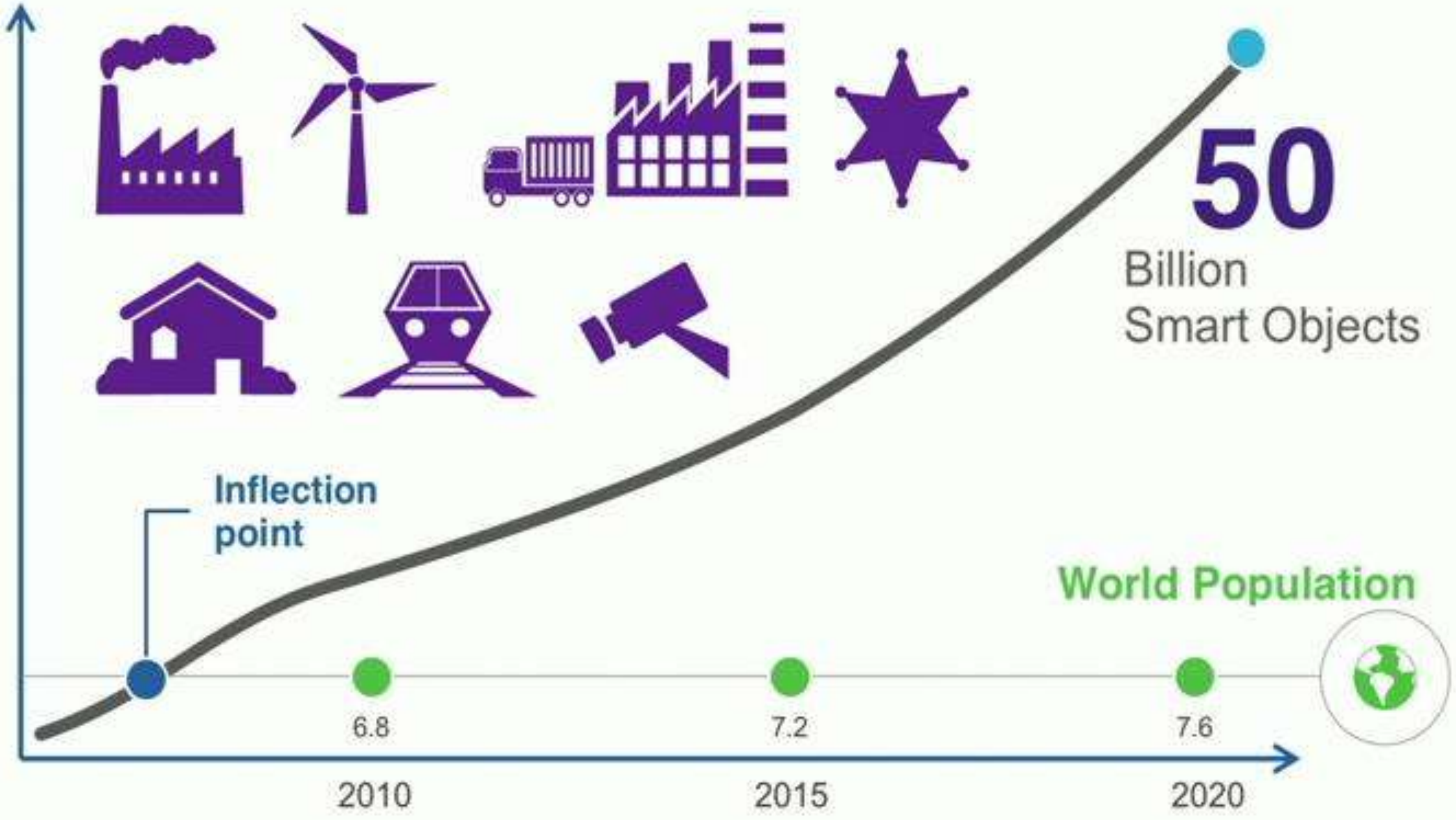
Survey, Gamut

" THE TOASTER HAS BEEN HACKED INTO THINKING IT'S A BLENDER. "

AI Security Problems Are Everywhere

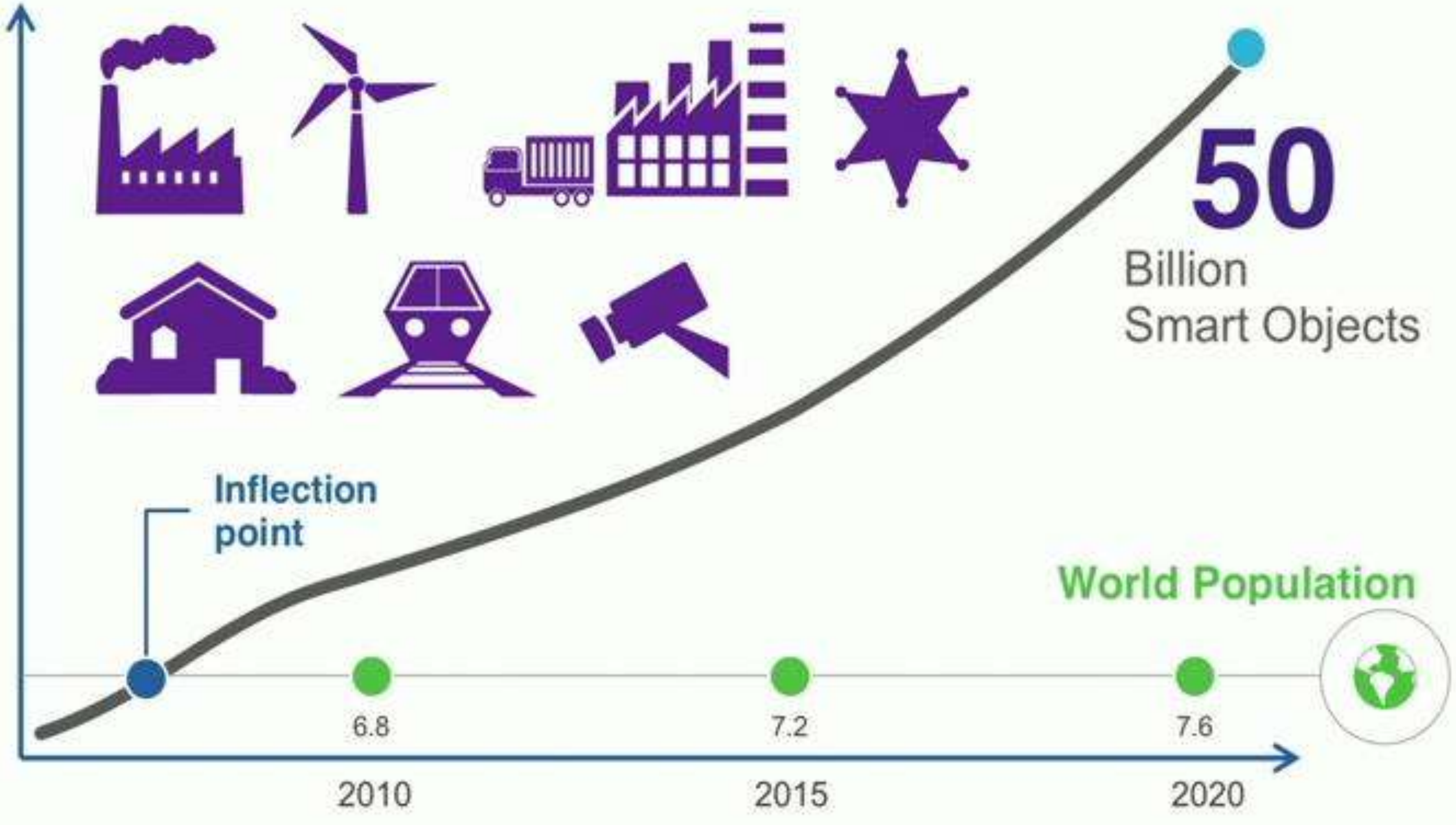
Smart toaster does exist!

AI Security is becoming increasingly important

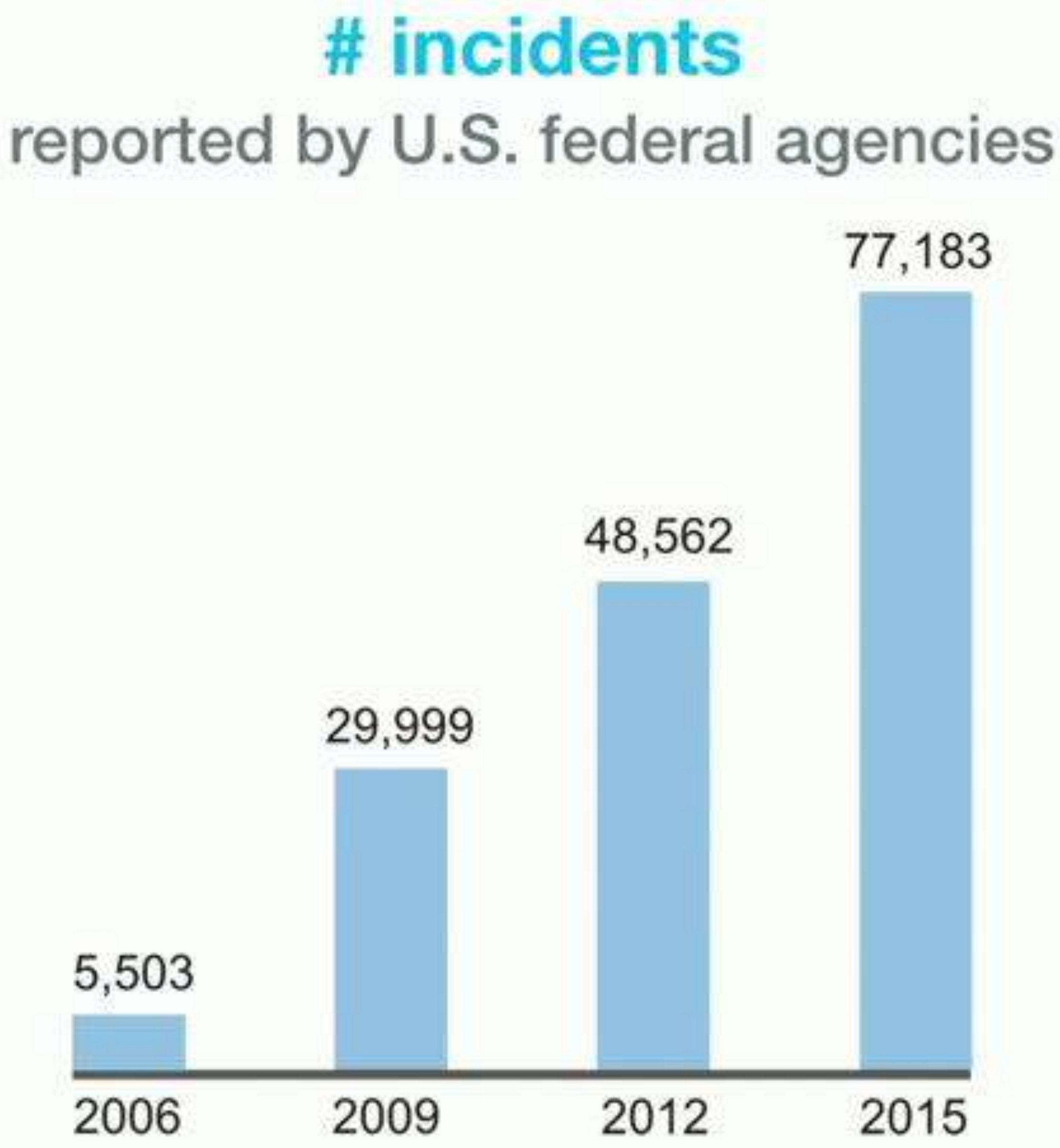


Source: Cisco

AI Security is becoming increasingly important

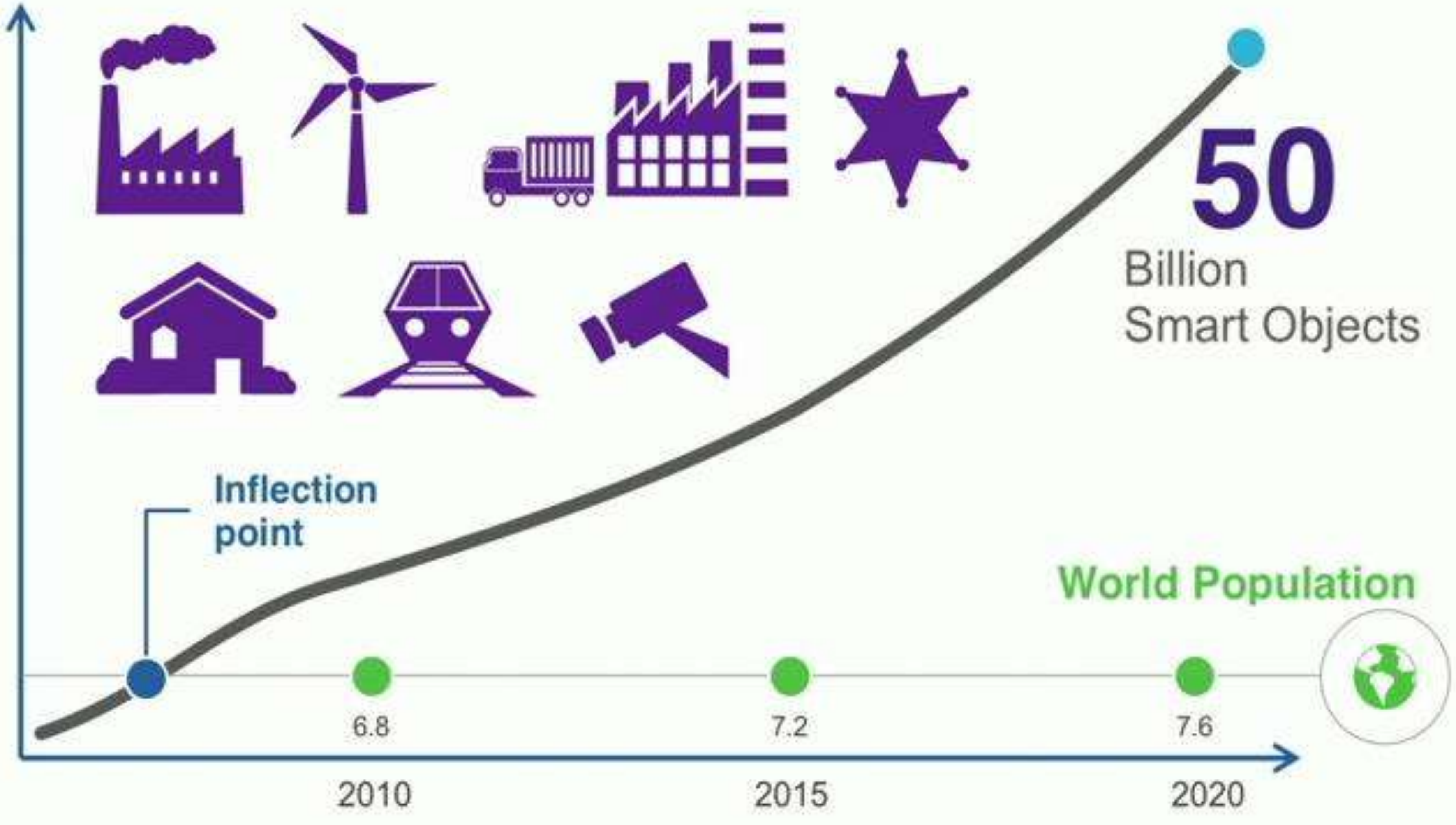


Source: Cisco



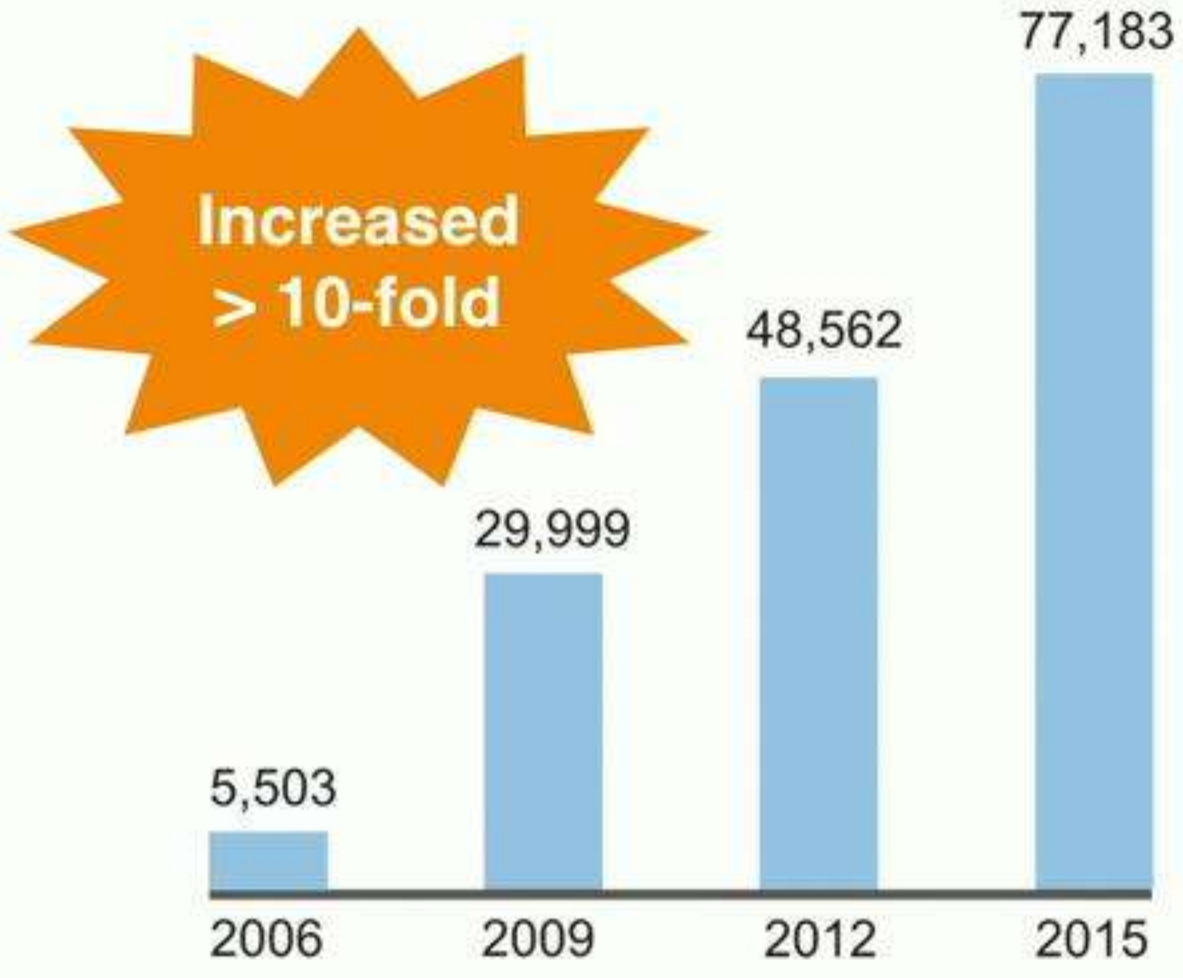
Source: US Department of Homeland Security

AI Security is becoming increasingly important



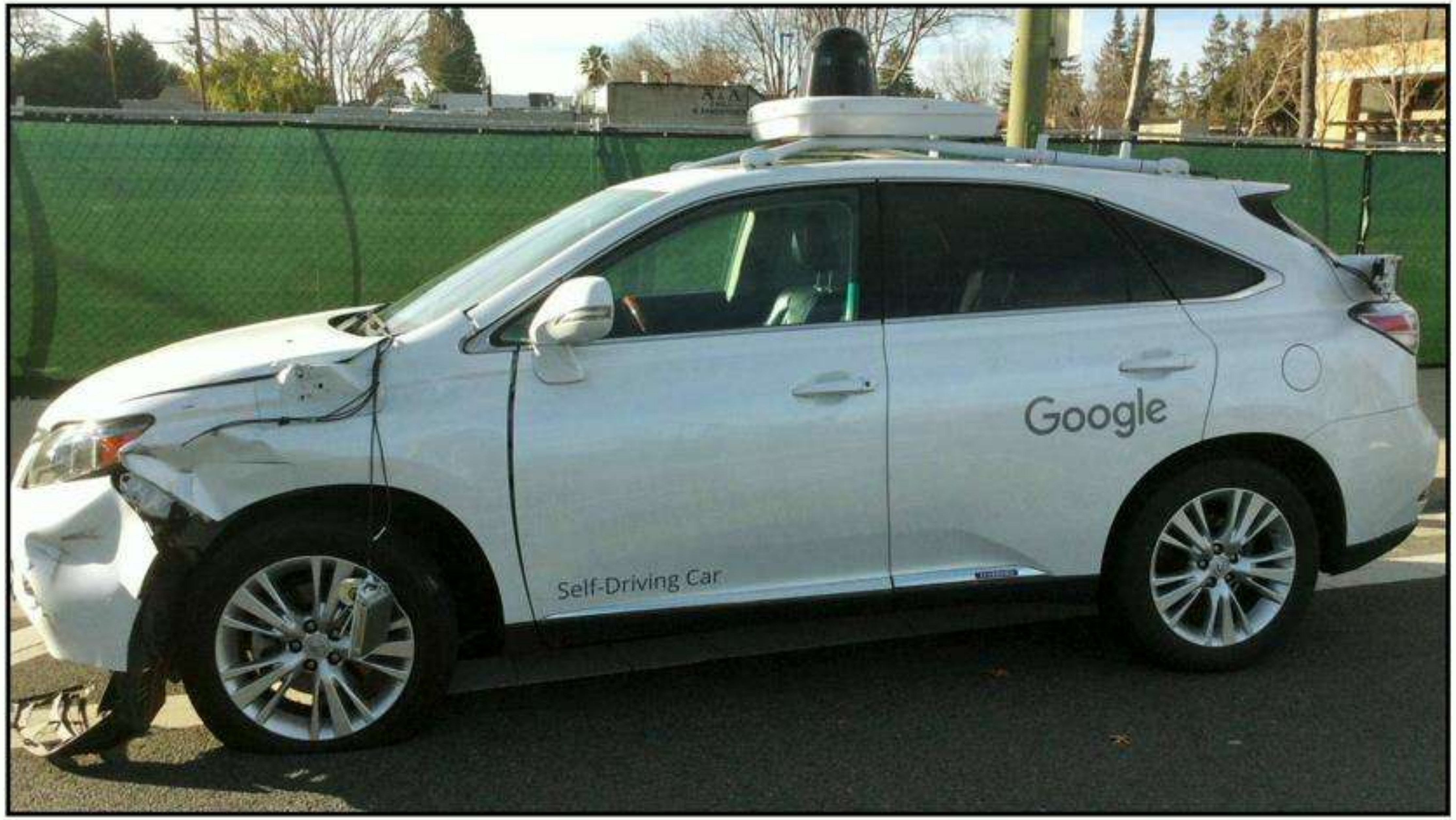
Source: Cisco

incidents
reported by U.S. federal agencies

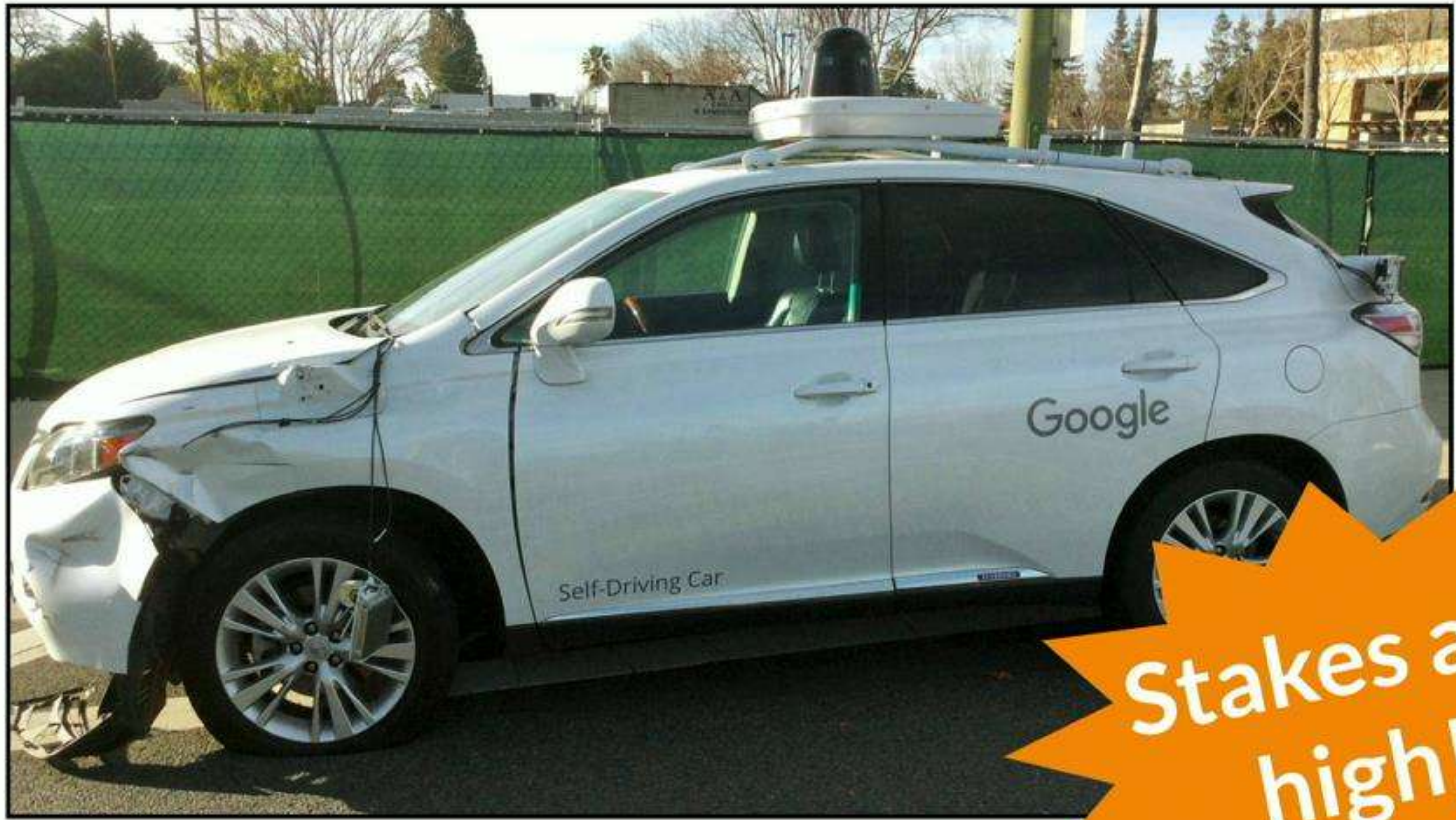


Source: US Department of Homeland Security

AI in Safety-Critical Applications



AI in Safety-Critical Applications



Stakes are high!

Our Goal

Study ML vulnerabilities and develop secure AI for high-stakes problems

Secure AI

Attack & Defense of Deep Neural Networks

ShapeShifter - Physical Adversarial Attack

SHIELD - Real-time Defense for *Images*

Do-it-yourself Adversarial ML

ADAGIO - Experimentation with Real-time Defense for *Audio*

MLsploit - Interactive Experimentation with Adversarial ML

ShapeShifter

ECML-PKDD 2018

First Targeted *Physical* Adversarial Attack for Object Detection




Shang-Tse Chen
Georgia Tech

Cory Cornelius
Intel

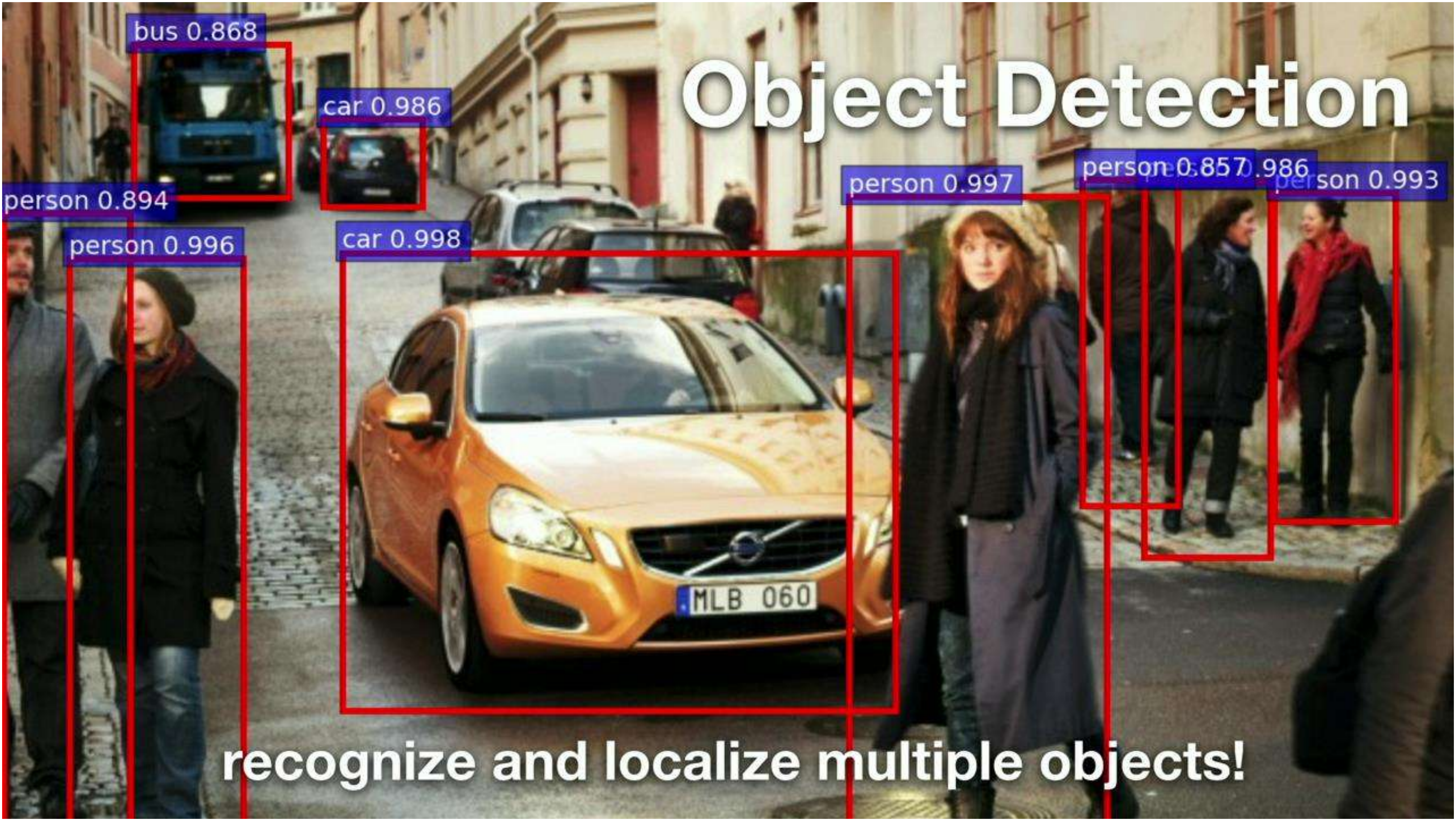
Jason Martin
Intel

Polo Chau
Georgia Tech

Georgia Tech intel


Image Classification
MLB 060
output a single label, e.g., “car”


Object Detection
MLB 060


Object Detection
bus 0.868
car 0.986
person 0.894
person 0.996
car 0.998
person 0.997
person 0.857
person 0.993
MLB 060
recognize and localize multiple objects!

Deep Neural Networks are vulnerable

Deep Neural Networks are vulnerable

Deep Neural Networks are vulnerable

Classified as
Stop Sign

Deep Neural Networks are vulnerable

Benign Image

Classified as
Stop Sign

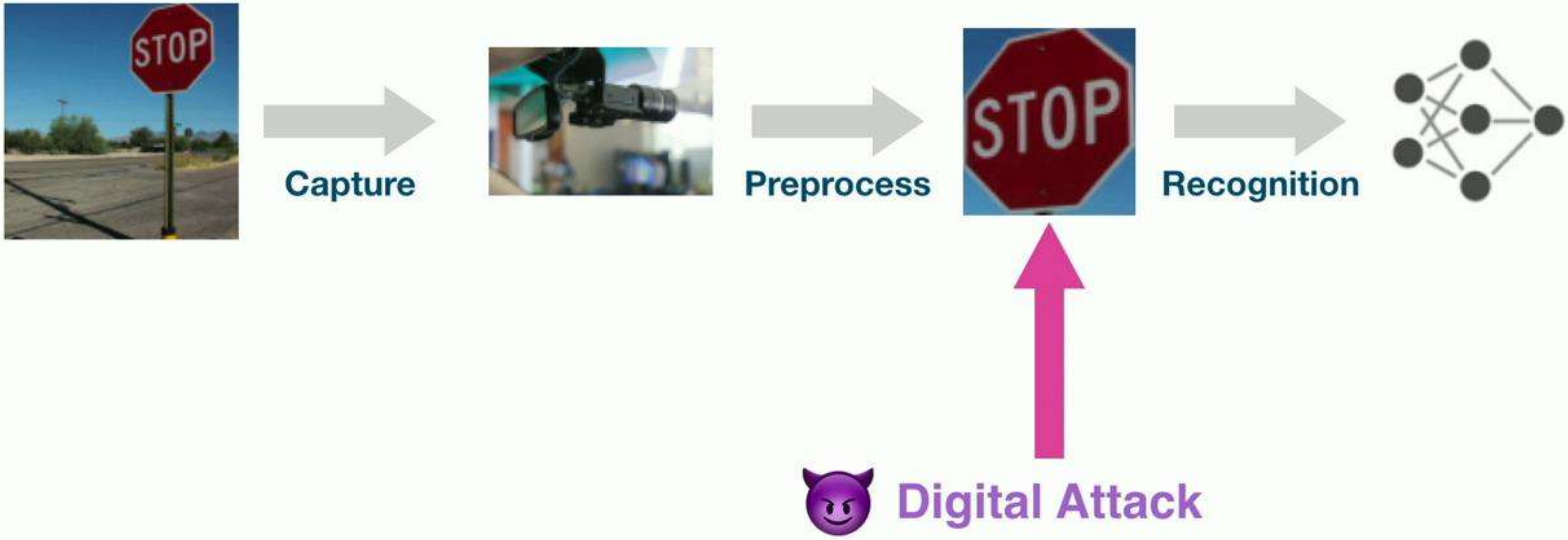
Adversarial Perturbation

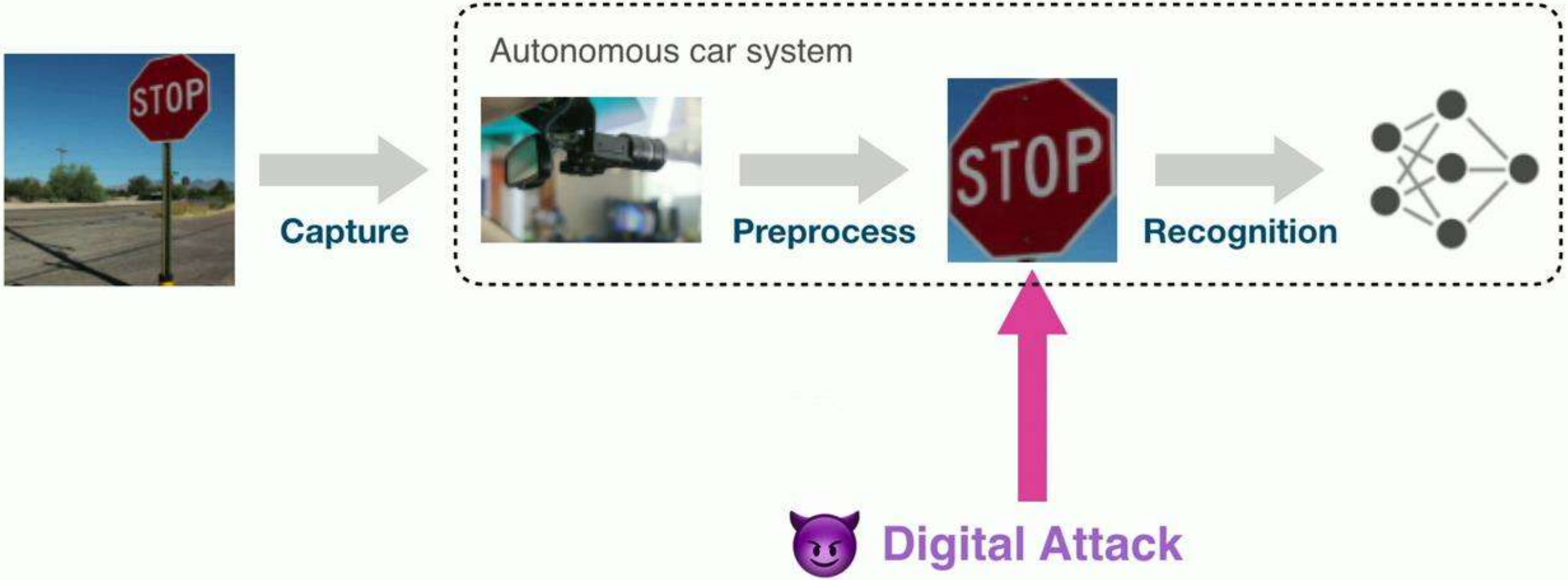
Misclassified as
Max Speed 100

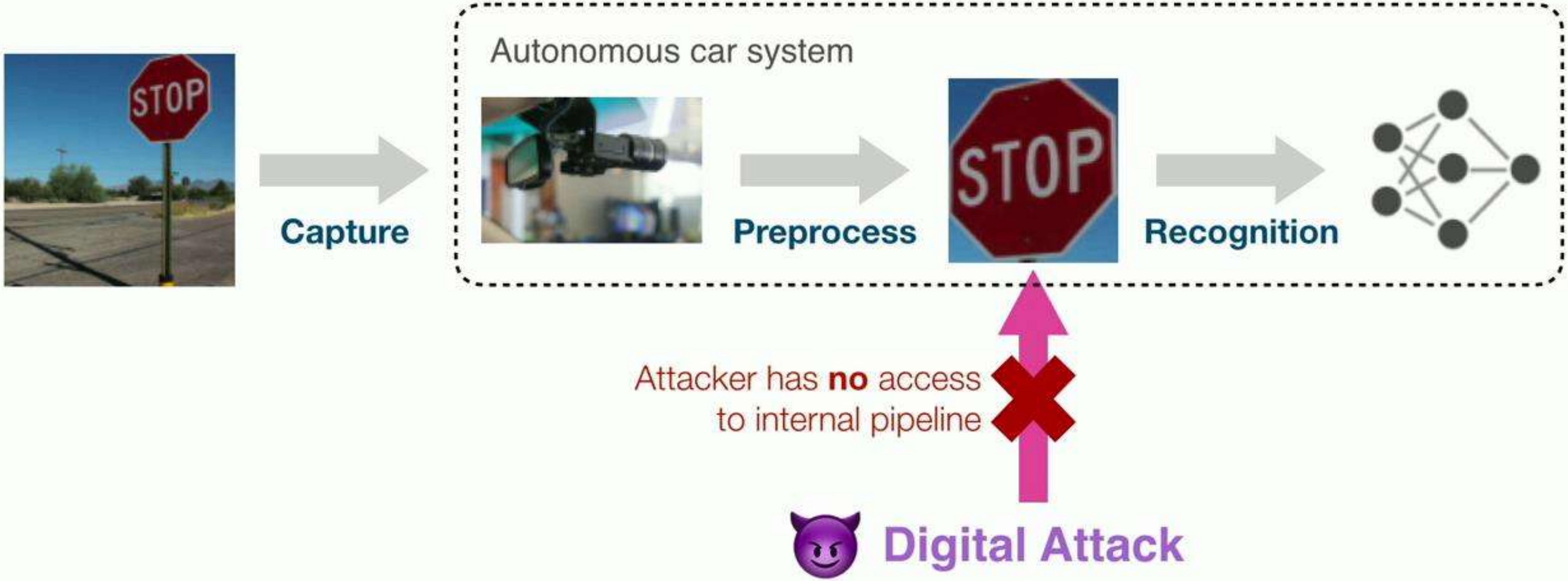
But most attacks have **impractical** threat model


STOP

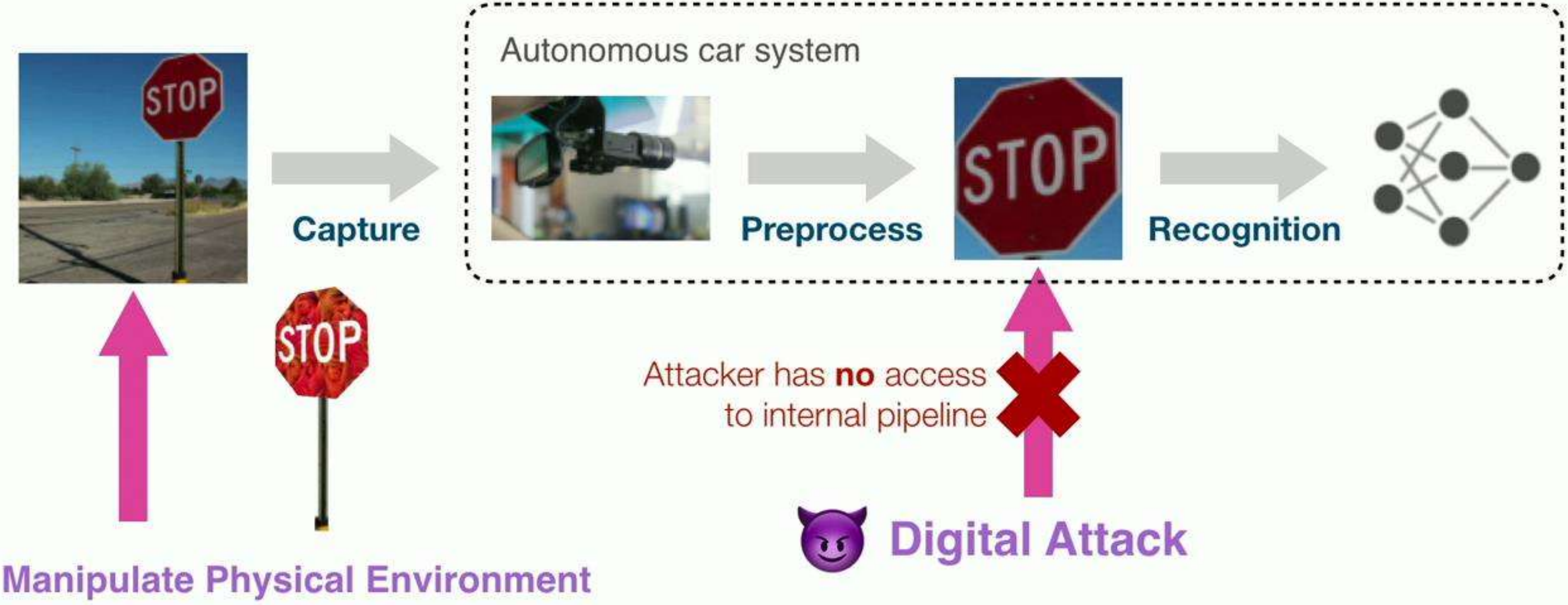

STOP
Capture
Preprocess
STOP
Recognition


STOP
Capture
Preprocess
STOP
Recognition
Digital Attack

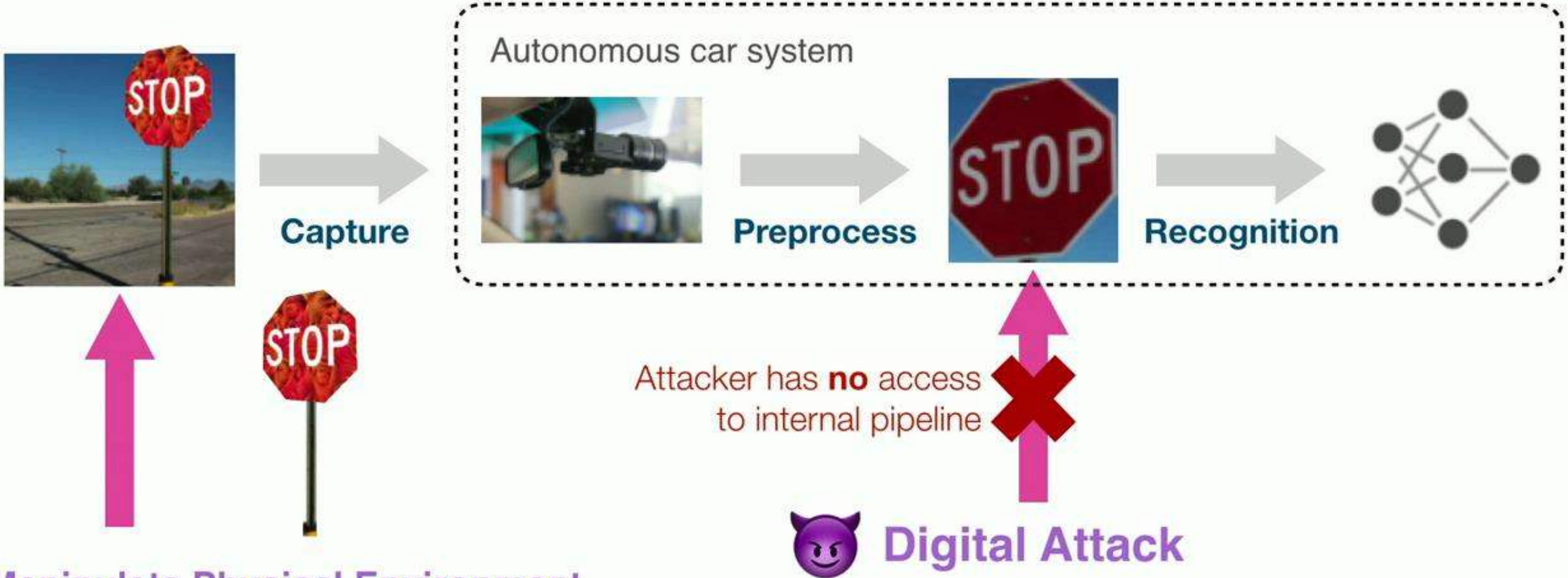

Autonomous car system
STOP
Capture
Preprocess
STOP
Recognition
Digital Attack


STOP
Capture
Autonomous car system
Preprocess
STOP
Recognition
Attacker has **no** access
to internal pipeline
Digital Attack

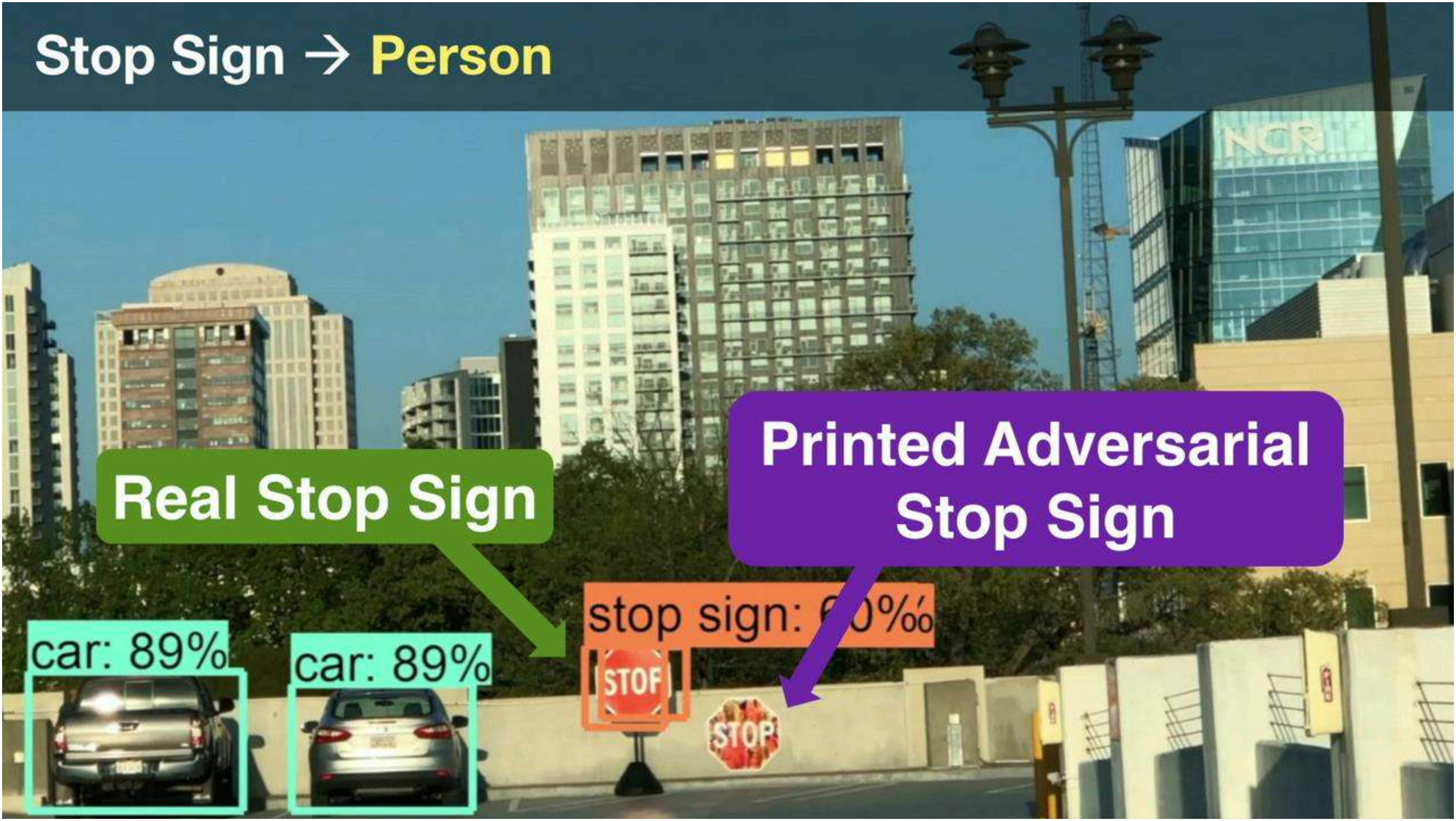
Physically Realizable Adversarial Attack



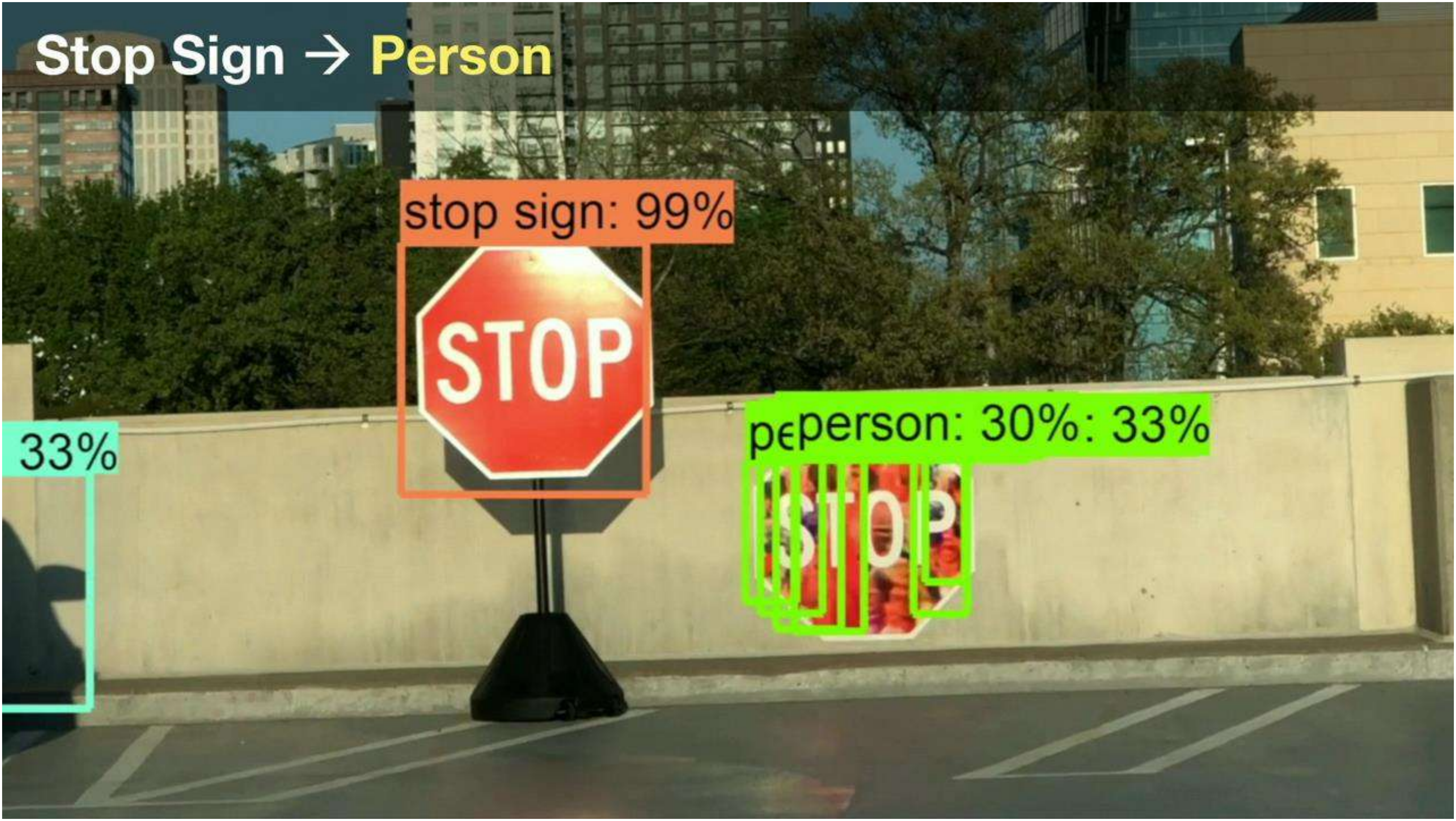
Physically Realizable Adversarial Attack



Manipulate Physical Environment
= More Realistic, *Targeted* Attack



Stop Sign → Person
NCR
Real Stop Sign
Printed Adversarial Stop Sign
stop sign:
car: 89%
car: 89%
STOP
STOP


Stop Sign → Person
stop sign: 99%
STOP
person: 30%: 33%
33%
STOP

Prior Work on Physical Attacks

Glasses that fool
a face classifier
[Sharif et al. CCS'16]

3D objects that fool
an image classifier
[Athalye et al. ICML'18]

Stickers that fool a
traffic sign classifier
[Evtimov et al. CVPR'18]

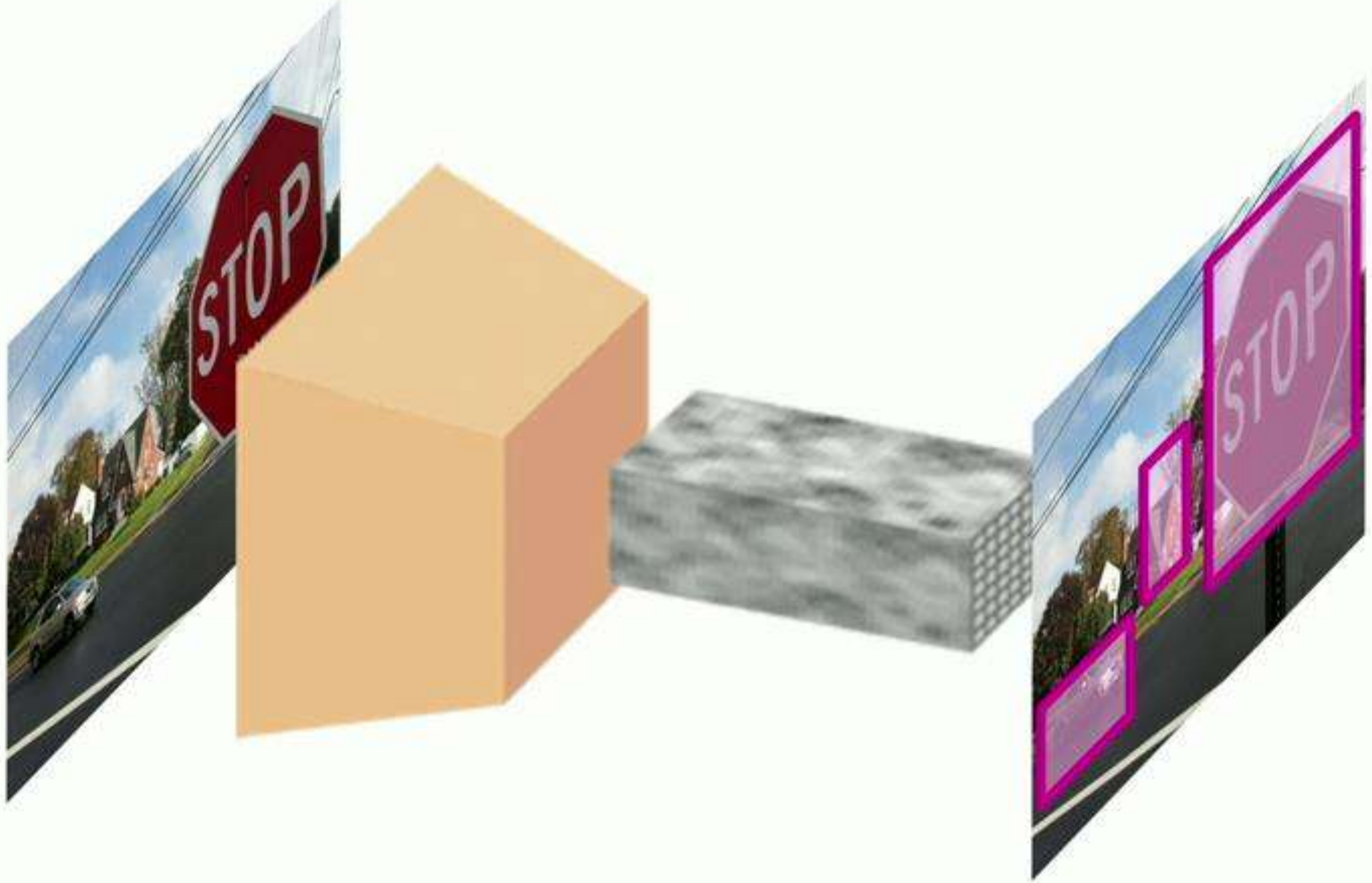
They all focus on attacking **image classifiers**

Attack Object Detectors: Naïve Approach

Lu et al. [1] show the current technique cannot fool state-of-the-art object detectors like Faster R-CNN and YOLO

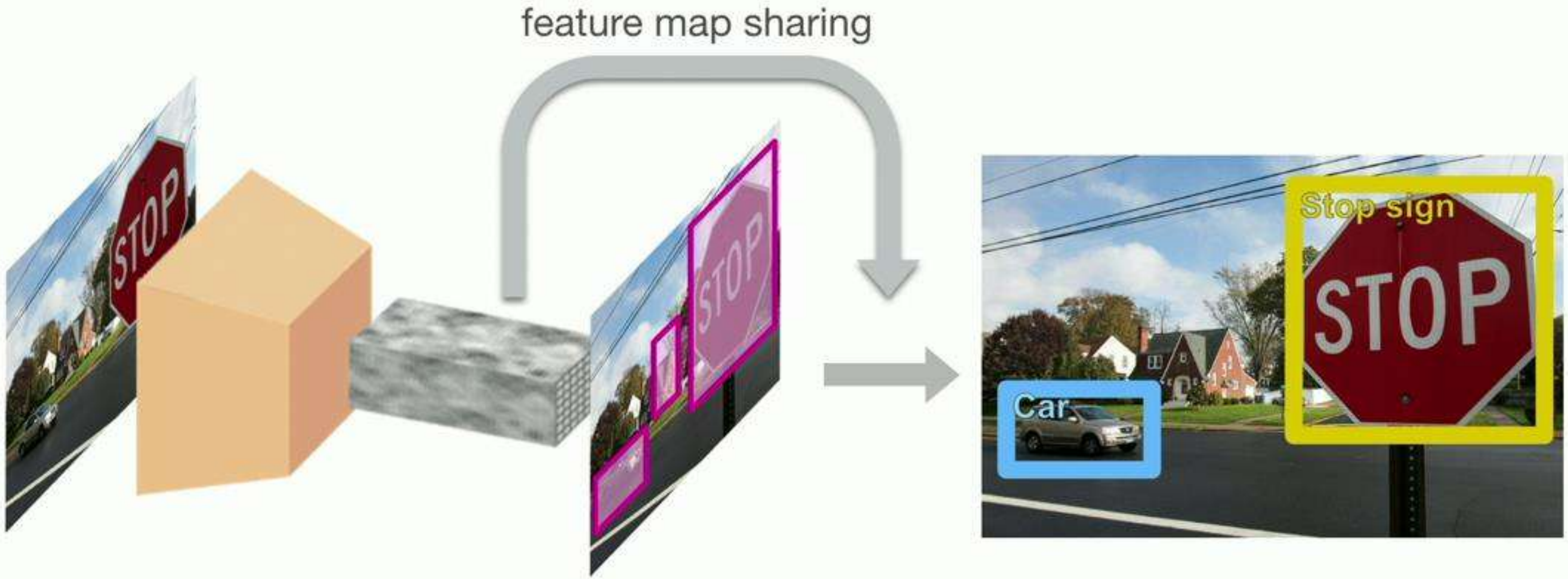
[1] Standard detectors aren't (currently) fooled by physical adversarial stop signs. Lu et al., arXiv '17

Brief Overview of Faster R-CNN



A state-of-the-art Object Detector Model

Brief Overview of Faster R-CNN

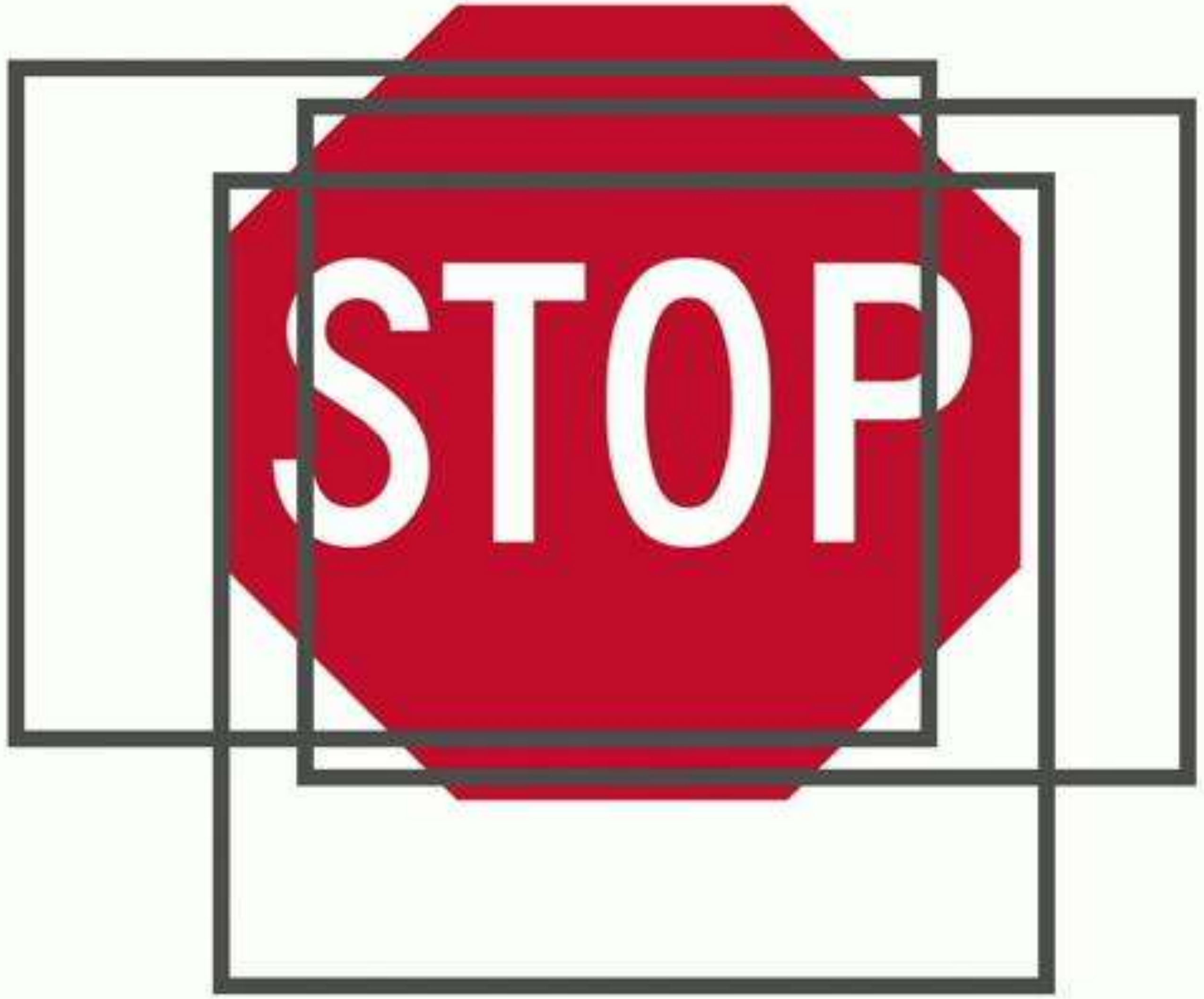


Stage 1:
Generate region proposals

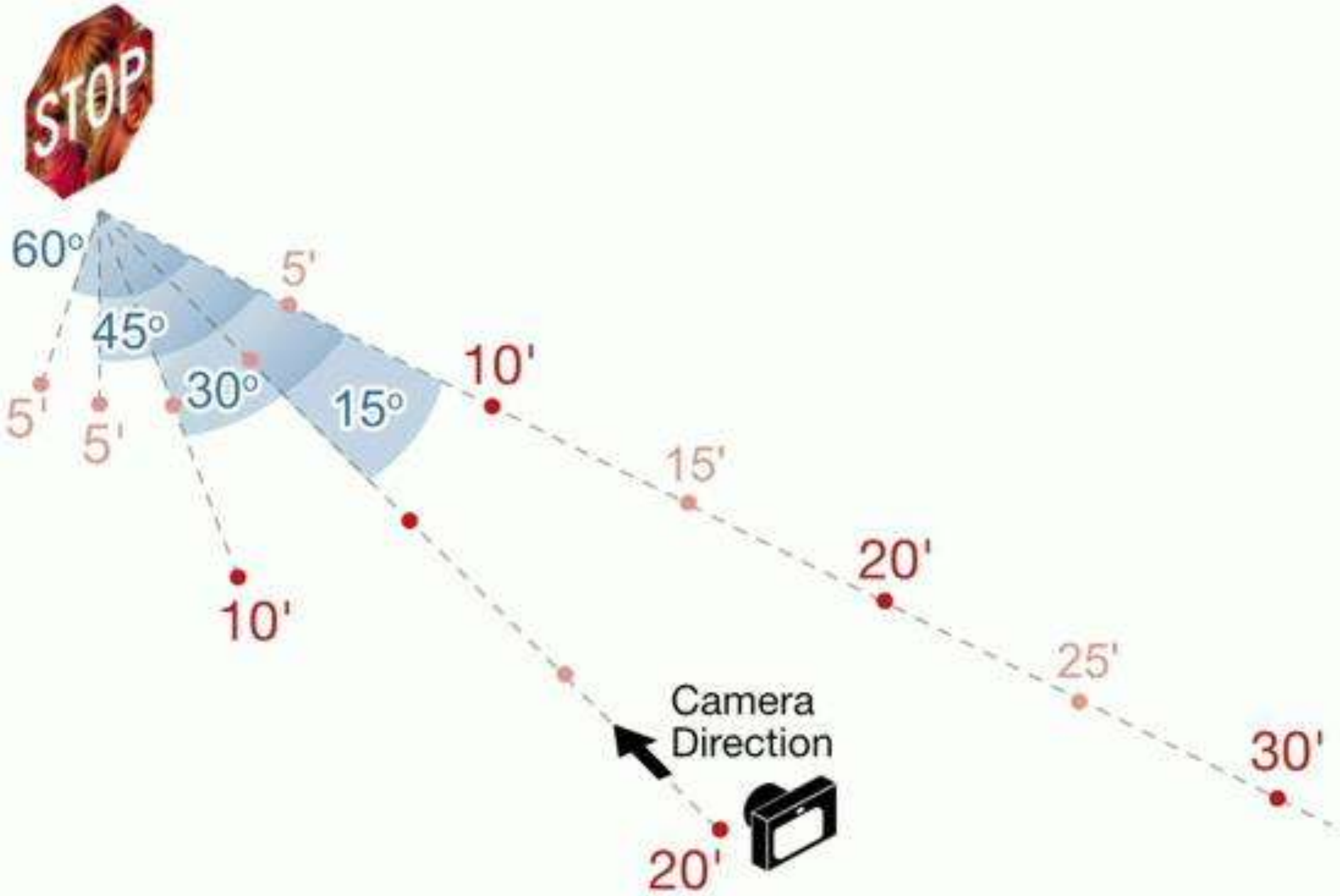
Stage 2:
Refined localization and classification

Challenges of Physically Attacking Faster R-CNN

1. Multiple region proposals



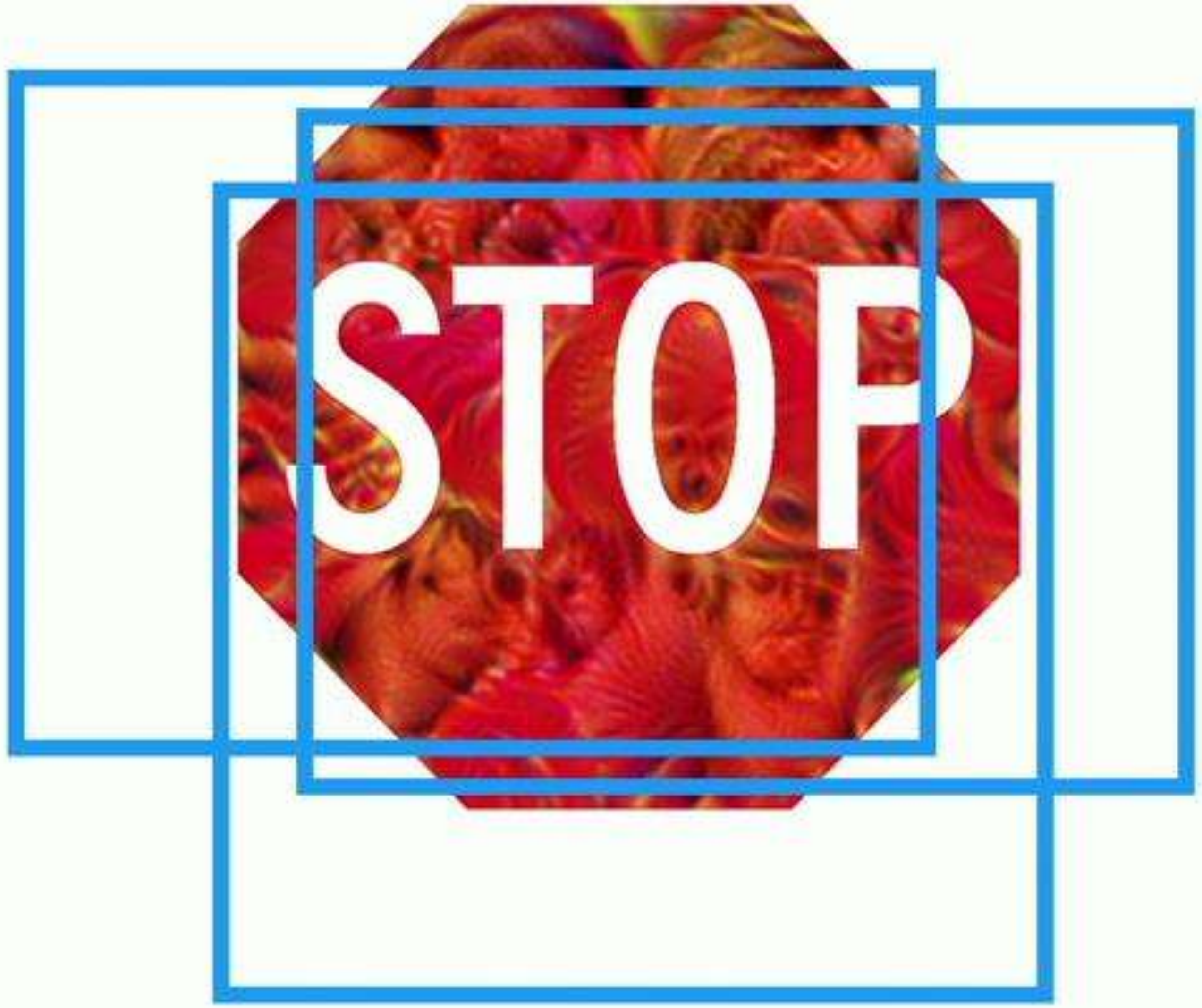
2. Distances, angles, lightings



Solution 1: Fool Multiple Region Proposals

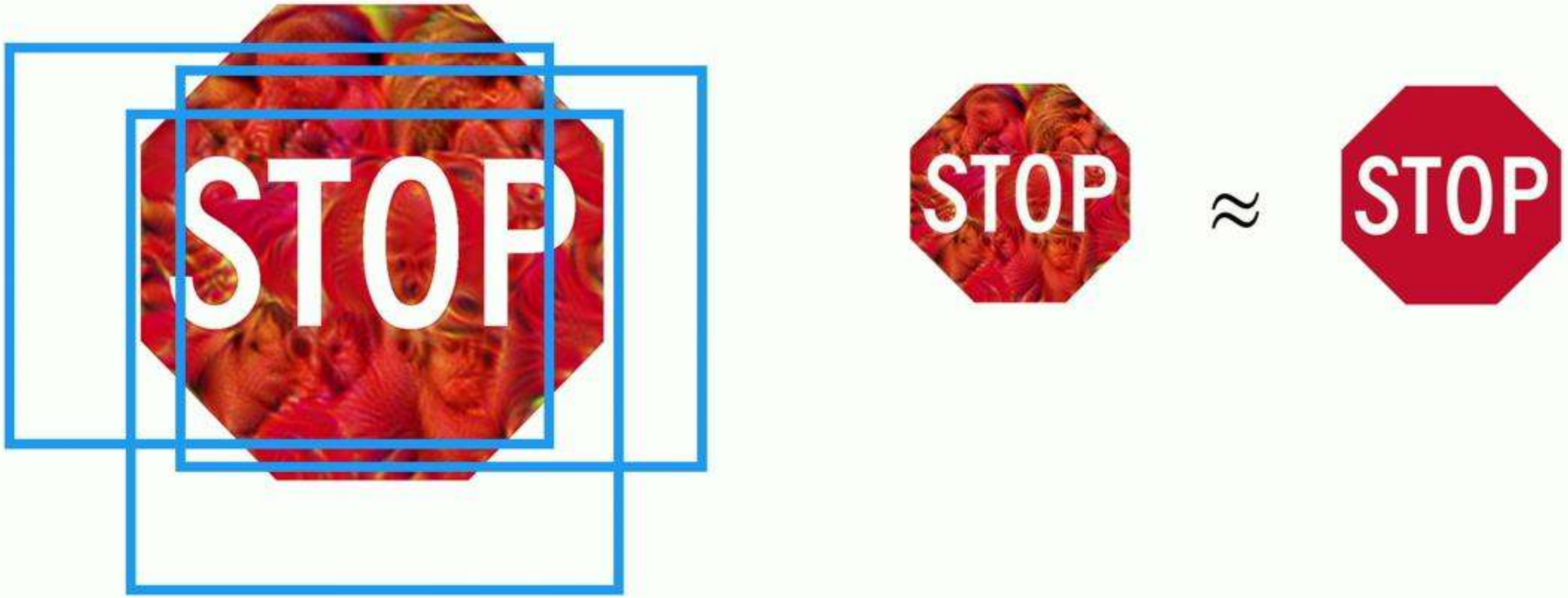
Solution 1: Fool Multiple Region Proposals

Minimize: sum of classification losses



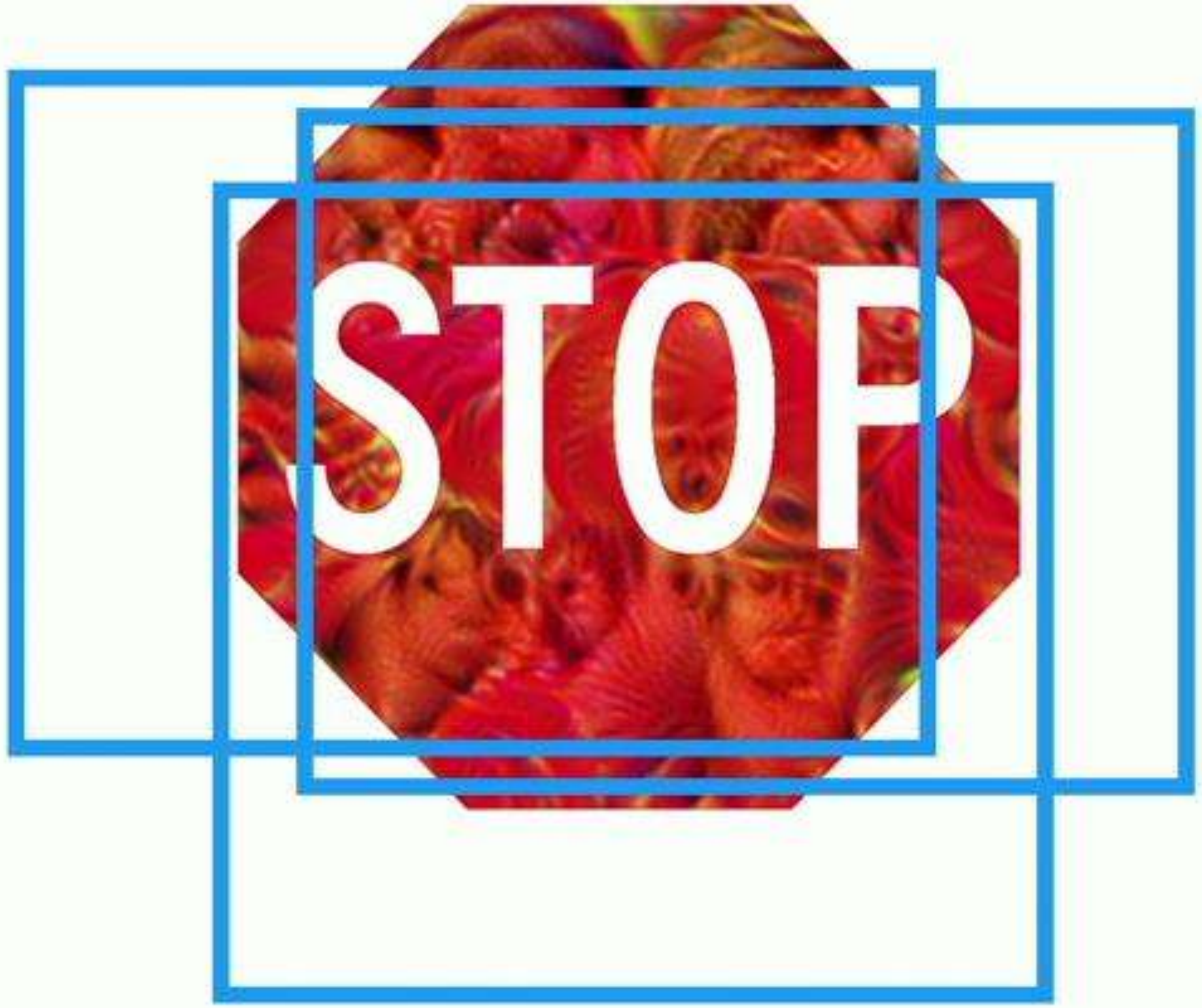
Solution 1: Fool Multiple Region Proposals

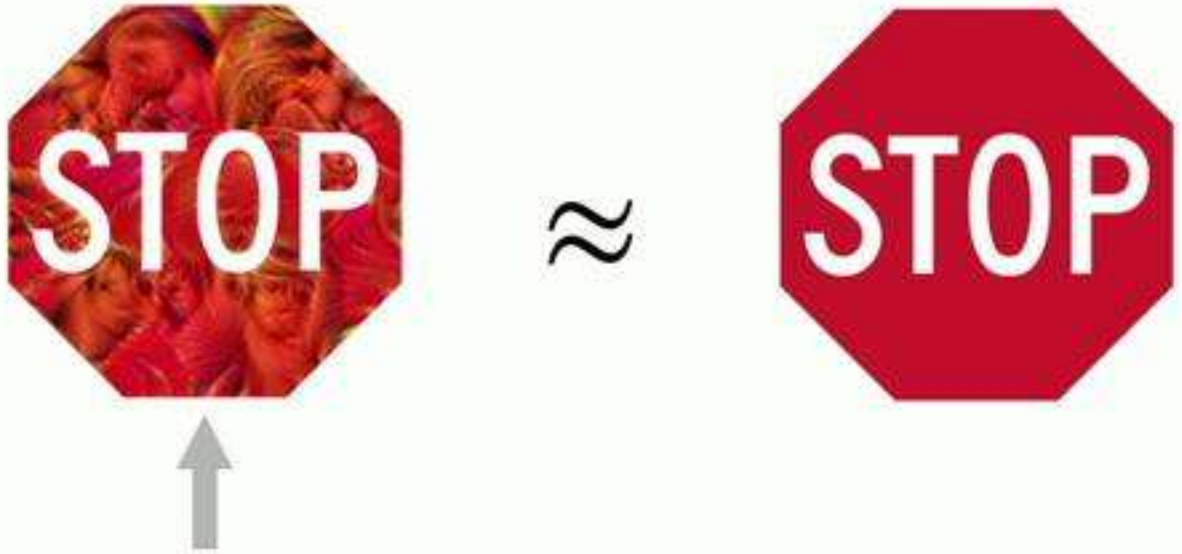
Minimize: sum of classification losses + deviation loss



Solution 1: Fool Multiple Region Proposals

Minimize: sum of classification losses + deviation loss





Only perturb RED area

Human eye is less sensitive to changes in darker color

Solution 2: Robust to Real-World Distortions

Solution 2: Robust to Real-World Distortions

Adapt Expectation over Transformation [Athalye et al, ICML'18]

Solution 2: Robust to Real-World Distortions

Adapt Expectation over Transformation [Athalye et al, ICML'18]

Optimize over different backgrounds, scales, rotations, lightings

Perturbed Stop Signs

Person

Sports Ball

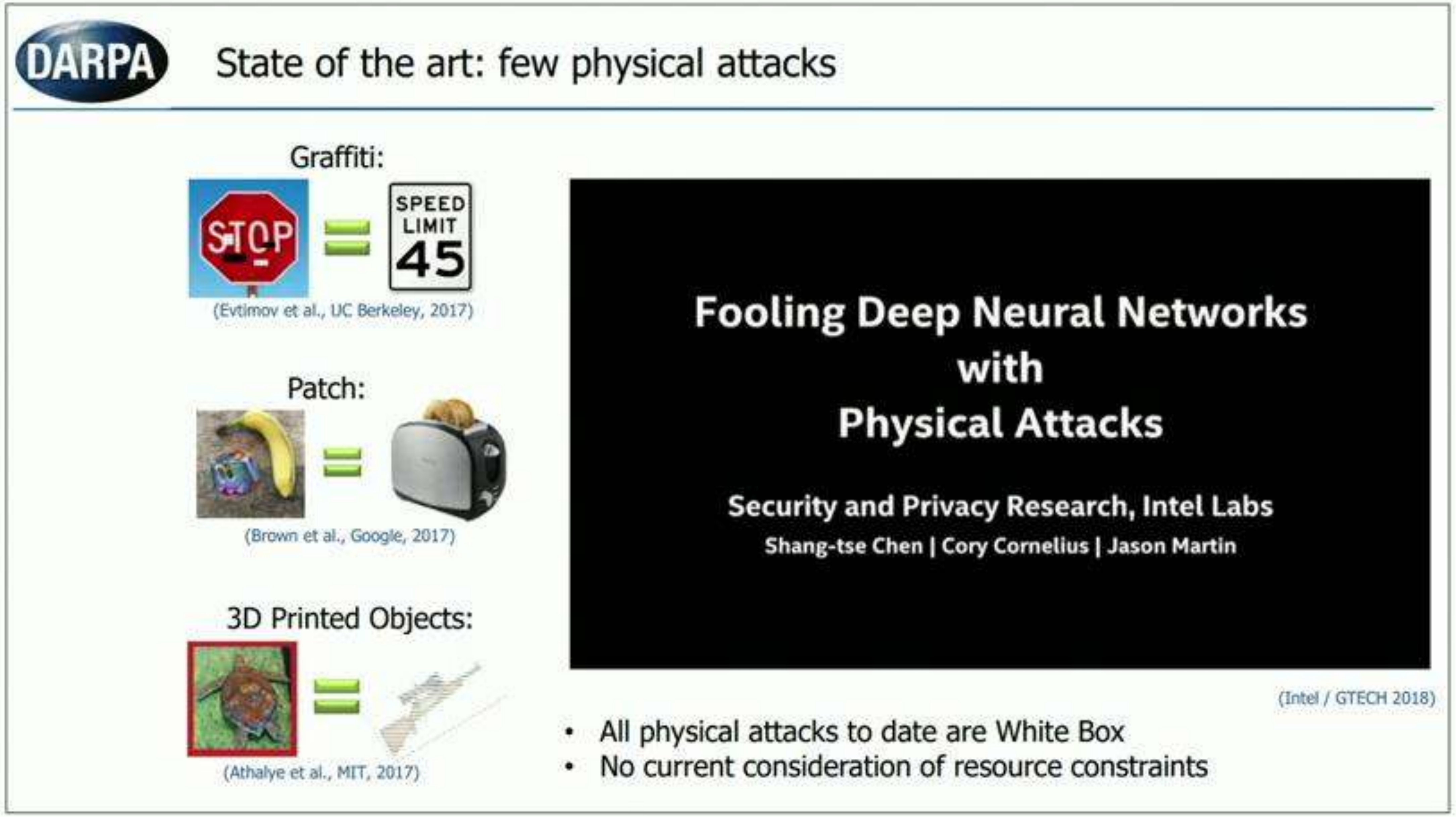
Untargeted

perturbations *more conspicuous* in order to survive
viewing distances, angles, lighting conditions, camera limitations

Untargeted Attack

Untargeted Attack

ShapeShifter Motivates **New DARPA Program**
GARD: Defense for AI



Highlights **ShapeShifter** as the state-of-the-art physical attack

https://www.darpa.mil/attachments/GARD_ProposersDay.pdf

SHIELD Fast, Practical Defense for Image Classification

KDD'18 Audience Appreciation Award (runner-up)
KDD'19 LEMINCS

[Open-sourced]


Nilaksh Das


Madhuri Shangbogue


Shang-Tse Chen


Fred Hohman


Siwei Li


Cory Cornelius


Li Chen


Michael Kounavis


Polo Chau

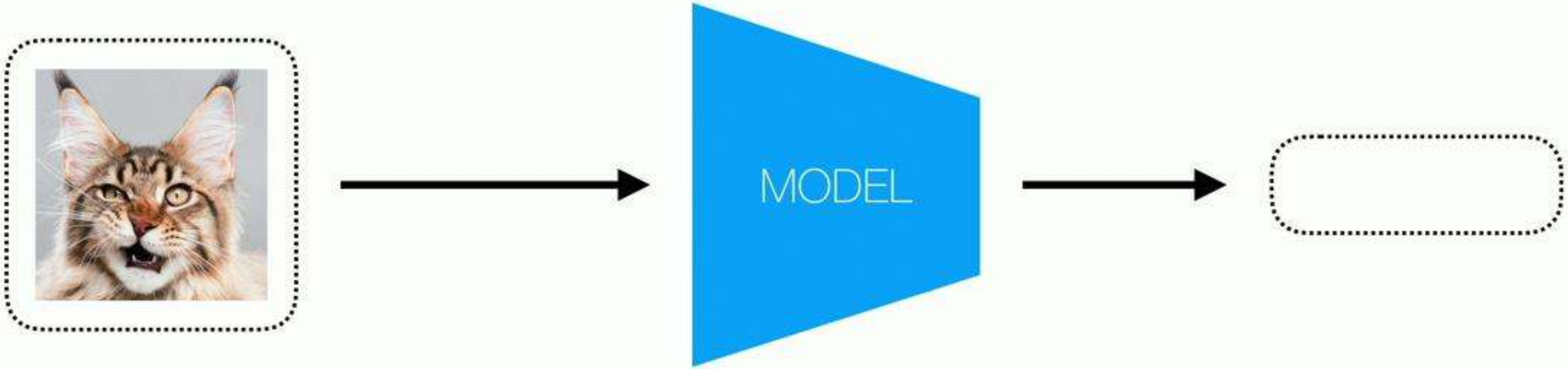
Georgia Tech
intel

Adversarial Machine Learning Landscape

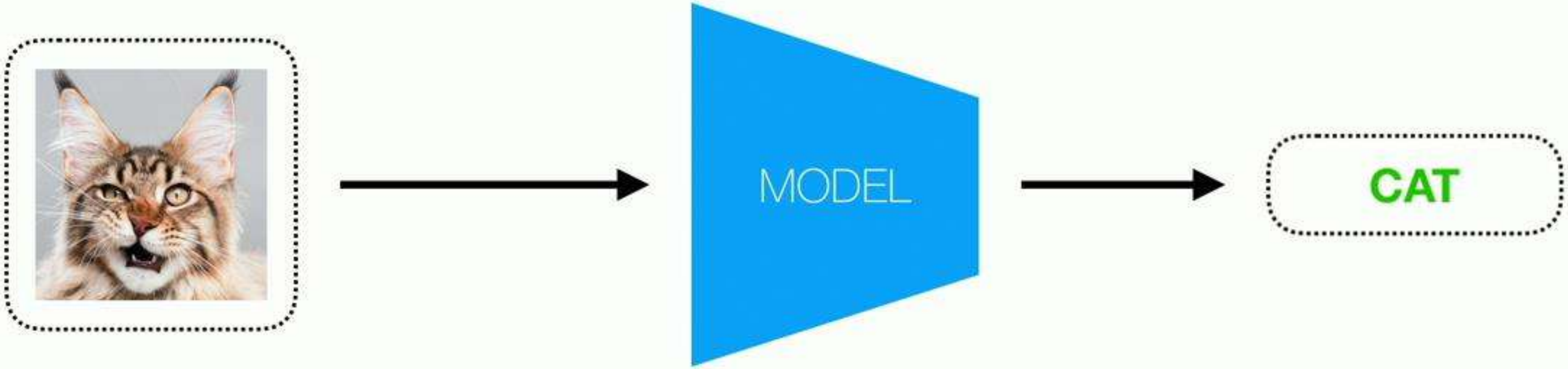
Adversarial Machine Learning Landscape

Our Focus:
Fast & Practical
(digital)

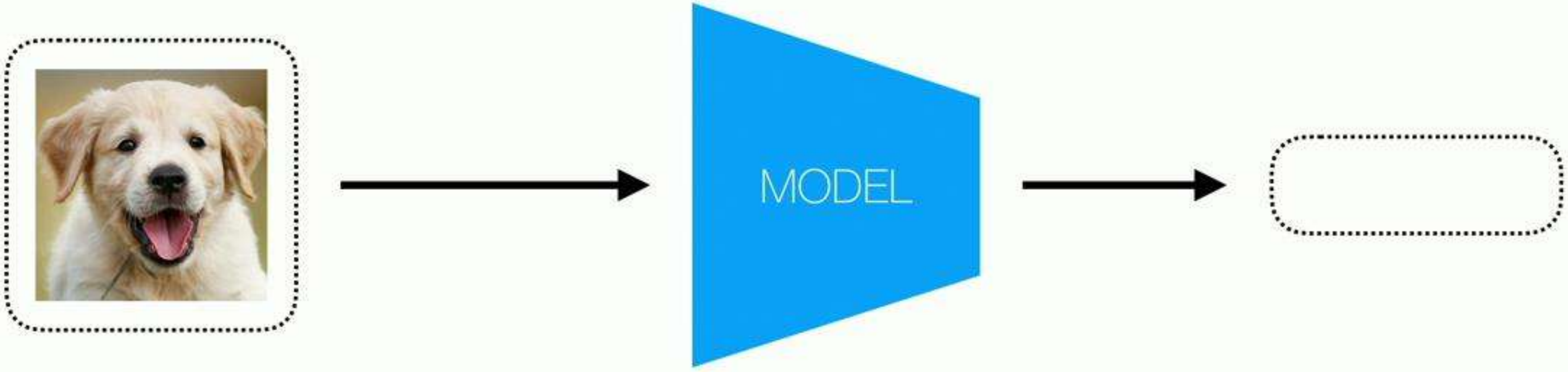
Deep Learning for Image Classification



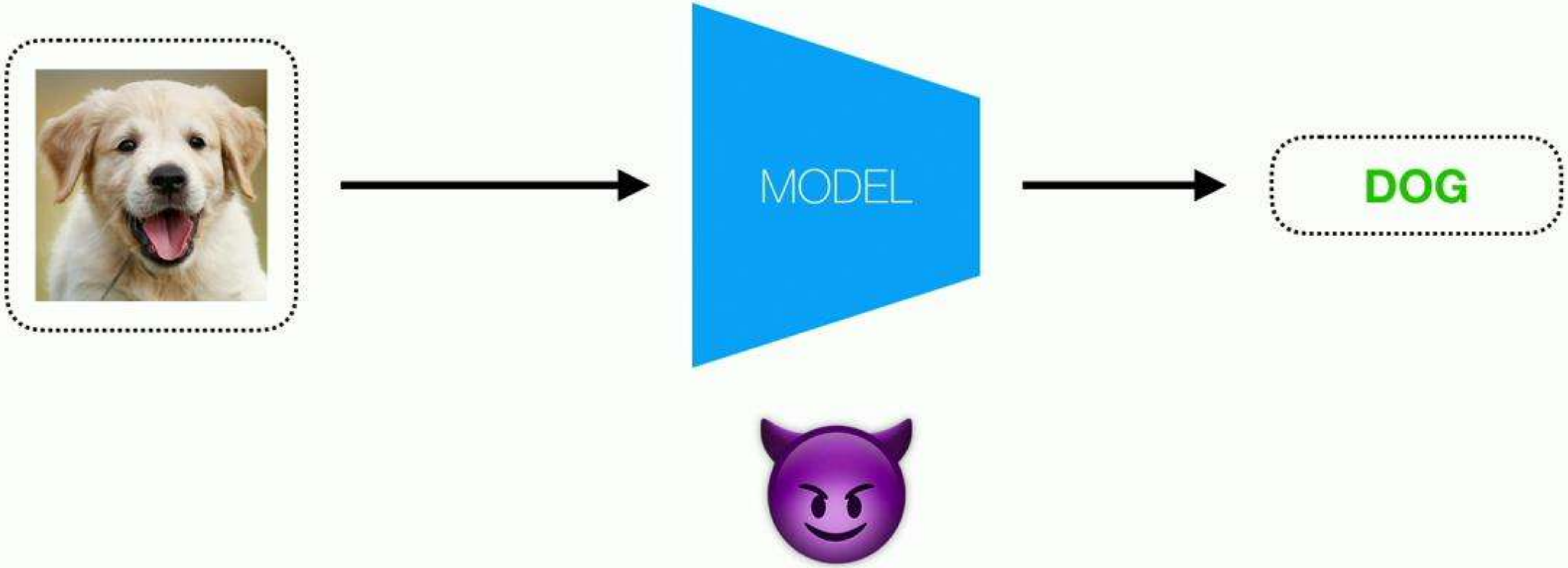
Deep Learning for Image Classification



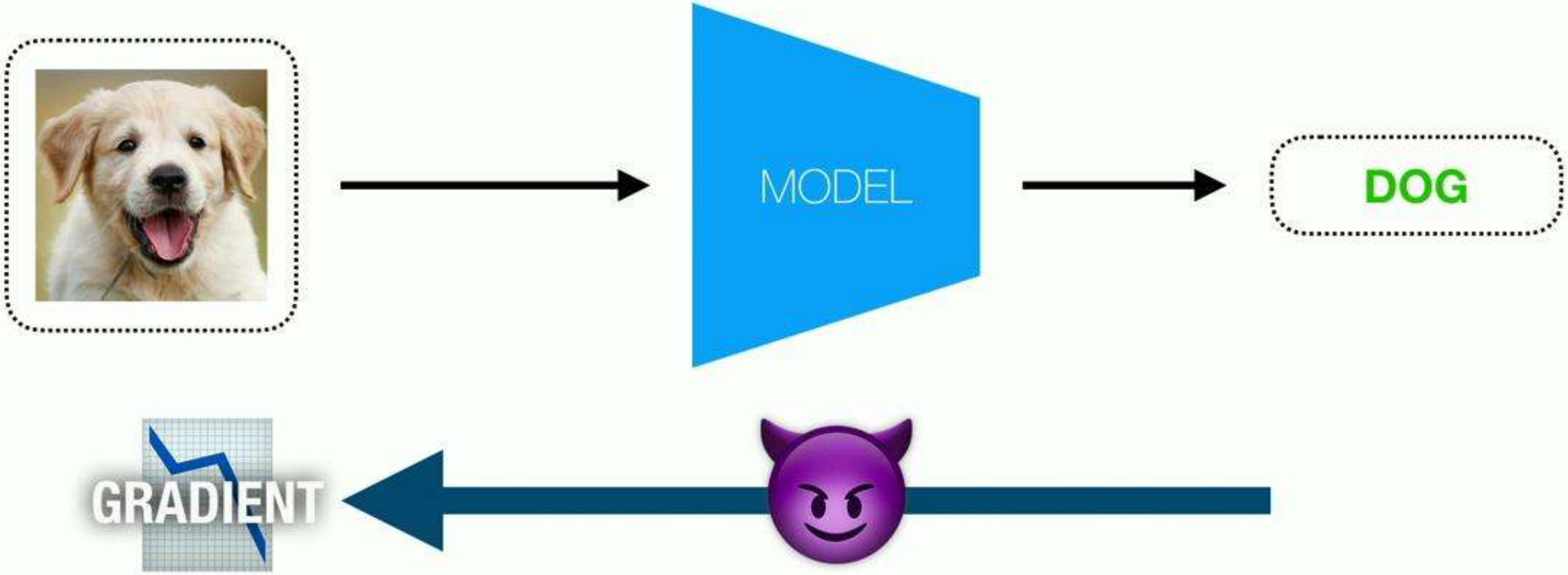
Deep Learning for Image Classification



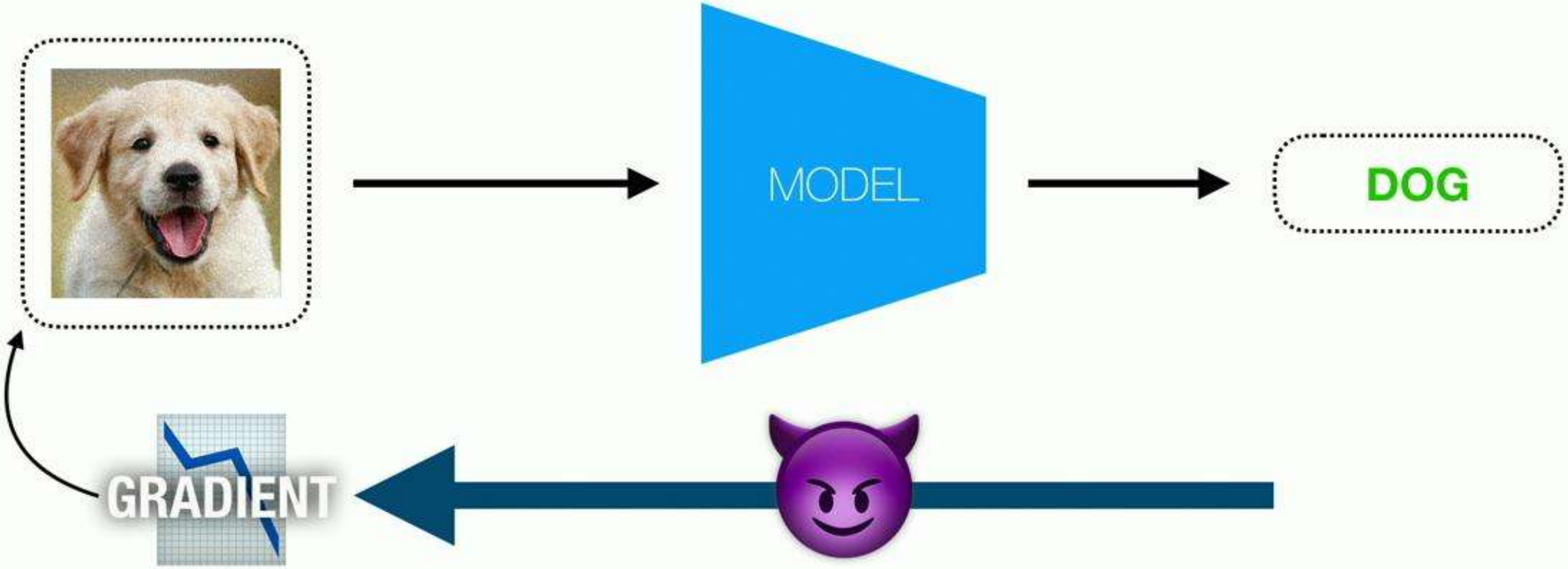
Adversarial Attack on Deep Learning



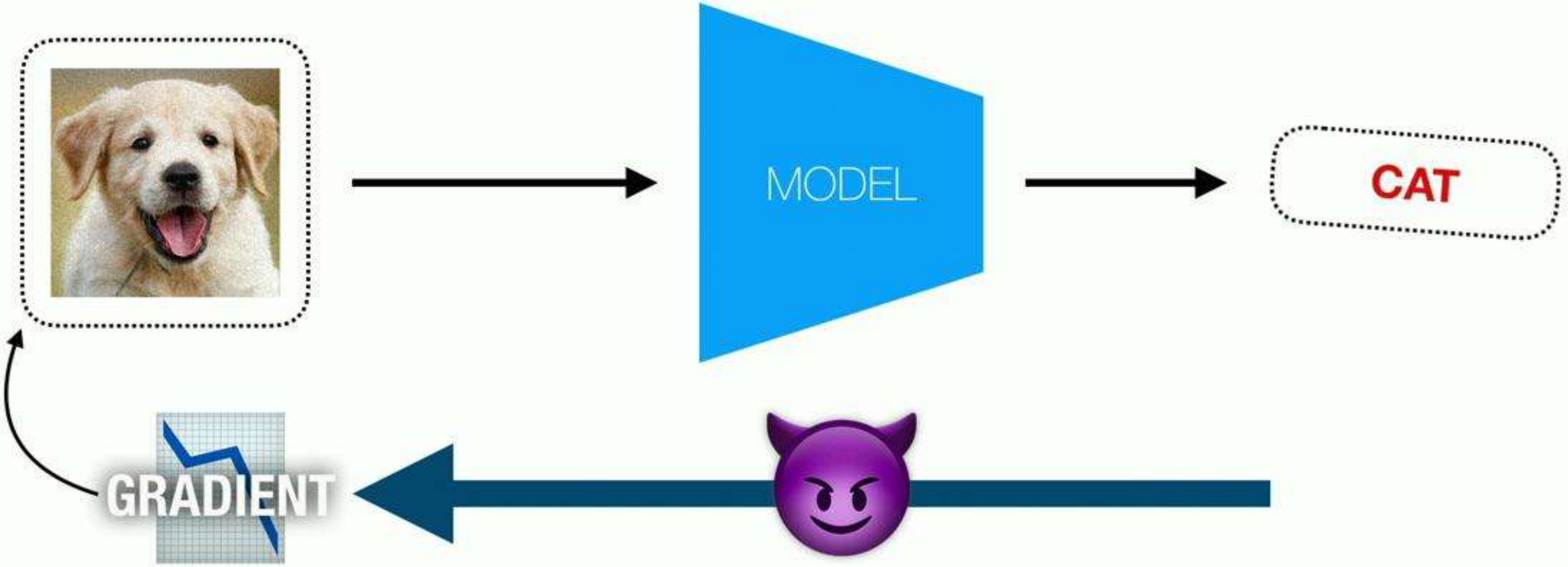
Adversarial Attack on Deep Learning



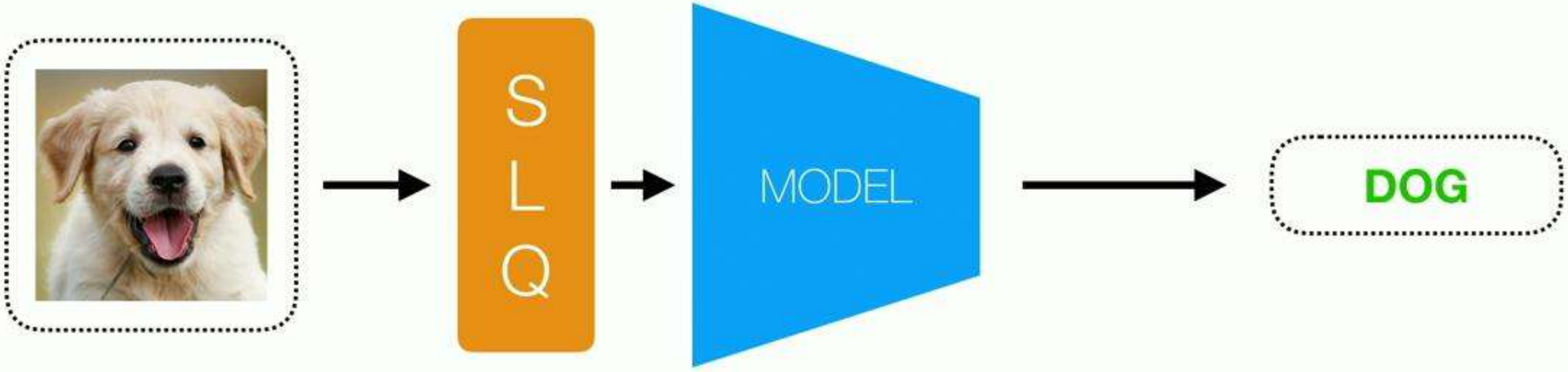
Adversarial Attack on Deep Learning



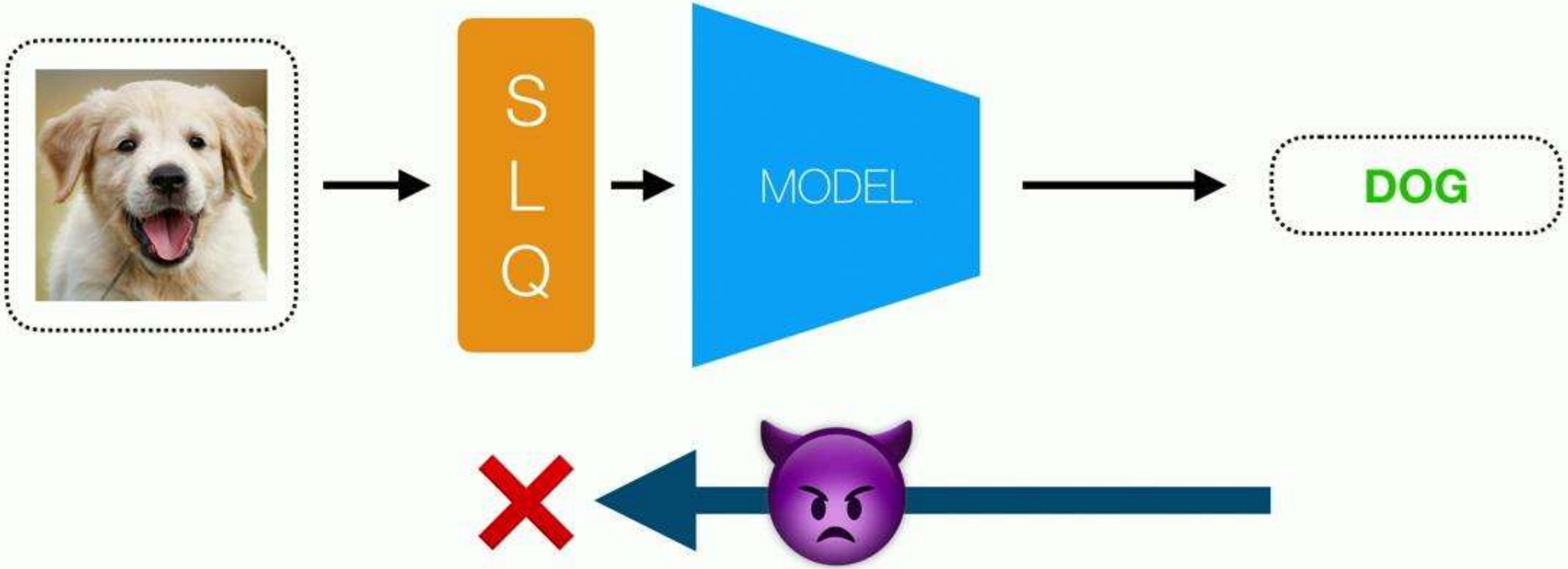
Adversarial Attack on Deep Learning



Stochastic Local Quantization (SLQ)



Stochastic Local Quantization (SLQ)



SLQ leverages JPEG compression

SLQ leverages JPEG compression

JPEQ Quality 80

JPEQ Quality 60

JPEQ Quality 40

JPEQ Quality 20

SLQ applies JPEG compression of a random quality to each 8 x 8 block of the image

* larger blocks shown for presentation

SLQ leverages JPEG compression

JPEQ Quality 80

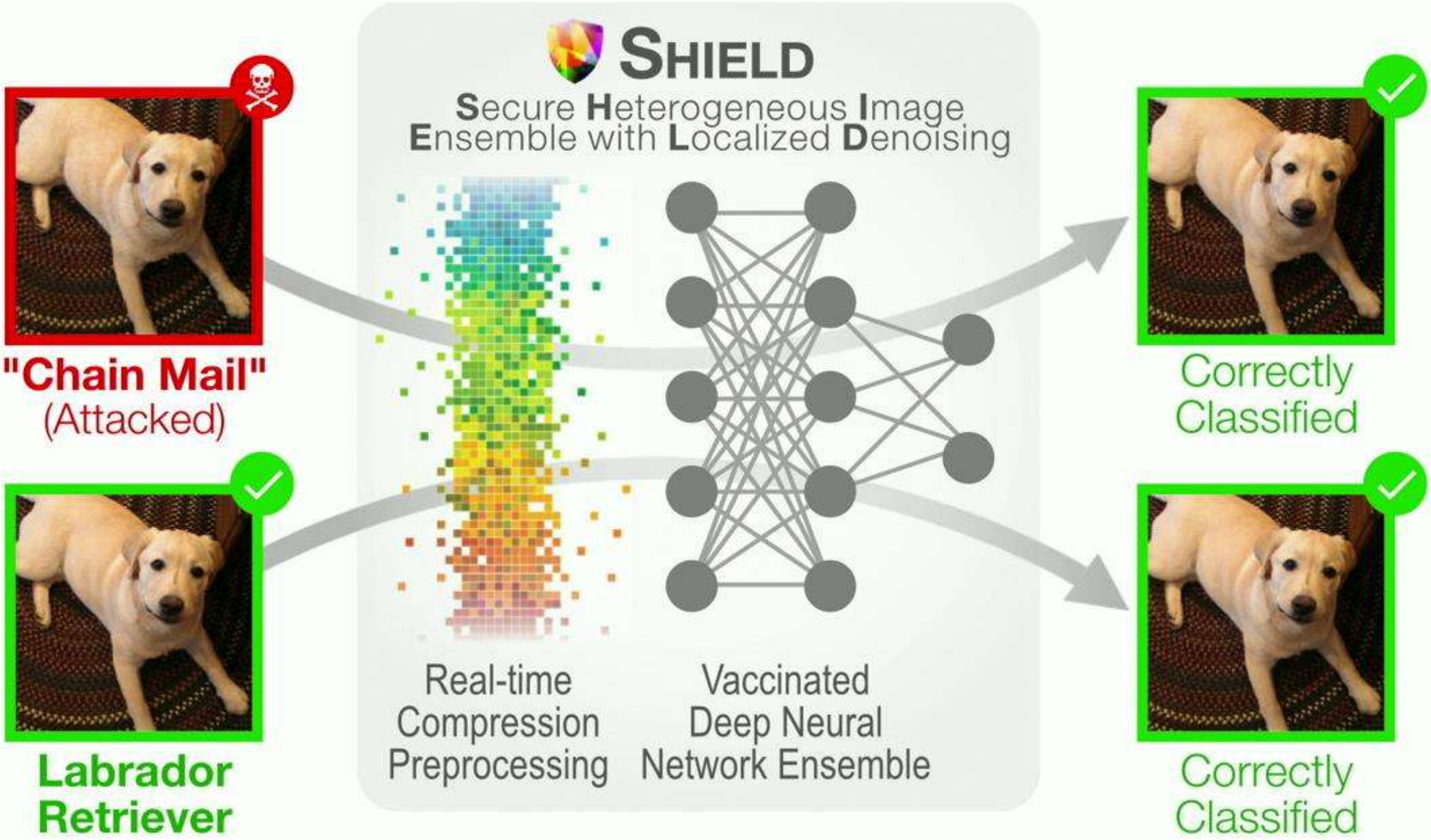
JPEQ Quality 60

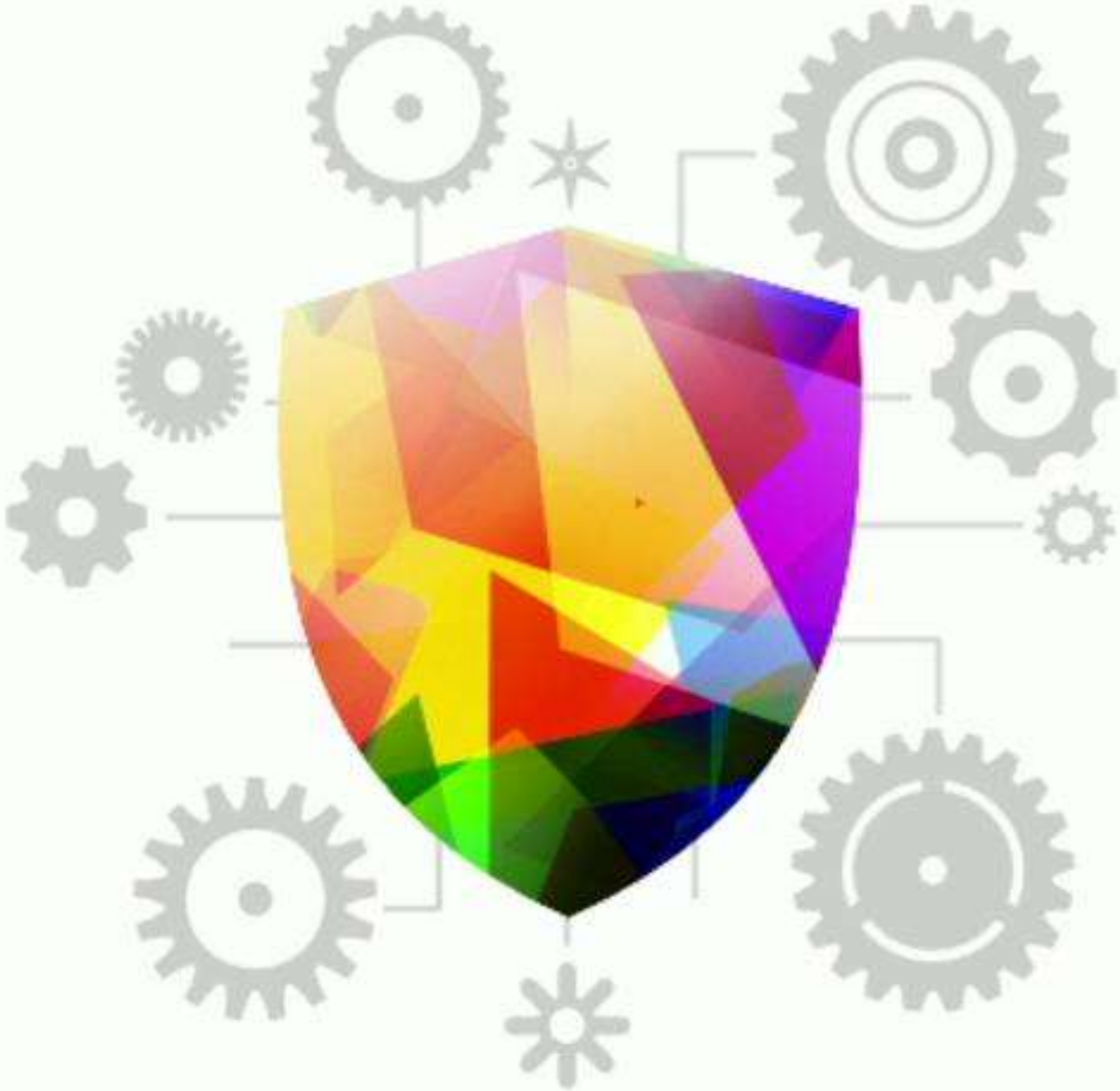
JPEQ Quality 40

JPEQ Quality 20

SLQ applies JPEG compression of a random quality to each 8 x 8 block of the image

* larger blocks shown for presentation


SHIELD
Secure Heterogeneous Image
Ensemble with Localized Denoising
"Chain Mail"
(Attacked)
Labrador
Retriever
Real-time
Compression
Preprocessing
Vaccinated
Deep Neural
Network Ensemble
Correctly
Classified
Correctly
Classified

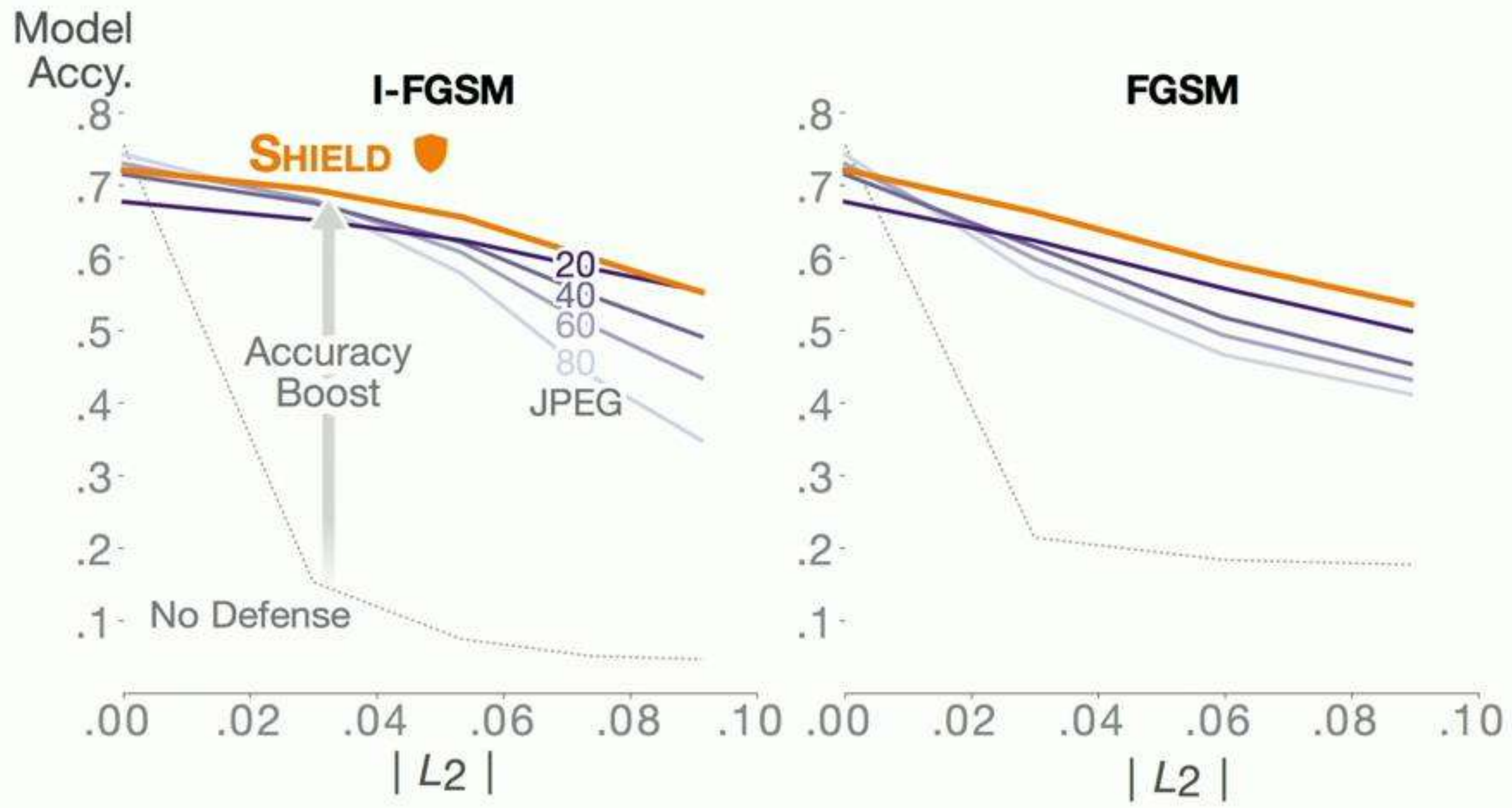


SHIELD is multi-pronged approach that incorporates

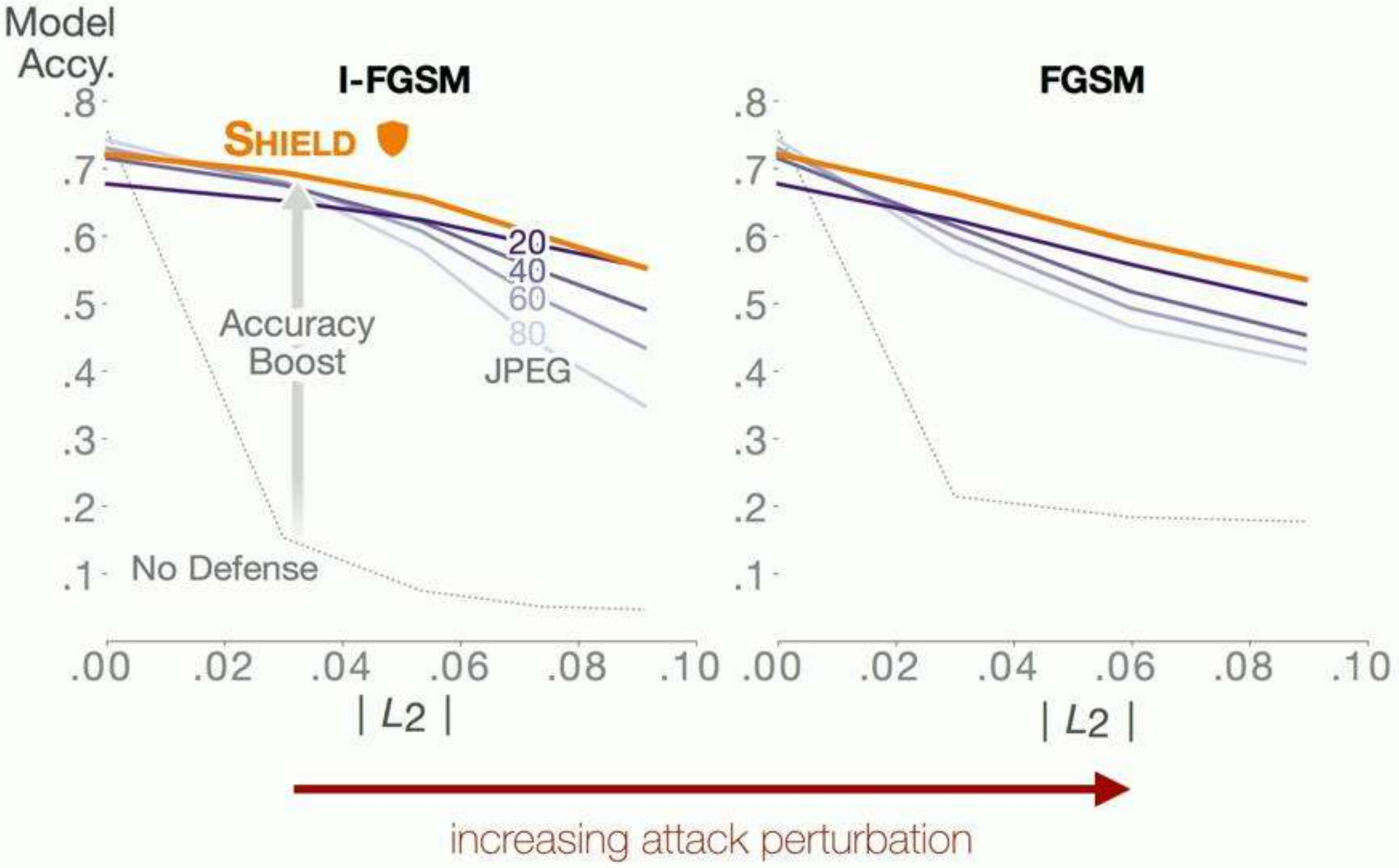
- Stochastic Local Quantization
- Model Vaccination (re-training)
- Ensembling

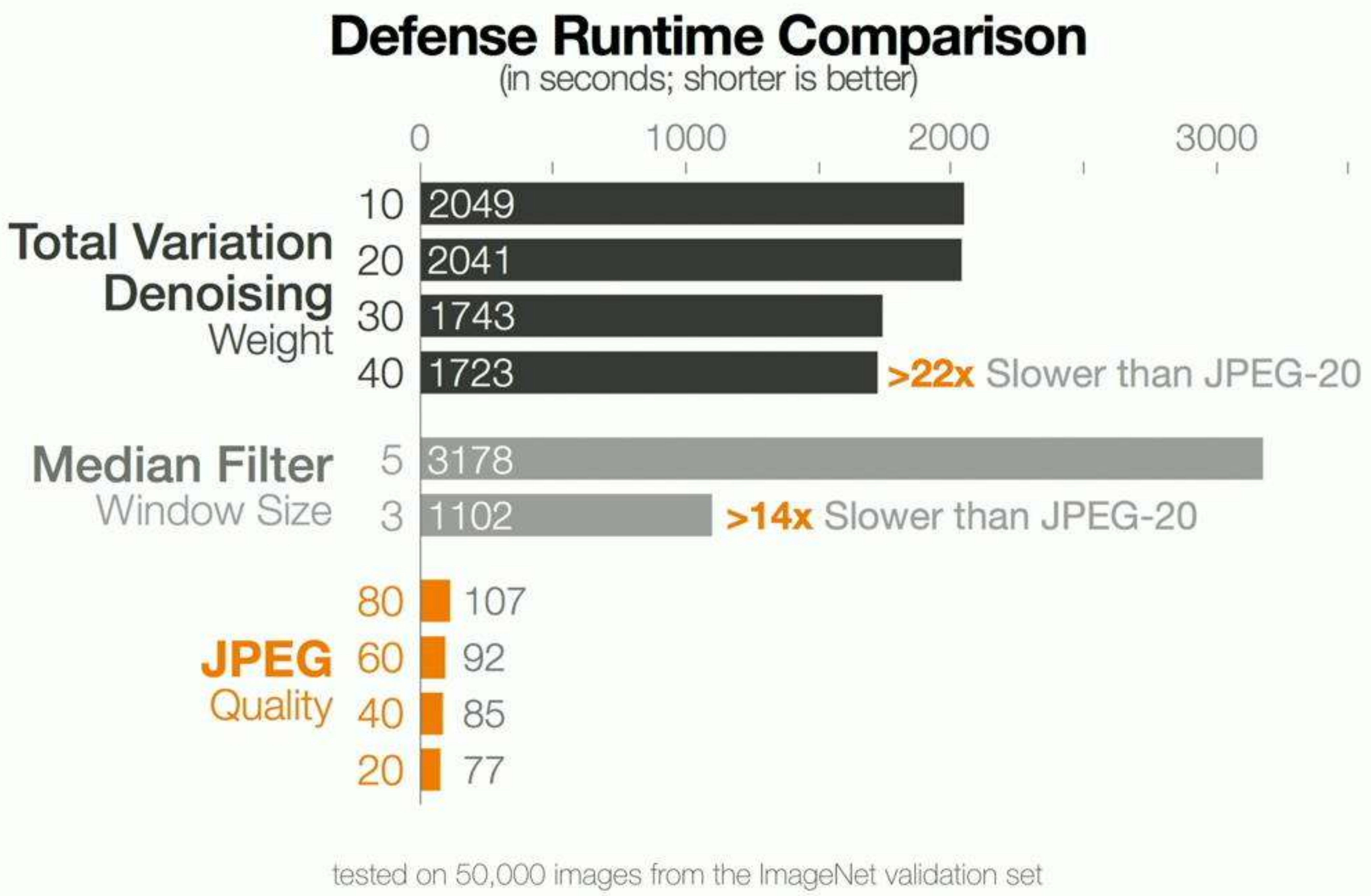
to mitigate adversarial attacks

Results with ResNet-50 v2 (on ImageNet validation set)



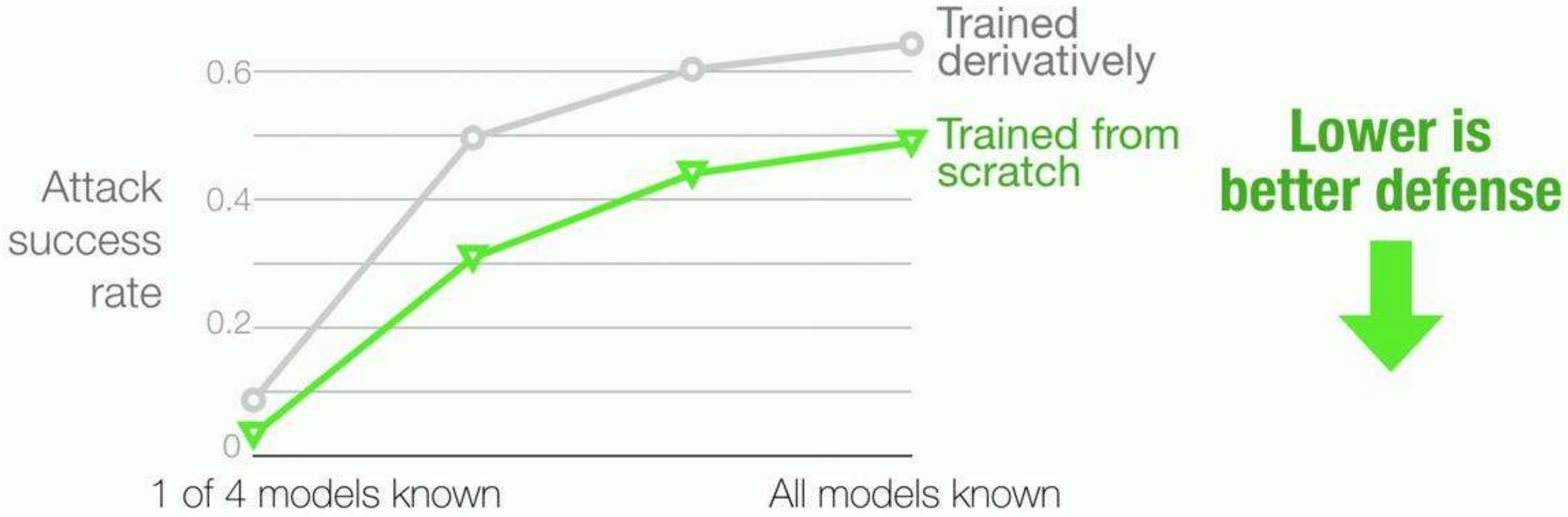
Results with ResNet-50 v2 (on ImageNet validation set)




Defense Runtime Comparison
(in seconds; shorter is better)
0
1000
2000
3000
Total Variation Denoising
Weight
10
2049
20
2041
30
1743
40
1723
>22x Slower than JPEG-20
Median Filter
Window Size
5
3178
3
1102
>14x Slower than JPEG-20
JPEG
Quality
80
107
60
92
40
85
20
77
tested on 50,000 images from the ImageNet validation set

Extending to Adaptive Attacks

SHIELD ensemble with **less correlated** model weights are **more robust** to *targeted* adaptive attacks [KDD'19 LEMINCS]



The Efficacy of SHIELD under Different Threat Models. Cory Cornelius, Nilaksh Das, Shang-Tse Chen, Li Chen, Michael E. Kounavis, Duen Horng (Polo) Chau. *KDD 2019 Workshop on Learning and Mining for Cybersecurity (LEMINCS).*

ADAGIO

ECML-PKDD 2018

Experimentation with Real-time Defense for Speech to Text

Nilaksh Das
GT

Madhuri Shangbogue
GT

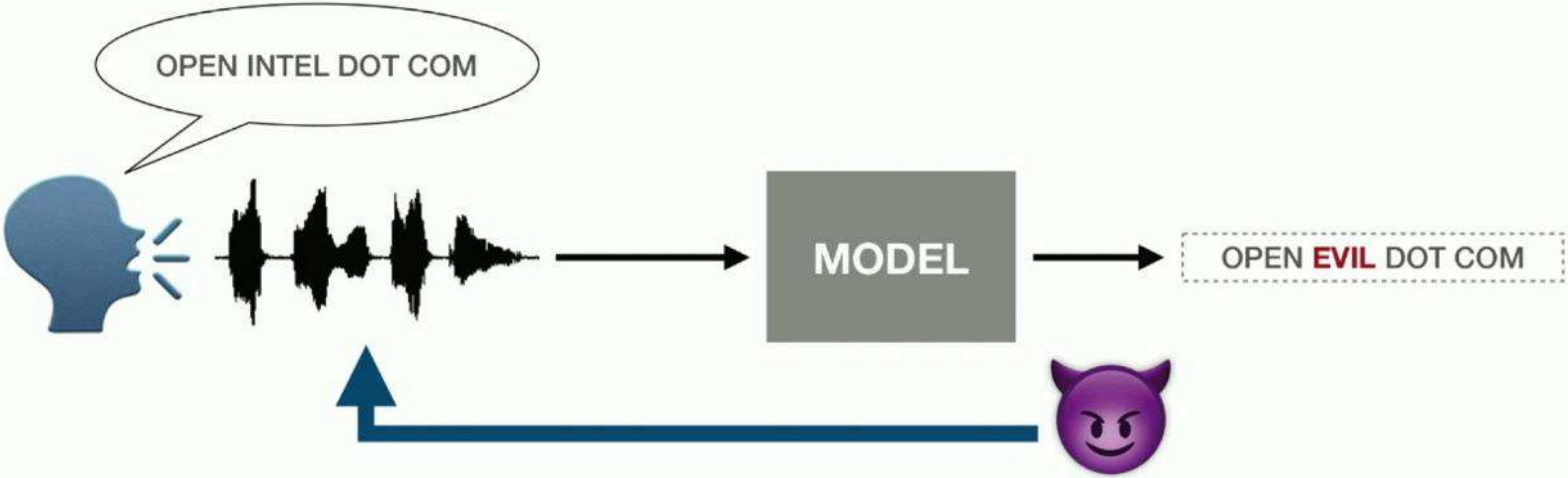
Shang-Tse Chen
GT

Li Chen
Intel

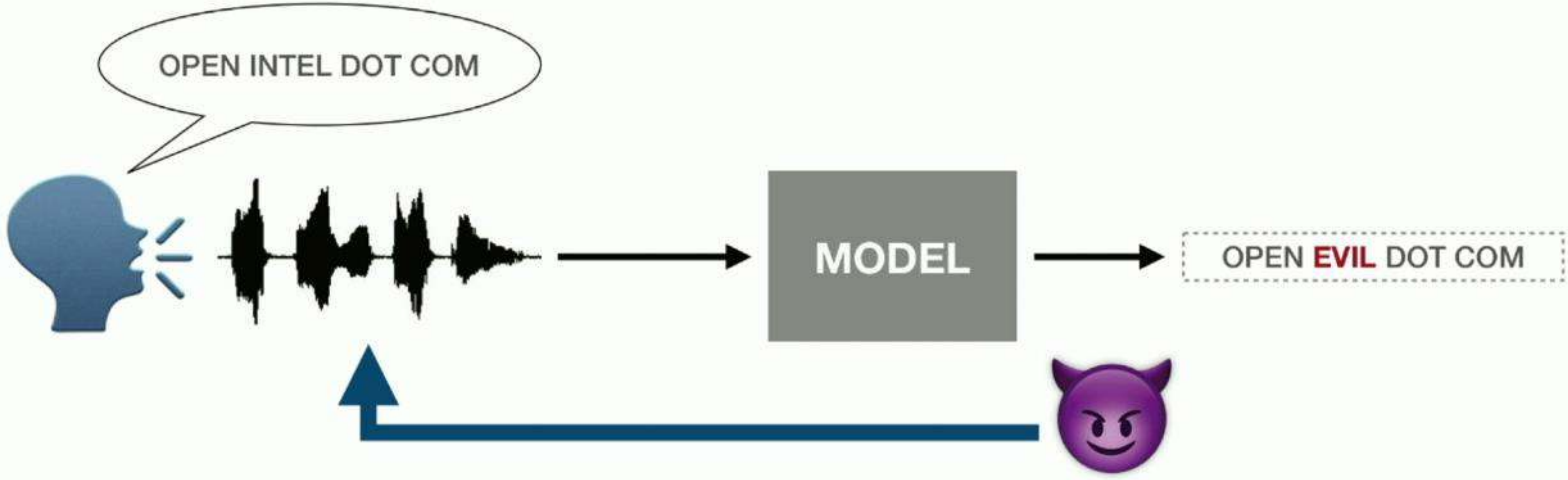
Michael Kounavis
Intel

Polo Chau
GT

Adversarial Attack on Speech-to-text

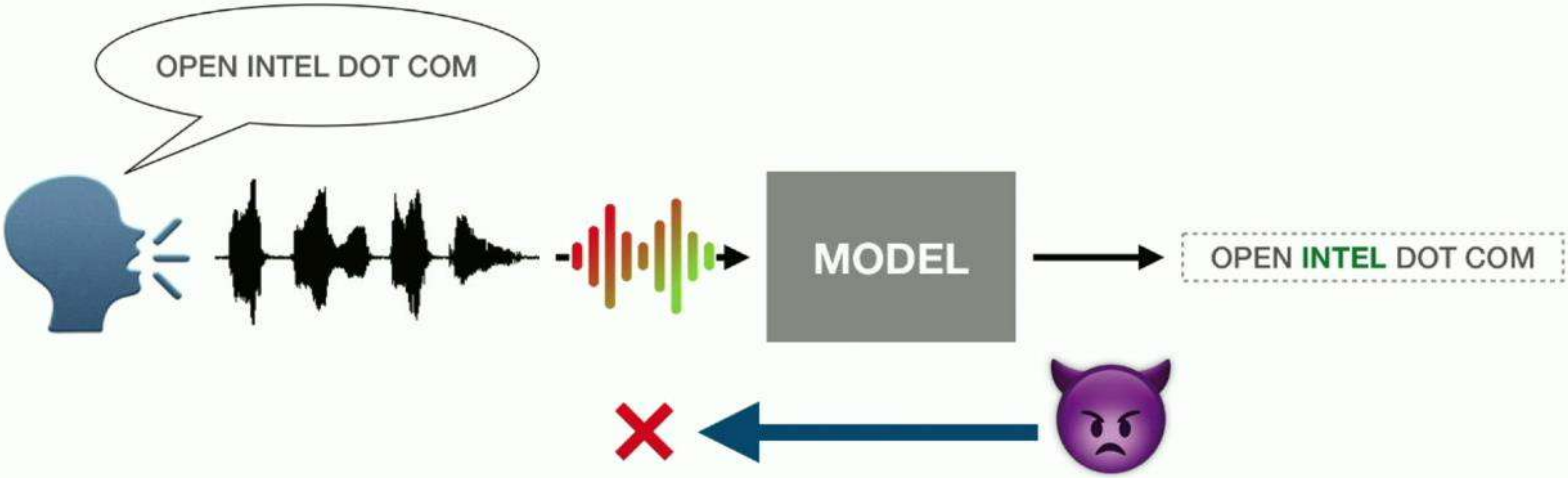


Adversarial Attack on Speech-to-text



An adversary uses backpropagation to attack the model.

Adversarial Attack on Speech-to-Text



ADAGIO incorporates compression as defense, which blocks the gradient to the attacker.

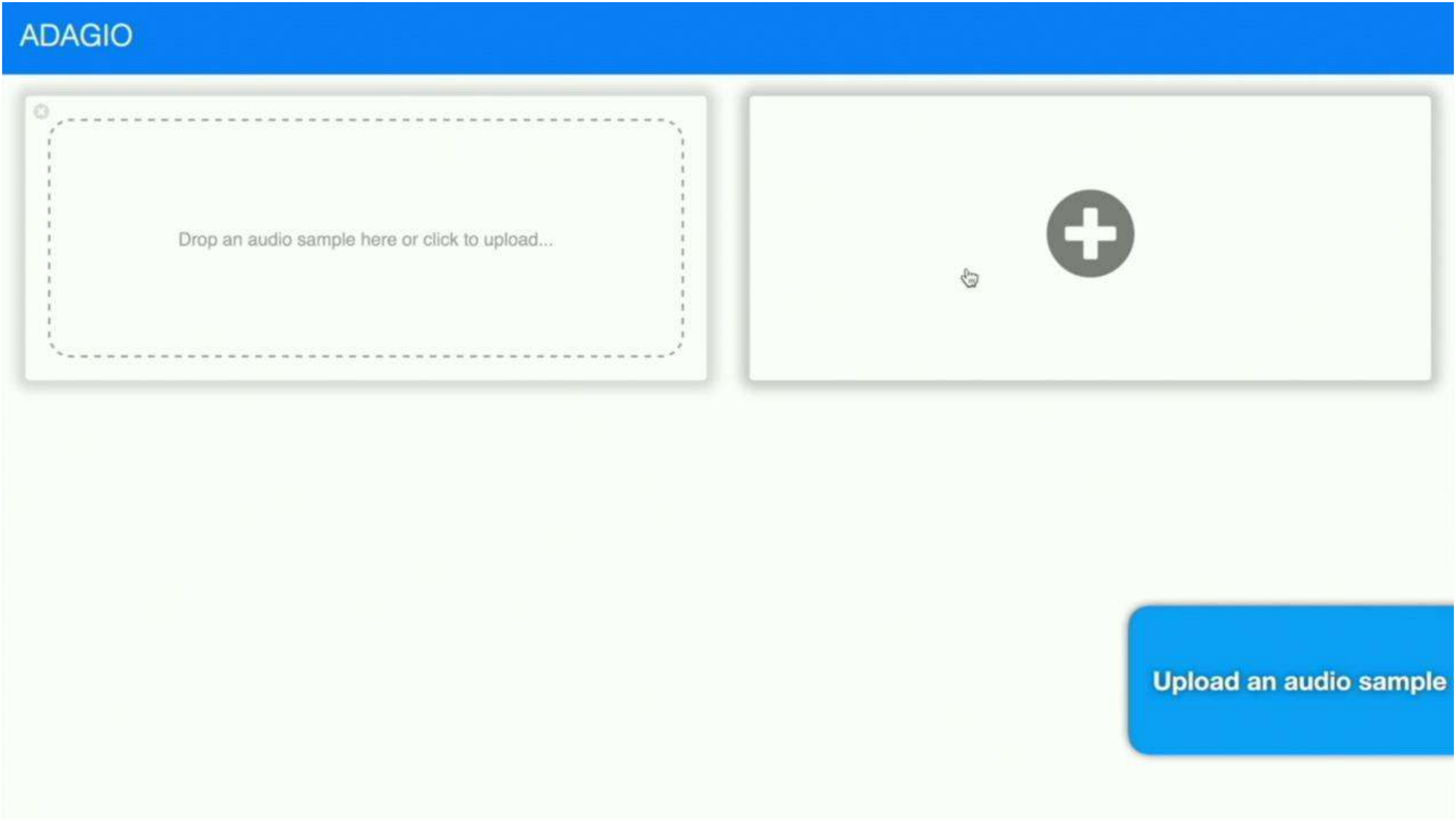
ADAGIO

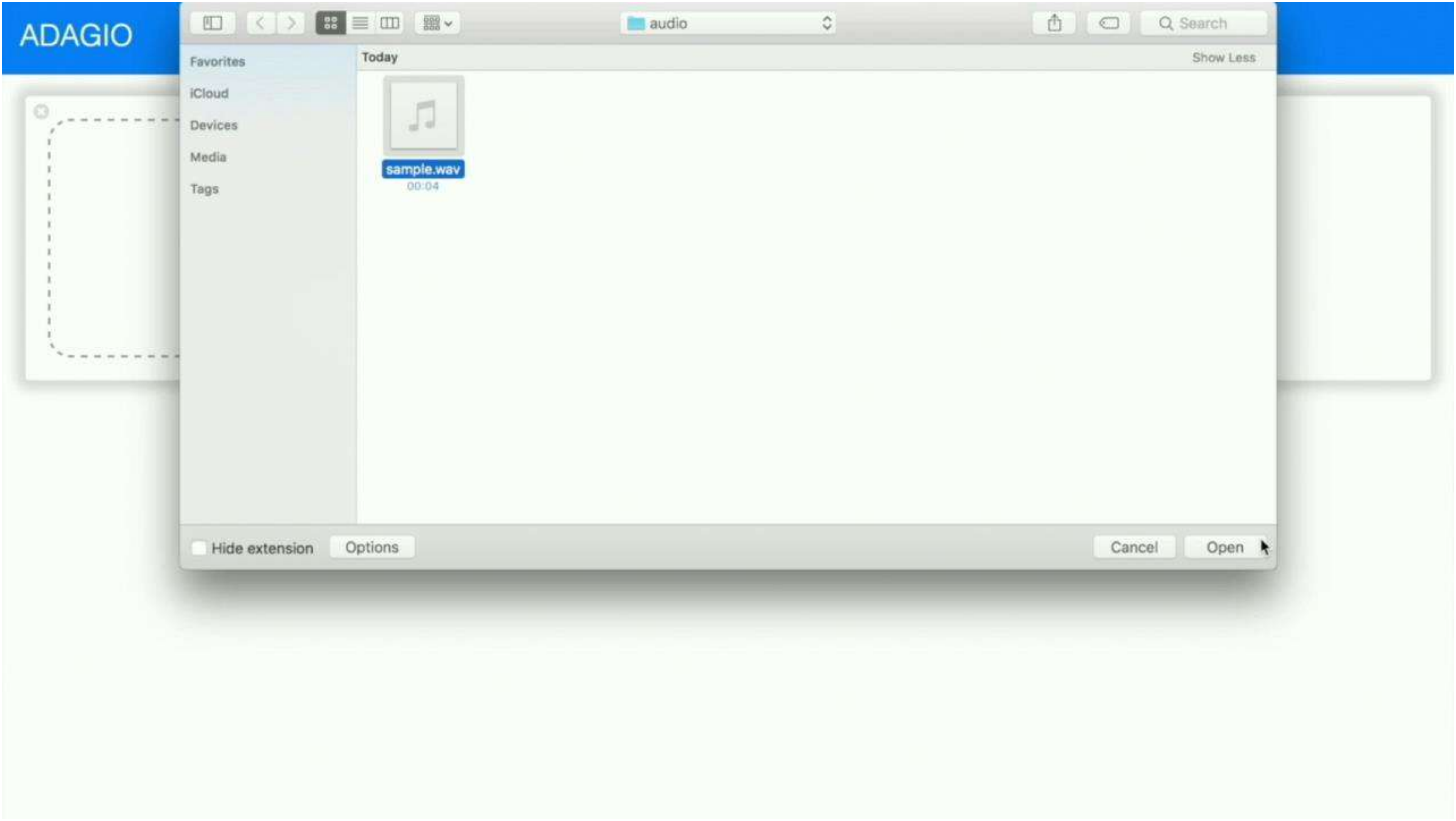
Interactive Experimentation with Adversarial Attack & Defense for Audio

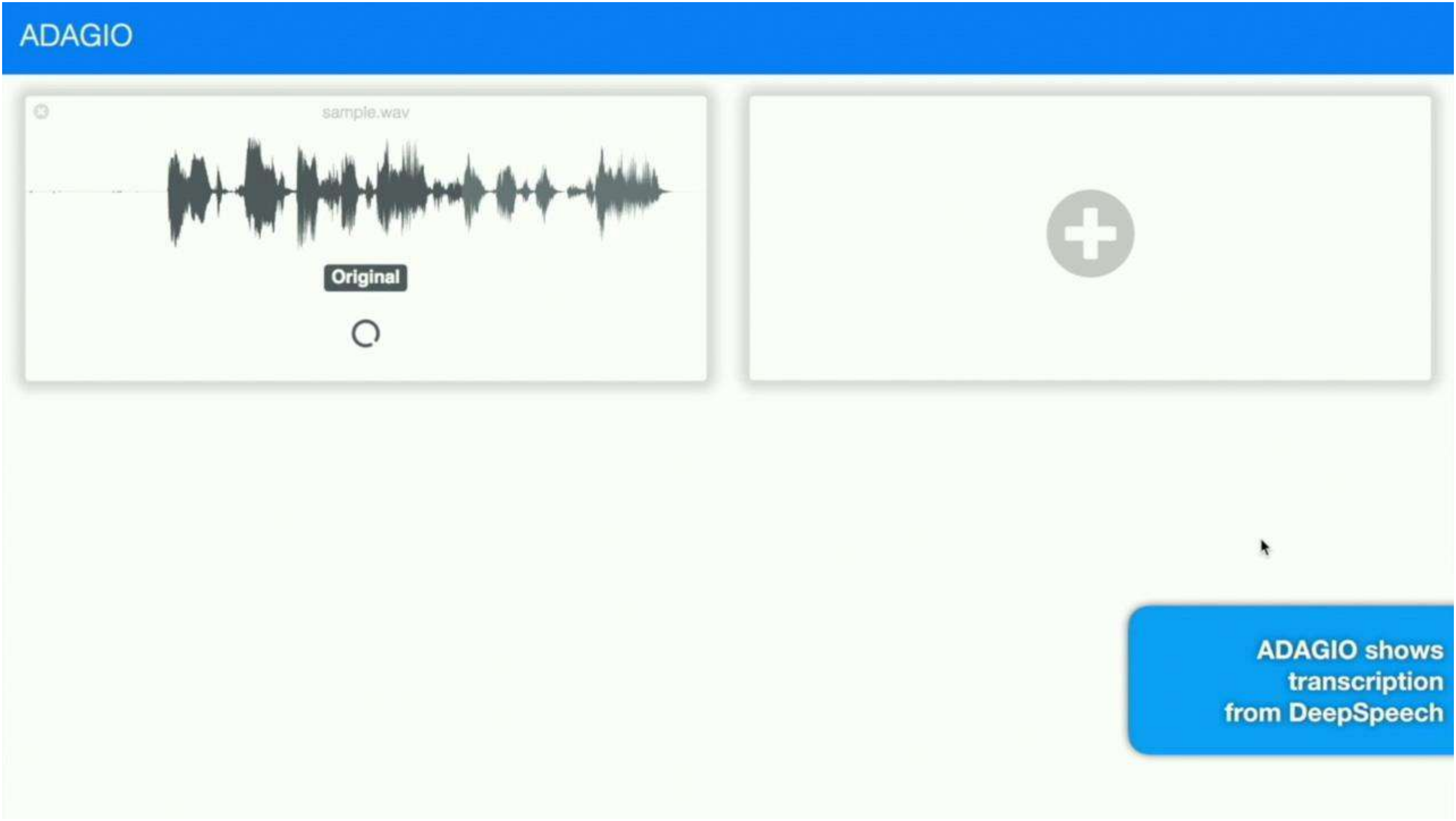
- 🗣 Upload your own audio sample
- ⚔ Perform audio adversarial attack
- 🔰 Apply compression to defend
- ▶️ Play audio, listen for differences

ADAGIO = **A**ttack & **D**efense for **A**udio in a **G**adget with **I**nteractive **O**perations

ADAGIO


ADAGIO
Drop an audio sample here or click to upload...
Upload an audio sample


ADAGIO
audio
Search
Favorites
iCloud
Devices
Media
Tags
Today
Show Less
sample.wav
00:04
Hide extension
Options
Cancel
Open


ADAGIO
sample.wav
Original
ADAGIO shows transcription from DeepSpeech

ADAGIO

sample.wav

Original

isn't the party also to announce his engagement to joanna

ADAGIO shows transcription from DeepSpeech

Secure AI

Attack & Defense of Deep Neural Networks

ShapeShifter - Physical Adversarial Attack

SHIELD - Real-time Defense for *Images*

Do-it-yourself Adversarial ML

ADAGIO - Experimentation with Real-time Defense for *Audio*

MLsploit - Interactive Experimentation with Adversarial ML

MLsploit

github.com/mlsploit

A Framework for Interactive Experimentation with Adversarial Machine Learning Research

Contributors from *Intel Science and Technology Center for Adversary-Resilient Security Analytics*:
Nilaksh Das, Siwei Li, Chanil Jeon, Jinho Jung*, Shang-Tse Chen*, Carter Yagemann*, Evan Downing*, Haekyu Park, Evan Yang, Li Chen, Michael Kounavis, Ravi Sahita, David Durham, Scott Buck, Polo Chau, Taesoo Kim, Wenke Lee
(*equal contribution)

[BlackHat Asia '19, KDD'19 Showcase]

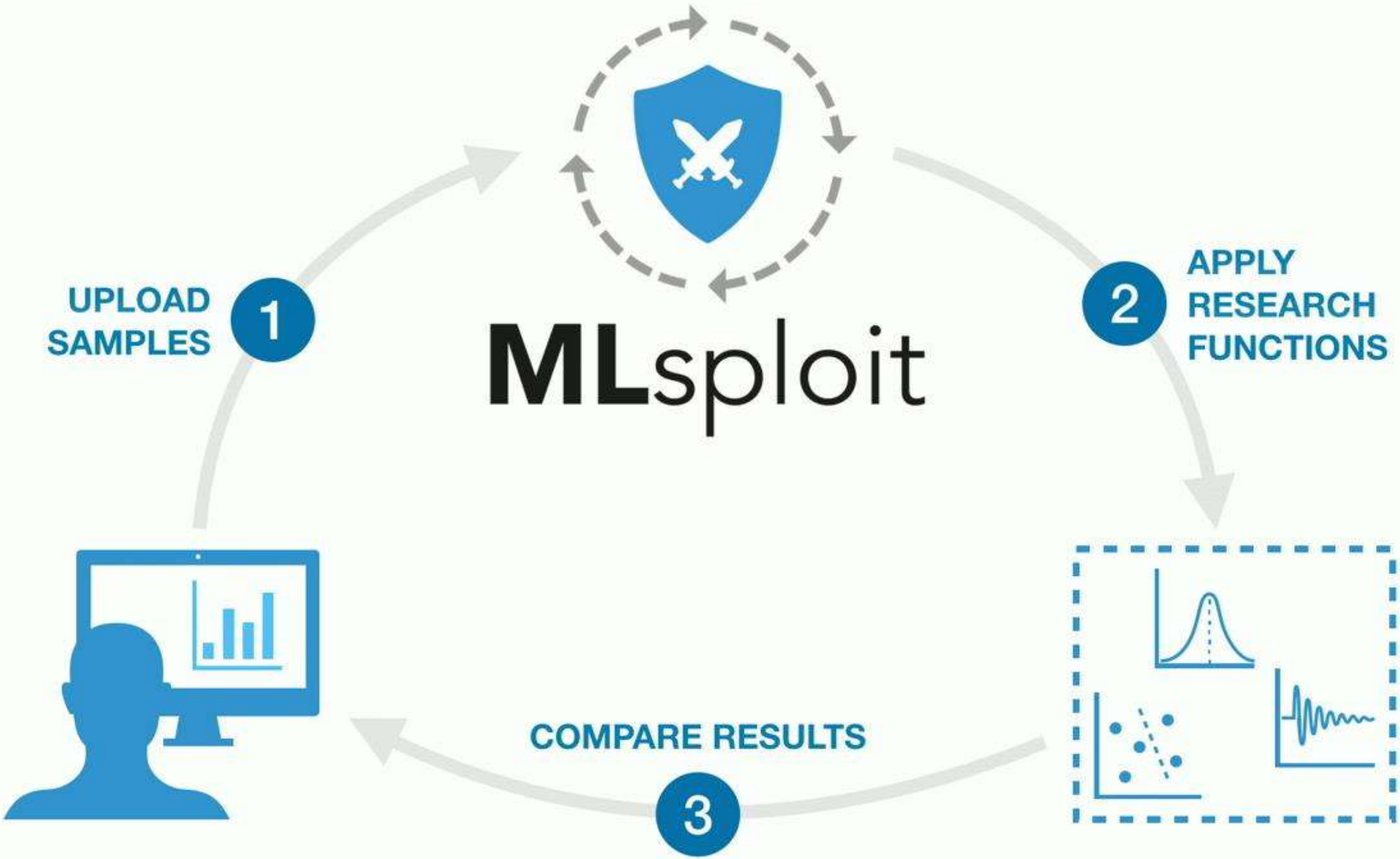
MLsploit

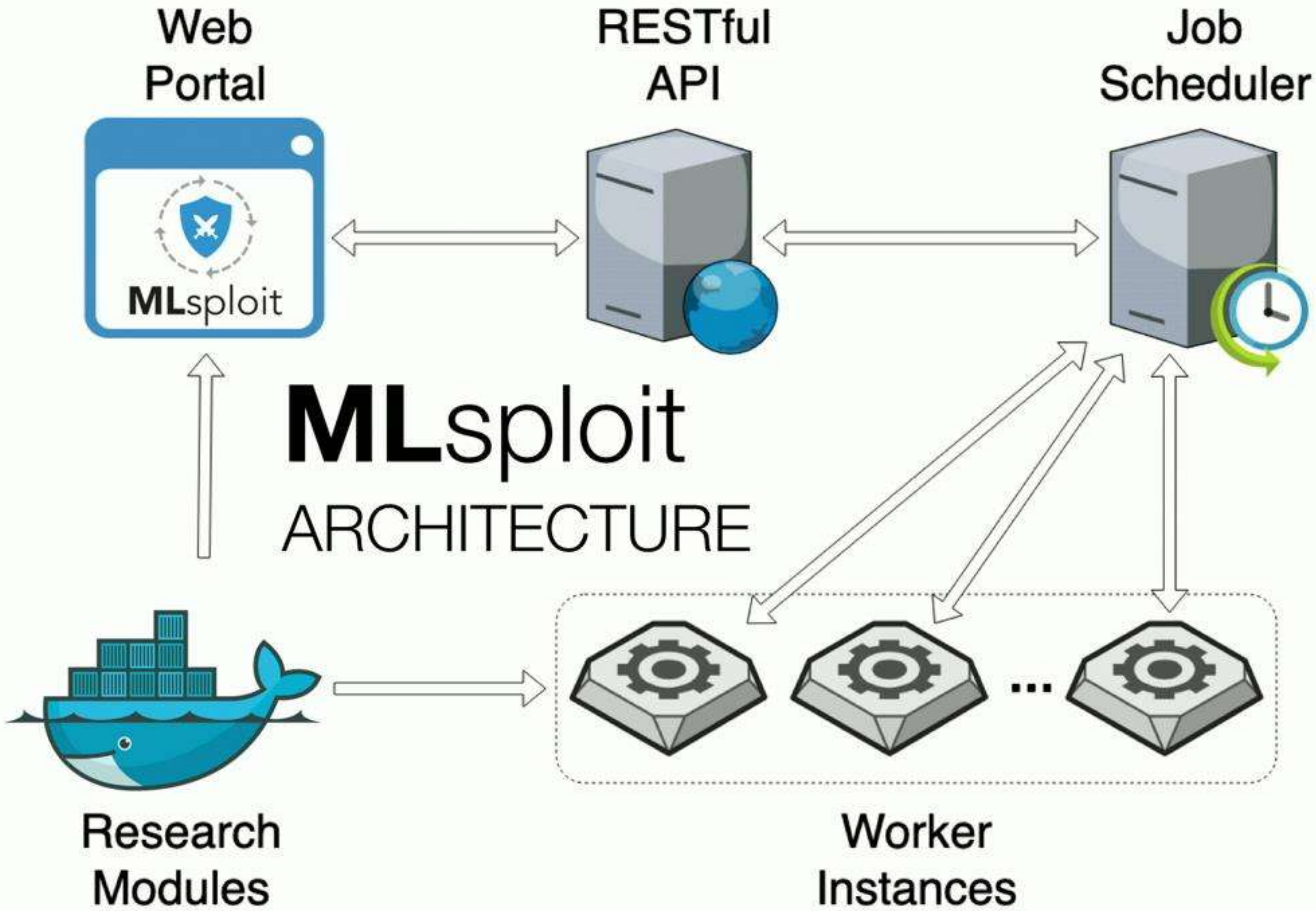
- **Research modules** for adversarial ML
  - Enables **comparison** of attacks and defenses
- **Interactive experimentation** with ML research
- Researchers can **easily integrate** novel research into an intuitive and seamless **user interface**

MLsploit

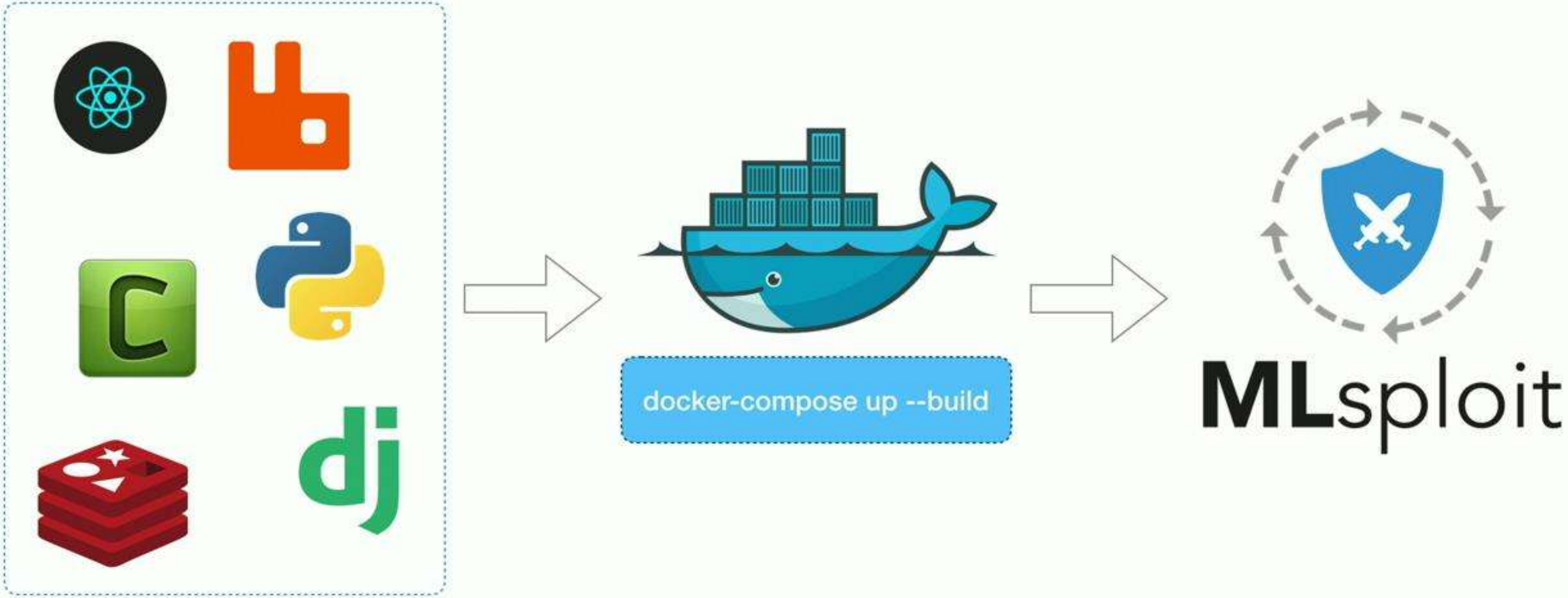
- ★ **AVPass** (leaking and bypassing Android malware detection systems)
- ★ **ELF** (bypassing Linux malware detection with API perturbation)
- ★ **PE** (create and attack ML models for detecting Windows PE malware)
- ★ **Intel®-Software Guard Extensions**

  (privacy preserving adversarial ML as a service)
- ★ **SHIELD** (attack and defend state-of-the-art image classification models)
  - ✳ Attacks: **FGSM, DeepFool, Carlini-Wagner**
  - ✳ Defenses: **SLQ, JPEG, Median Filter, TV-Bregman**

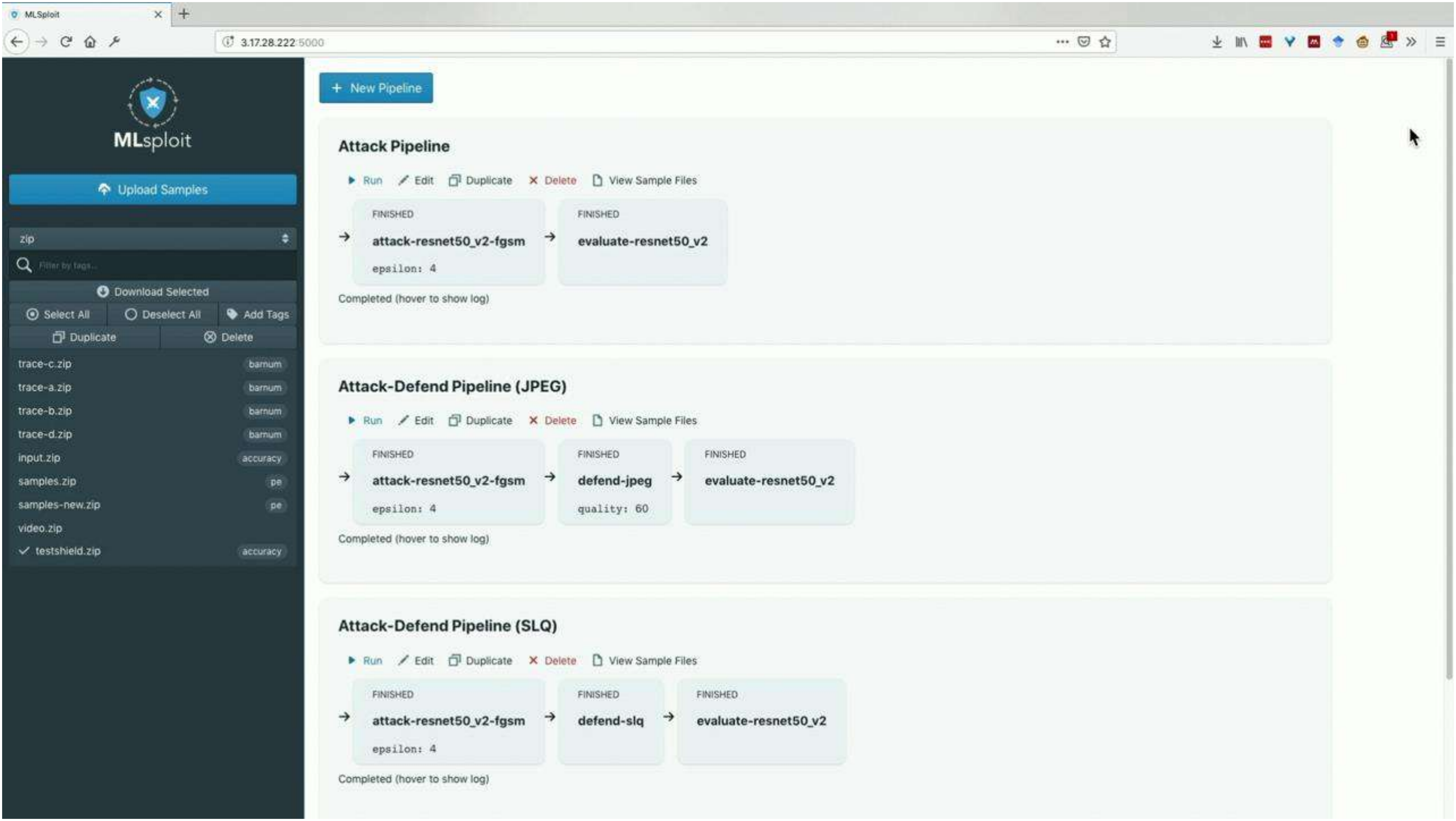

MLsploit
UPLOAD SAMPLES
1
APPLY RESEARCH FUNCTIONS
2
COMPARE RESULTS
3

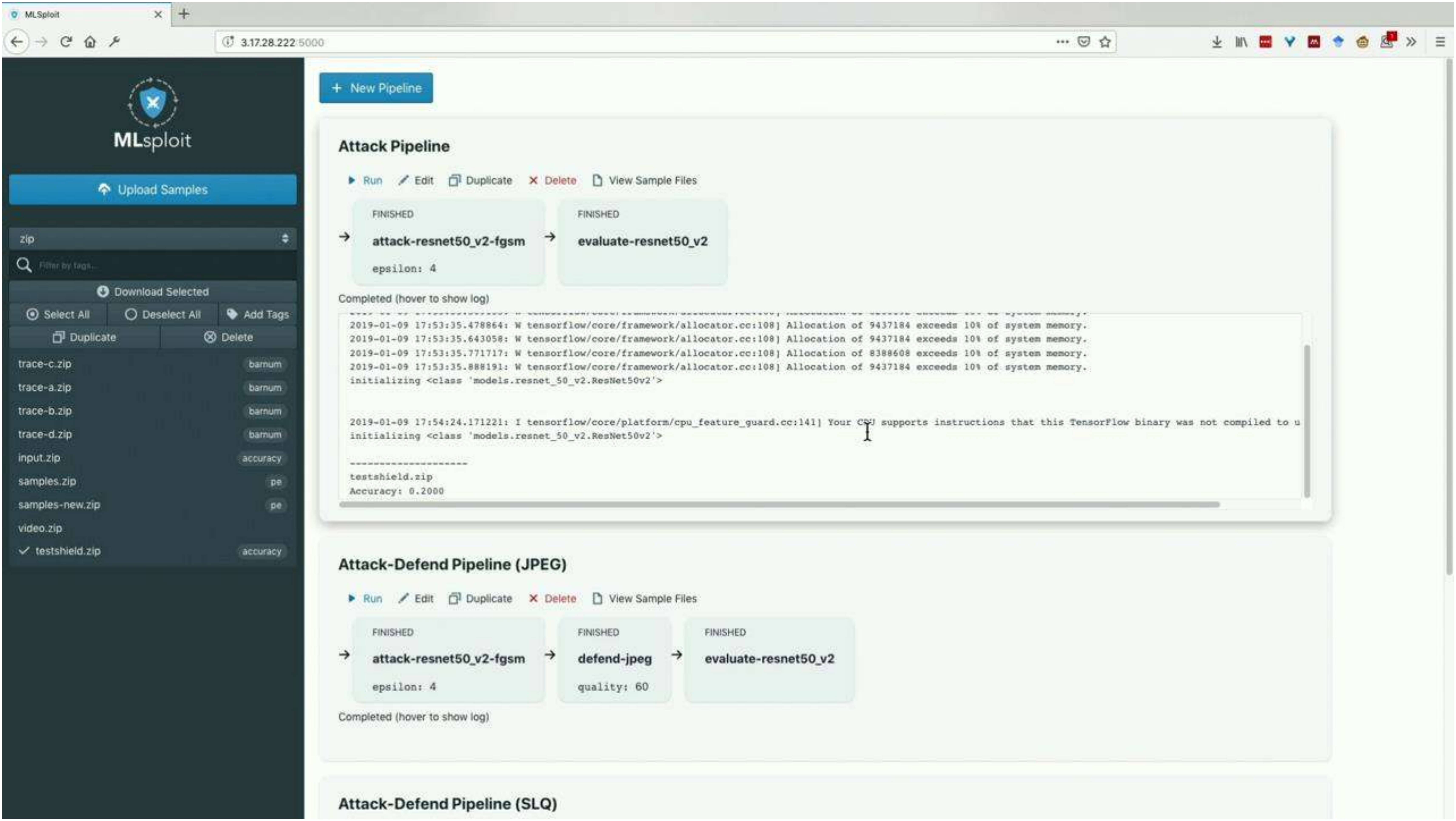

Web
Portal
MLsploit
RESTful
API
Job
Scheduler
MLsploit
ARCHITECTURE
...
Research
Modules
Worker
Instances

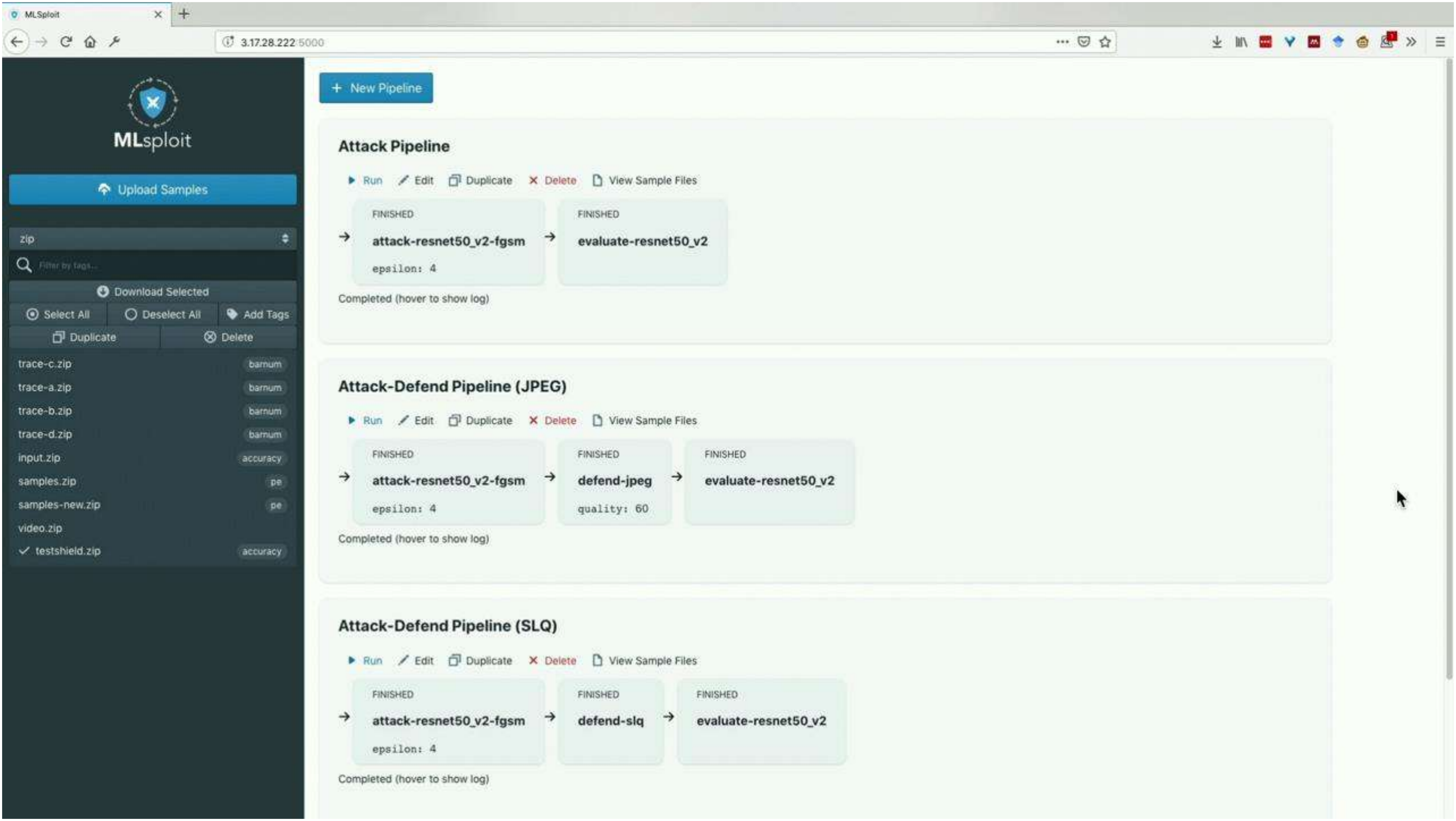
ONE-STEP INSTALLATION

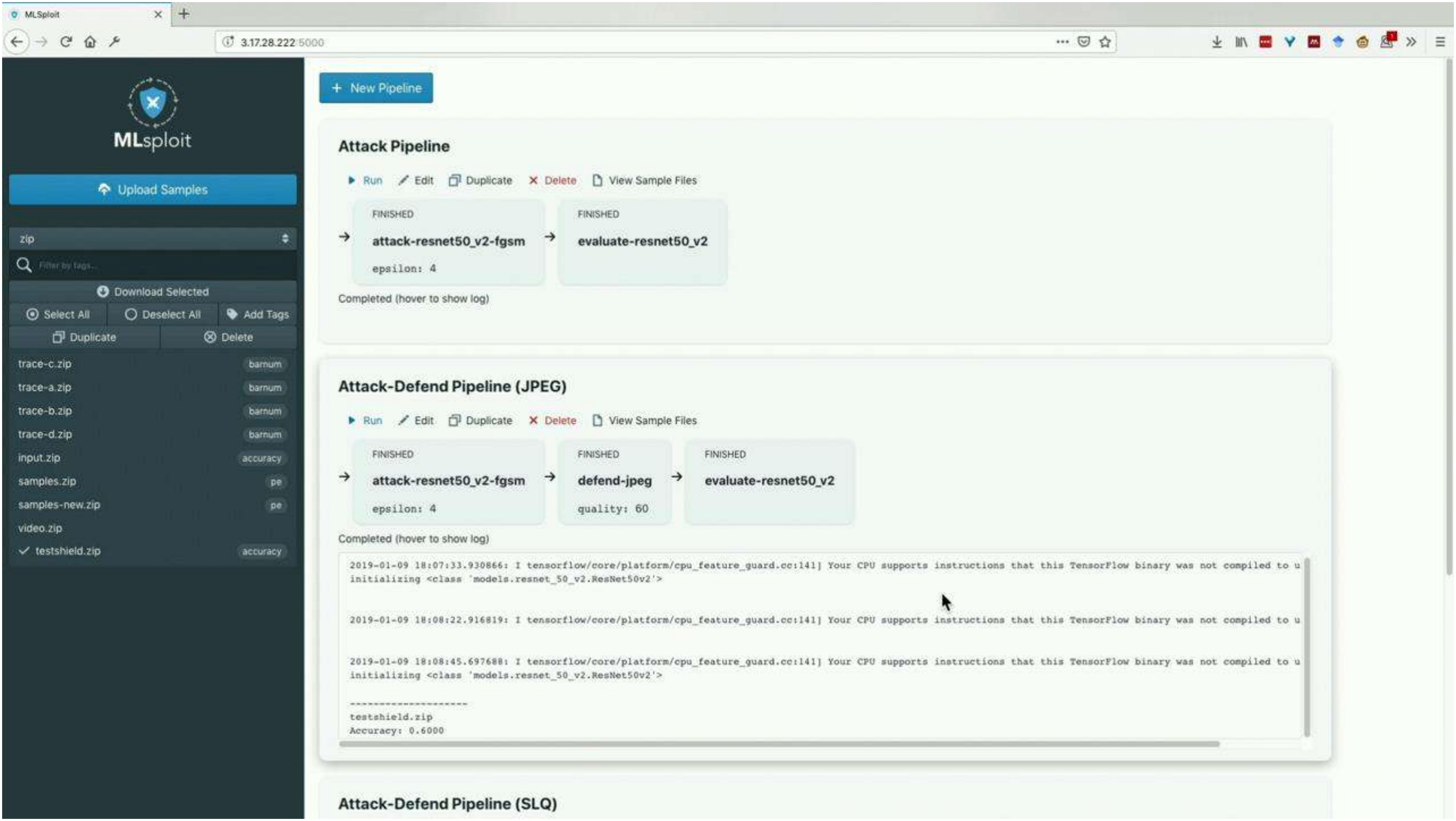


EASY INTEGRATION OF RESEARCH


MLSploit
3.17.28.222:5000
MLsploit
Upload Samples
zip
Filter by tags...
Download Selected
Select All
Deselect All
Add Tags
Duplicate
Delete
trace-c.zip
barnum
trace-a.zip
barnum
trace-b.zip
barnum
trace-d.zip
barnum
input.zip
accuracy
samples.zip
pe
samples-new.zip
pe
video.zip
testshield.zip
accuracy
+ New Pipeline
Attack Pipeline
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
Attack-Defend Pipeline (JPEG)
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
defend-jpeg
quality: 60
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
Attack-Defend Pipeline (SLQ)
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
defend-slq
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)


MLSploit
3.17.28.222:5000
MLsploit
Upload Samples
zip
Download Selected
Select All
Deselect All
Add Tags
Duplicate
Delete
trace-c.zip barnum
trace-a.zip barnum
trace-b.zip barnum
trace-d.zip barnum
input.zip accuracy
samples.zip pe
samples-new.zip pe
video.zip
testshield.zip accuracy
+ New Pipeline
Attack Pipeline
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
2019-01-09 17:53:35.478864: W tensorflow/core/framework/allocator.cc:108] Allocation of 9437184 exceeds 10% of system memory.
2019-01-09 17:53:35.643058: W tensorflow/core/framework/allocator.cc:108] Allocation of 9437184 exceeds 10% of system memory.
2019-01-09 17:53:35.771717: W tensorflow/core/framework/allocator.cc:108] Allocation of 8388608 exceeds 10% of system memory.
2019-01-09 17:53:35.888191: W tensorflow/core/framework/allocator.cc:108] Allocation of 9437184 exceeds 10% of system memory.
initializing <class 'models.resnet_50_v2.ResNet50v2'>
2019-01-09 17:54:24.171221: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to u
initializing <class 'models.resnet_50_v2.ResNet50v2'>
----
testshield.zip
Accuracy: 0.2000
Attack-Defend Pipeline (JPEG)
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
defend-jpeg
quality: 60
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
Attack-Defend Pipeline (SLQ)

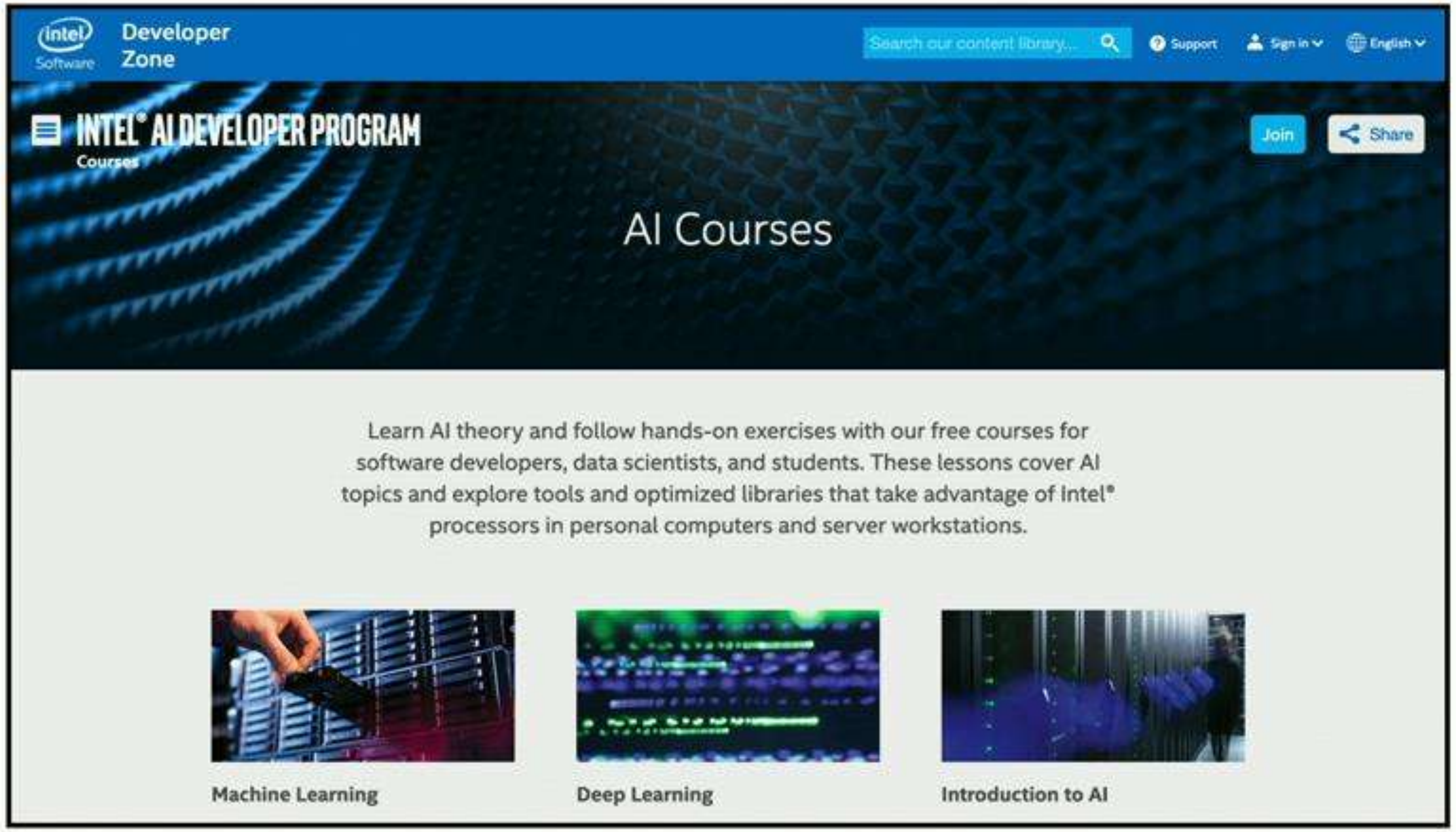

MLSploit
3.17.28.222:5000
MLsploit
Upload Samples
zip
Filter by tags...
Download Selected
Select All
Deselect All
Add Tags
Duplicate
Delete
trace-c.zip barnum
trace-a.zip barnum
trace-b.zip barnum
trace-d.zip barnum
input.zip accuracy
samples.zip pe
samples-new.zip pe
video.zip
testshield.zip accuracy
+ New Pipeline
Attack Pipeline
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
Attack-Defend Pipeline (JPEG)
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
defend-jpeg
quality: 60
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
Attack-Defend Pipeline (SLQ)
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
defend-slq
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)


MLSploit
3.17.28.222:5000
MLsploit
Upload Samples
zip
Filter by tags...
Download Selected
Select All
Deselect All
Add Tags
Duplicate
Delete
trace-c.zip
barnum
trace-a.zip
barnum
trace-b.zip
barnum
trace-d.zip
barnum
input.zip
accuracy
samples.zip
pe
samples-new.zip
pe
video.zip
testshield.zip
accuracy
+ New Pipeline
Attack Pipeline
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
Attack-Defend Pipeline (JPEG)
Run
Edit
Duplicate
Delete
View Sample Files
FINISHED
attack-resnet50_v2-fgsm
epsilon: 4
FINISHED
defend-jpeg
quality: 60
FINISHED
evaluate-resnet50_v2
Completed (hover to show log)
2019-01-09 18:07:33.930866: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to u
initializing <class 'models.resnet_50_v2.ResNet50v2'>
2019-01-09 18:08:22.916819: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to u
2019-01-09 18:08:45.697688: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to u
initializing <class 'models.resnet_50_v2.ResNet50v2'>
----
testshield.zip
Accuracy: 0.6000
Attack-Defend Pipeline (SLQ)

Intel® AI Courses

MLsploit

github.com/mlsploit



software.intel.com/en-us/ai/courses

Secure AI

Attack & Defense (DNN)

ShapeShifter

SHIELD

Do-it-yourself Adversarial ML

ADAGIO

MLsploit

Interpretable AI

Understand Industry Models

ActiVis

Interactive Learning (Education)

GAN Lab

Research landscape

Survey, Gamut

Practitioners very interested in **why** and **how** AI works

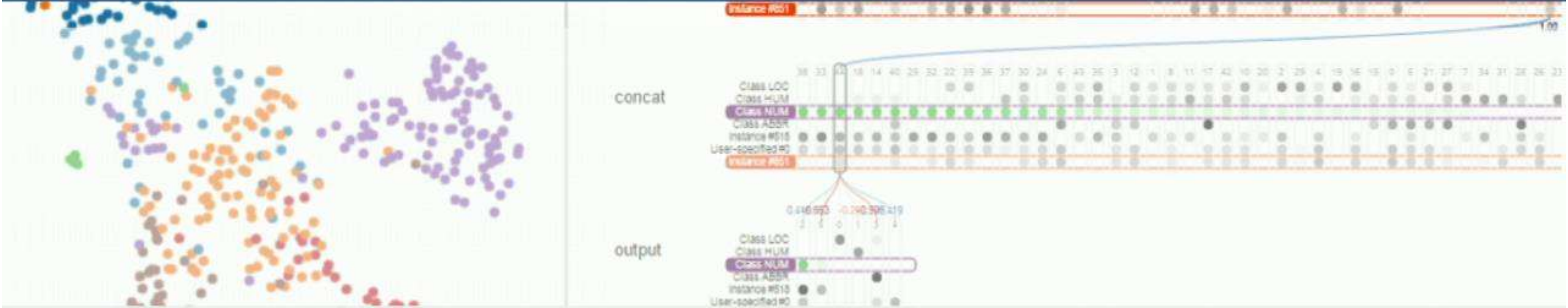
Figure from Washington Post

OUR KEY IDEA

Scalable Interactive visualization

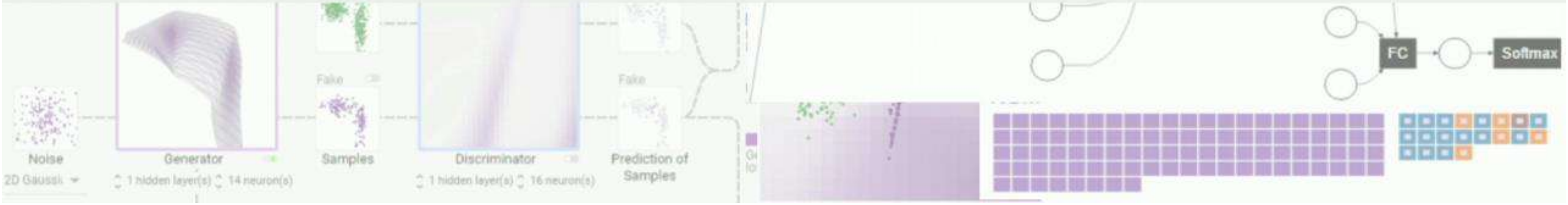
as a medium for connecting users with ML models

Why **interactive visualization**?



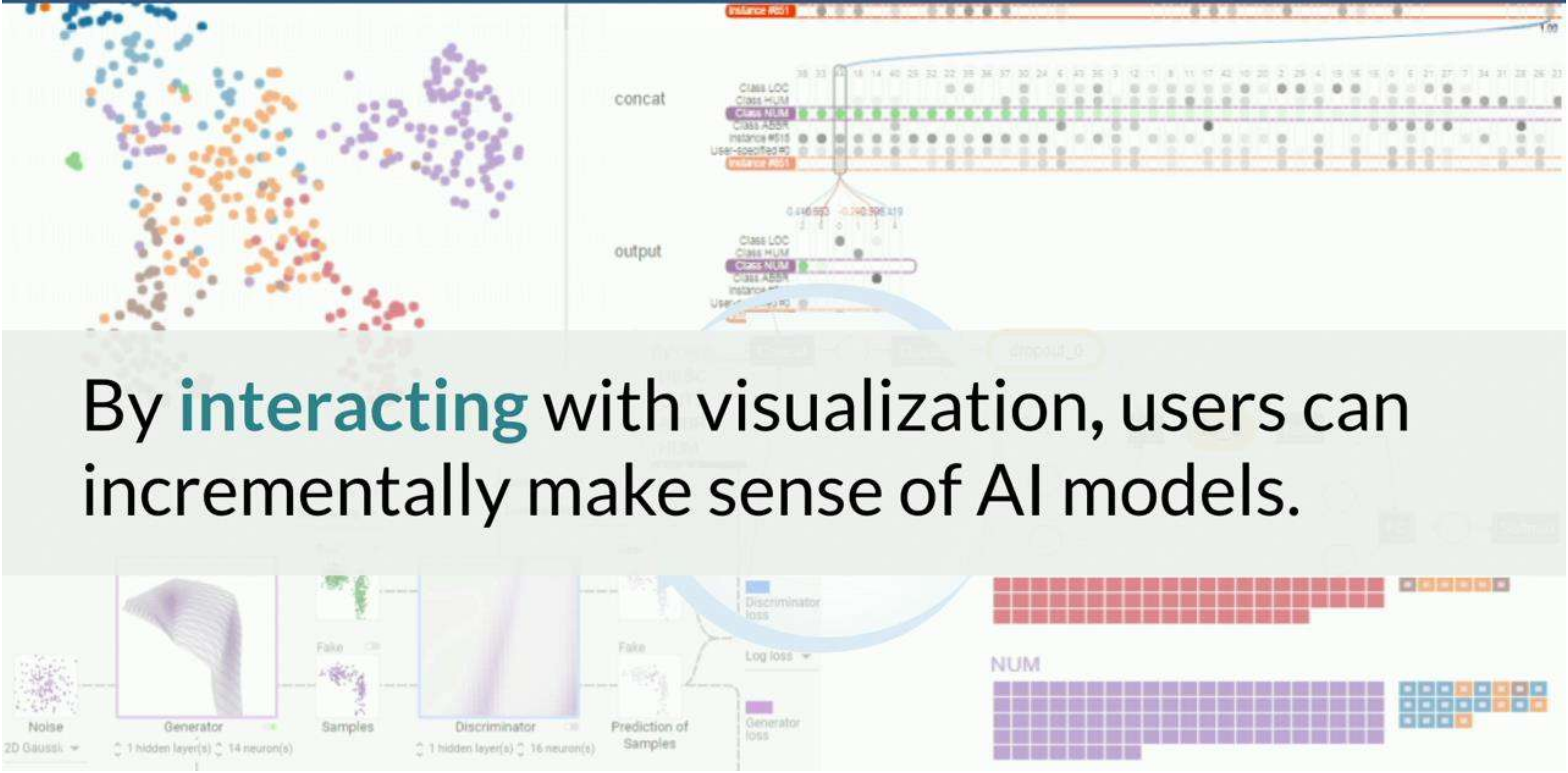
Machine learning aims to find patterns from data.

Visualization amplifies human cognition to find patterns.



Why interactive visualization?

By interacting with visualization, users can incrementally make sense of AI models.



Interpretable AI via Visual Analytics

Understanding Industry-Scale Models

ActiVis - Activation analysis by subsets

Interactive Learning of Complex Models

GAN Lab - Experimentation with GANs

Research Landscape

Survey, Gamut

ActiVis Scaling Visualization to Industry-scale Models & Data

IEEE VIS 2017

Deployed by facebook

Minsuk
Kahng
Georgia Tech

Pierre
Andrews
Facebook

Aditya
Kalro
Facebook

Polo
Chau
Georgia Tech


Facebook data scientists need visualization tools to interpret complex models

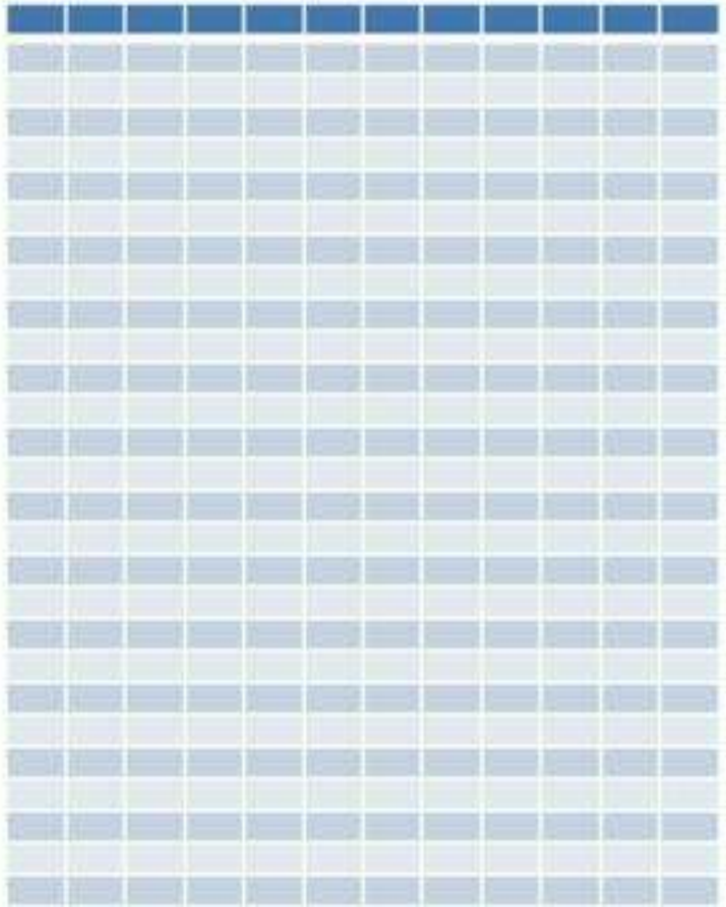
Practical Design Challenges

DEEP/WIDE MODELS

1,000+
operations/layers

LARGE DATASETS

1 billion+
instances

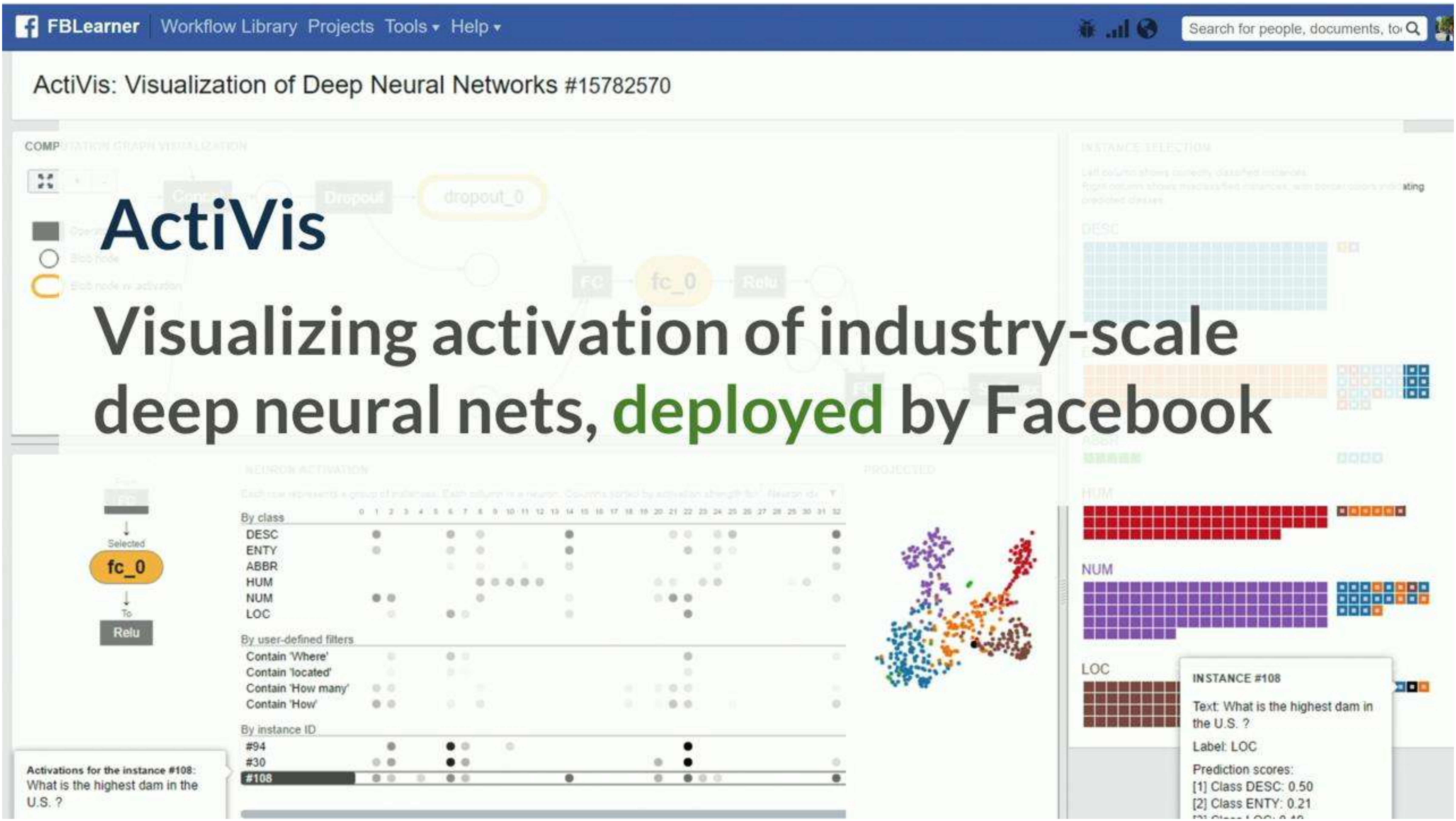


DIVERSE FEATURES

image, text, numerical, categorical, …

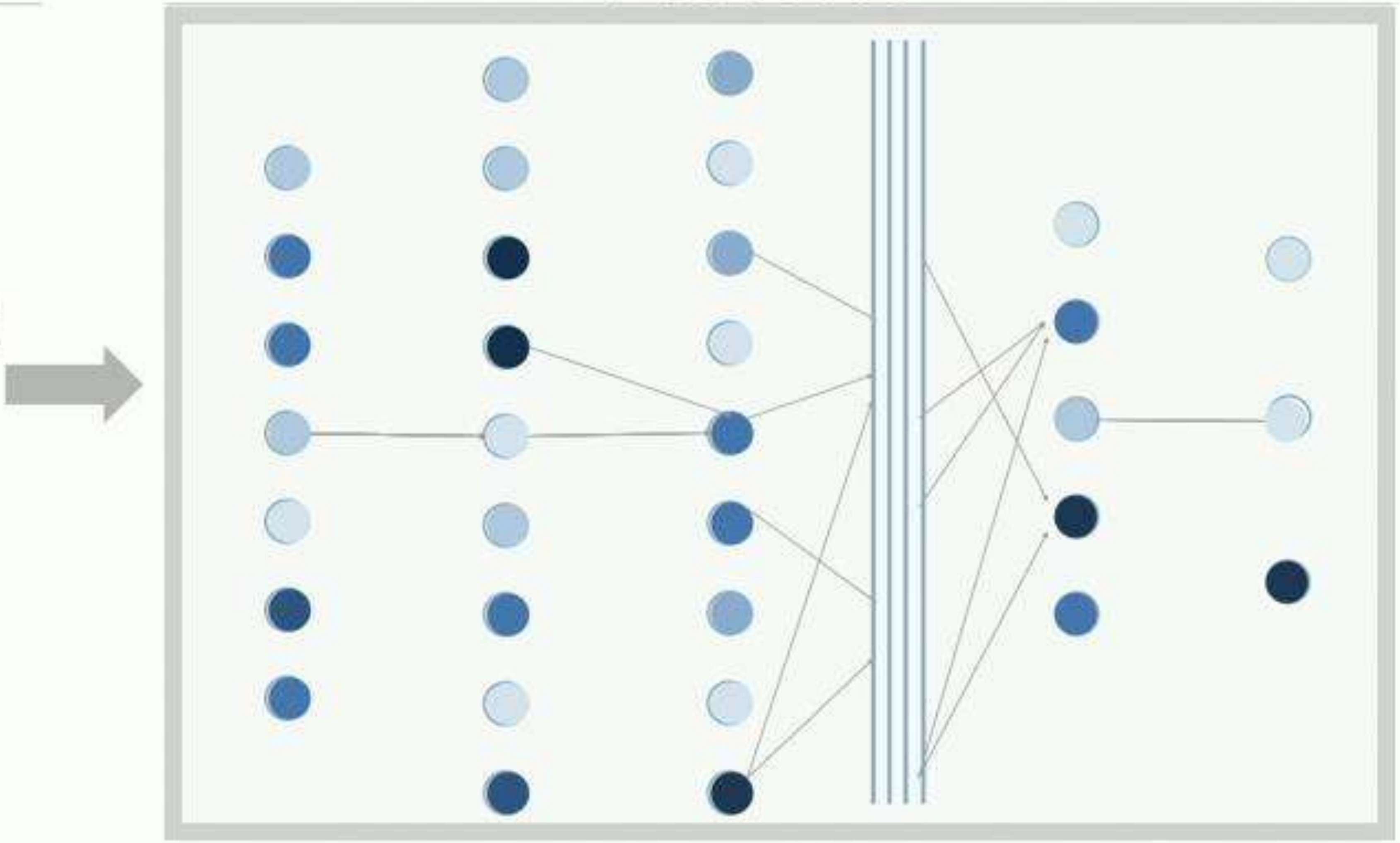
UNDERSTANDING USERS' NEEDS

Participatory design sessions with 15+ researchers, engineers & data scientists at Facebook over 11 months


ActiVis
Visualizing activation of industry-scale deep neural nets, deployed by Facebook
FBLearner
Workflow Library
Projects
Tools
Help
Search for people, documents, to
ActiVis: Visualization of Deep Neural Networks #15782570
dropout_0
fc_0
Selected
fc_0
To
Relu
By class
DESC
ENTY
ABBR
HUM
NUM
LOC
By user-defined filters
Contain 'Where'
Contain 'located'
Contain 'How many'
Contain 'How'
By instance ID
#94
#30
#108
Activations for the instance #108:
What is the highest dam in the U.S. ?
HUM
NUM
LOC
INSTANCE #108
Text: What is the highest dam in the U.S. ?
Label: LOC
Prediction scores:
[1] Class DESC: 0.50
[2] Class ENTY: 0.21

Challenge #1

How to visualize many model parameters?

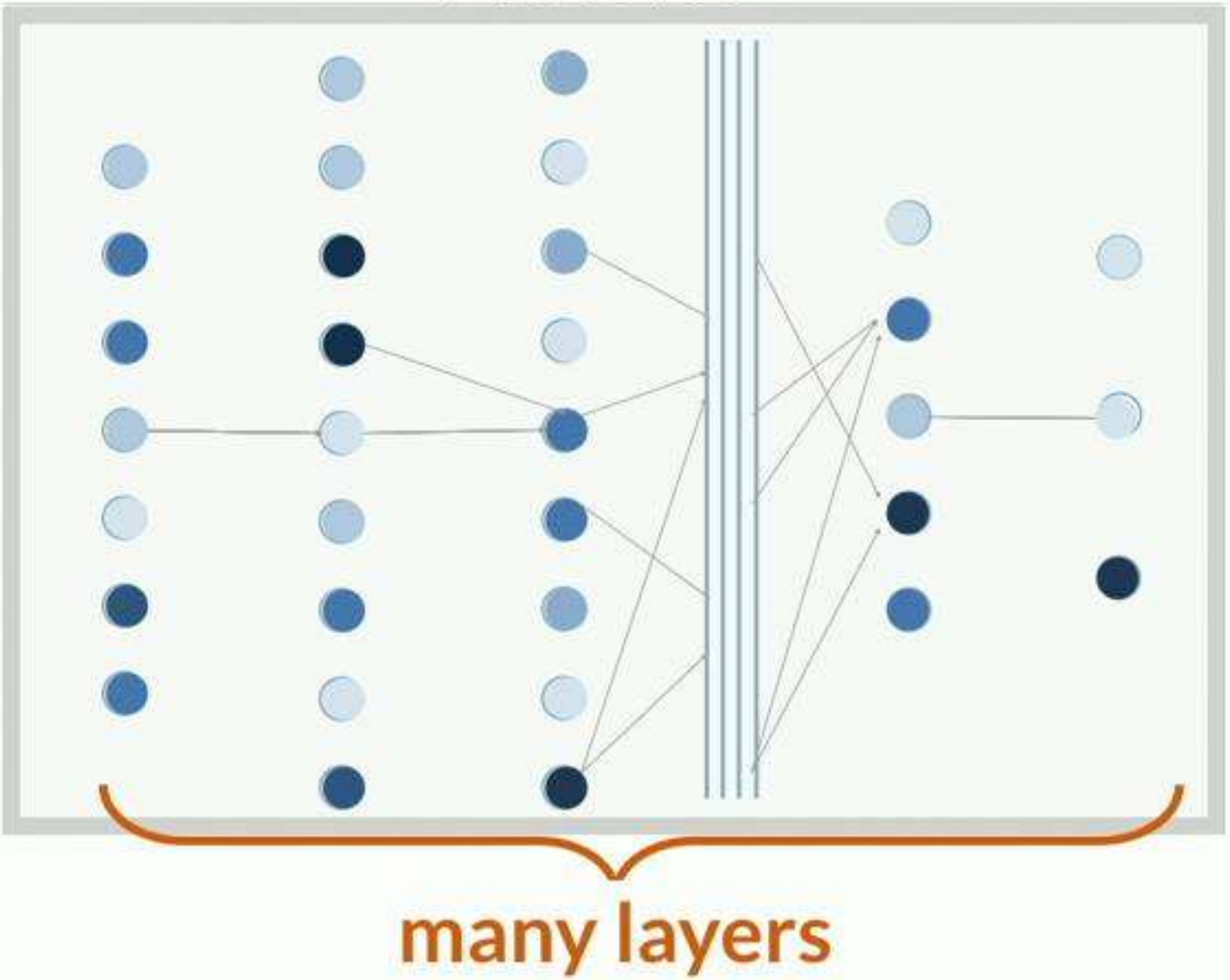
INPUT	MODEL	OUTPUT
Where is Mercedes-Benz Stadium located?		Number 11% Person 8% **Location 81%**

Challenge #1

How to visualize many model parameters?

INPUT	MODEL	OUTPUT

Where is Mercedes-Benz Stadium located?



Number 11%

Person 8%

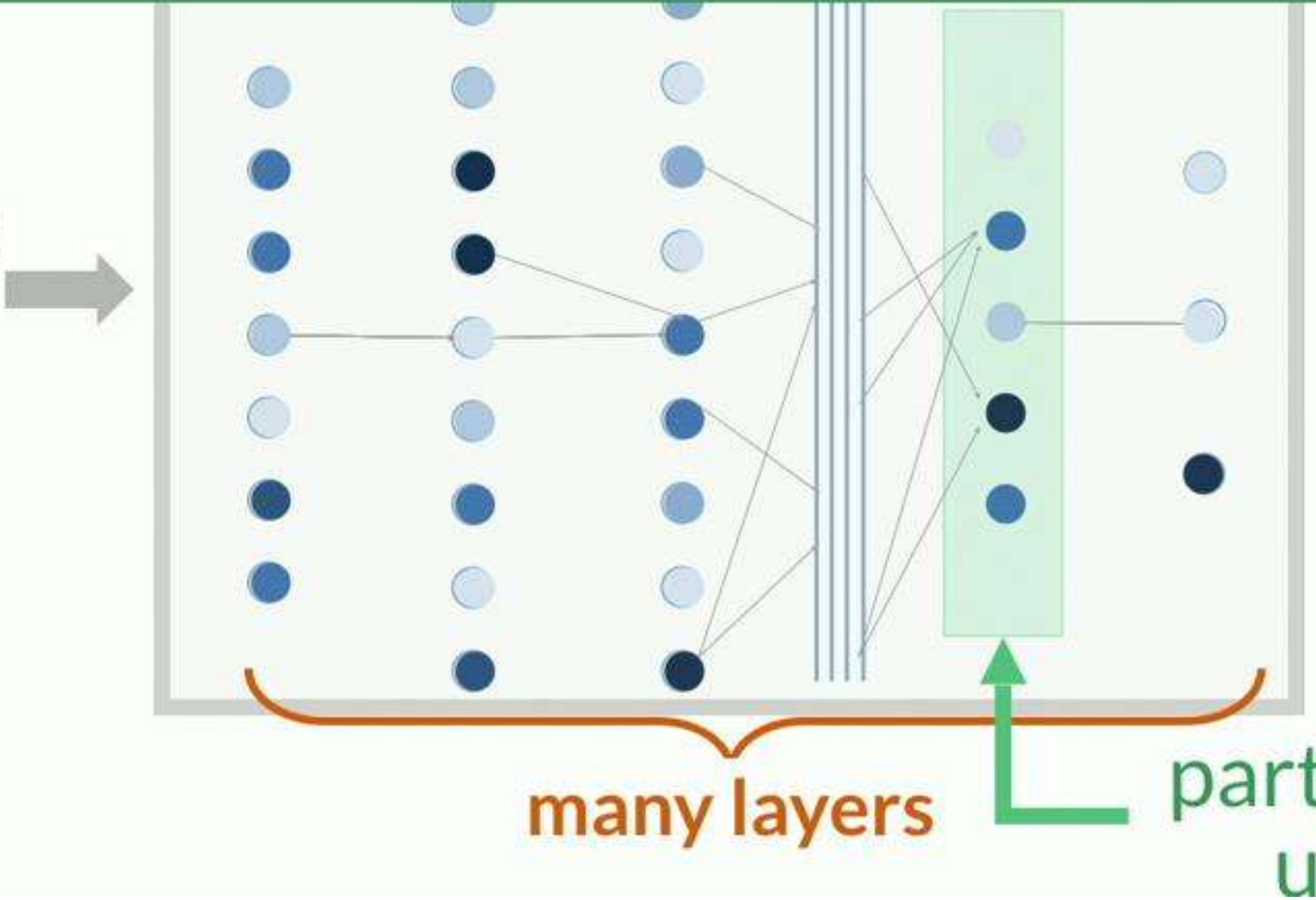
Location 81%

Challenge #1

How to visualize many model parameters?

Observation: No need to show everything

Where is Mercedes-Benz Stadium located?



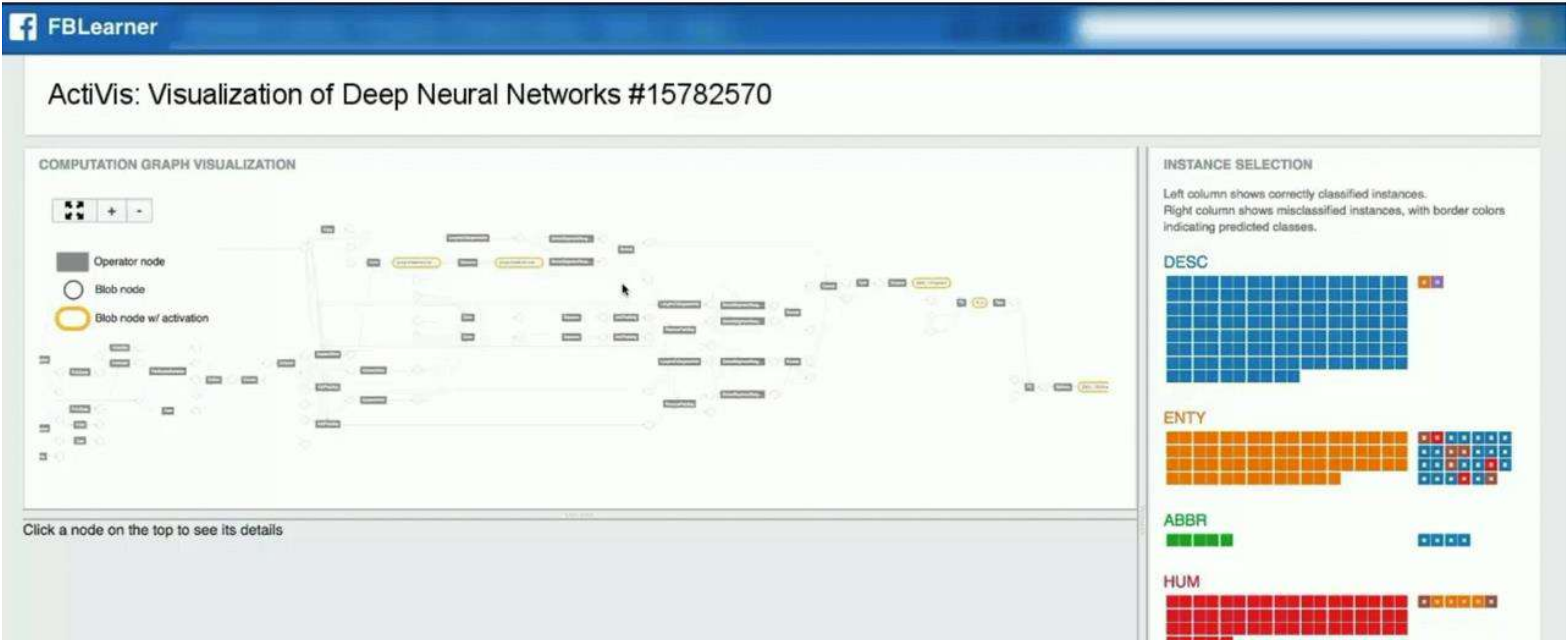
Number 11%

Person 8%

Location 81%

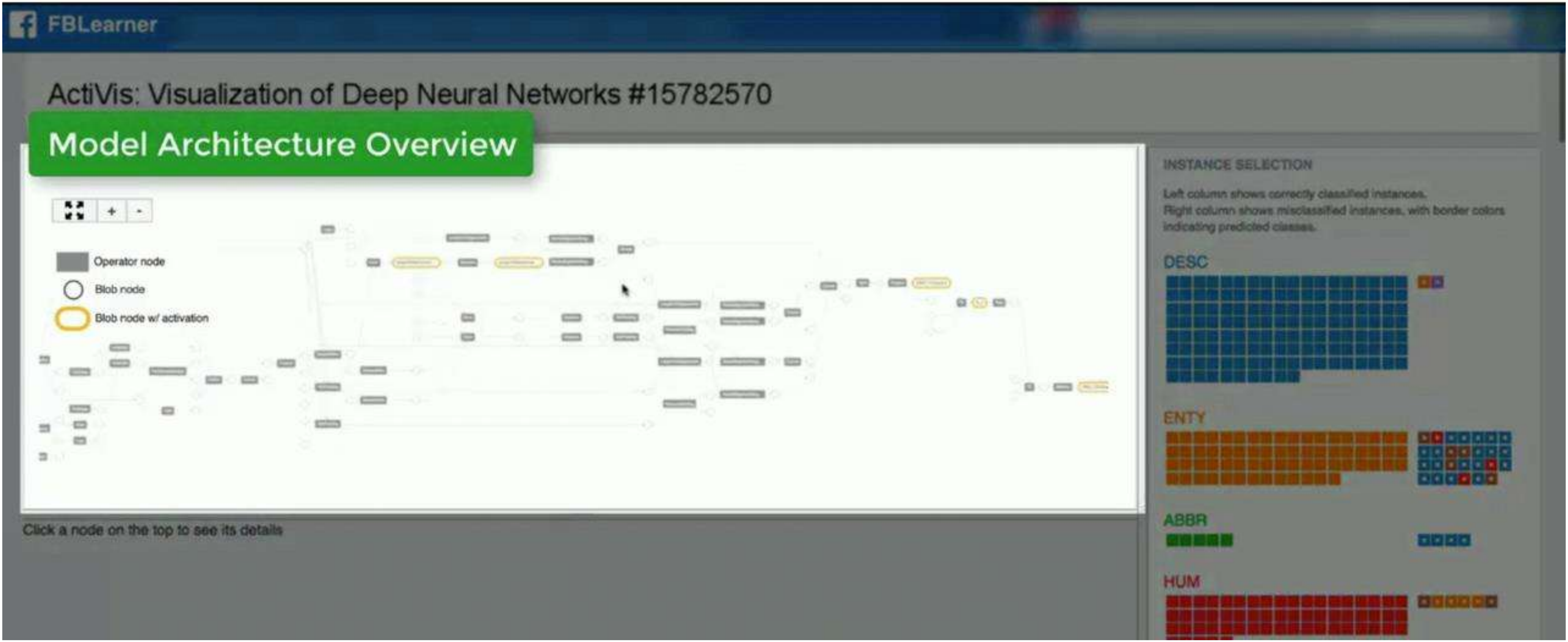
ActiVis Key Ideas #1

Model Overview to Activation Details



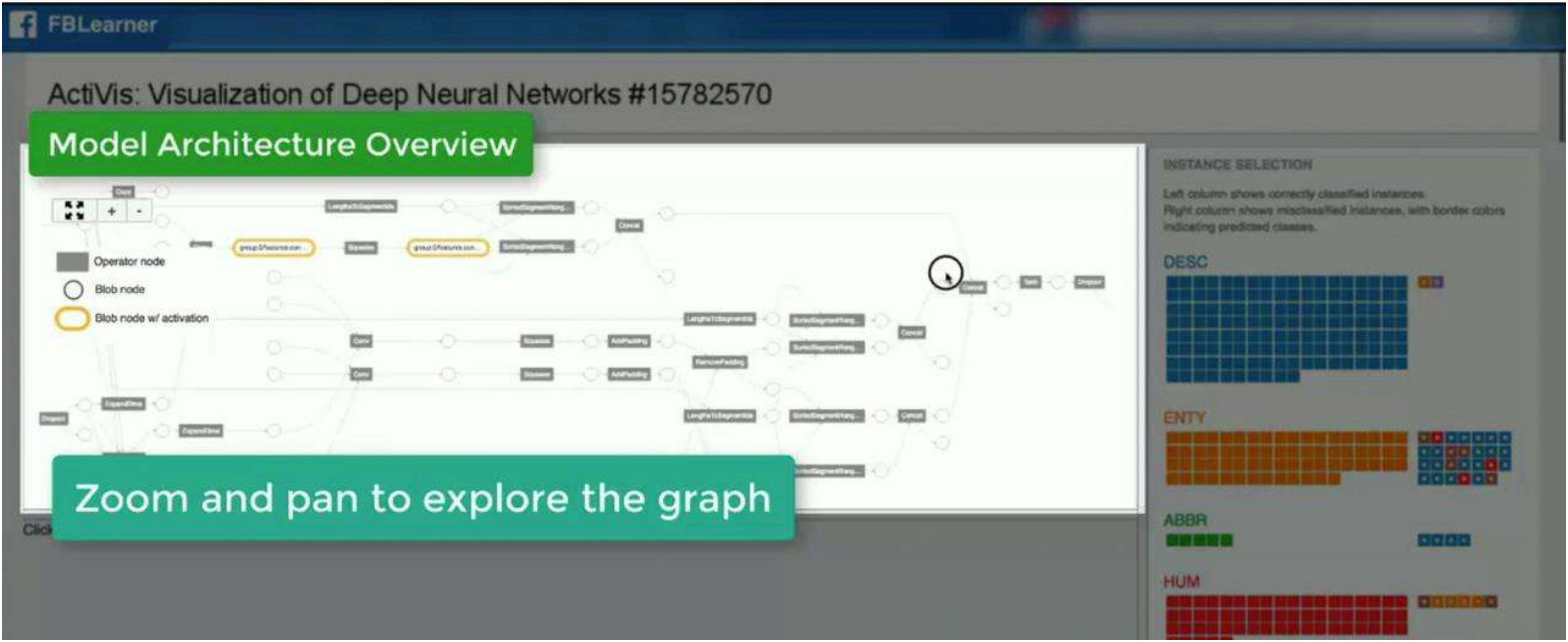
ActiVis Key Ideas #1

Model Overview to Activation Details



ActiVis Key Ideas #1

Model Overview to Activation Details



Challenge #2

How to analyze many data instances?

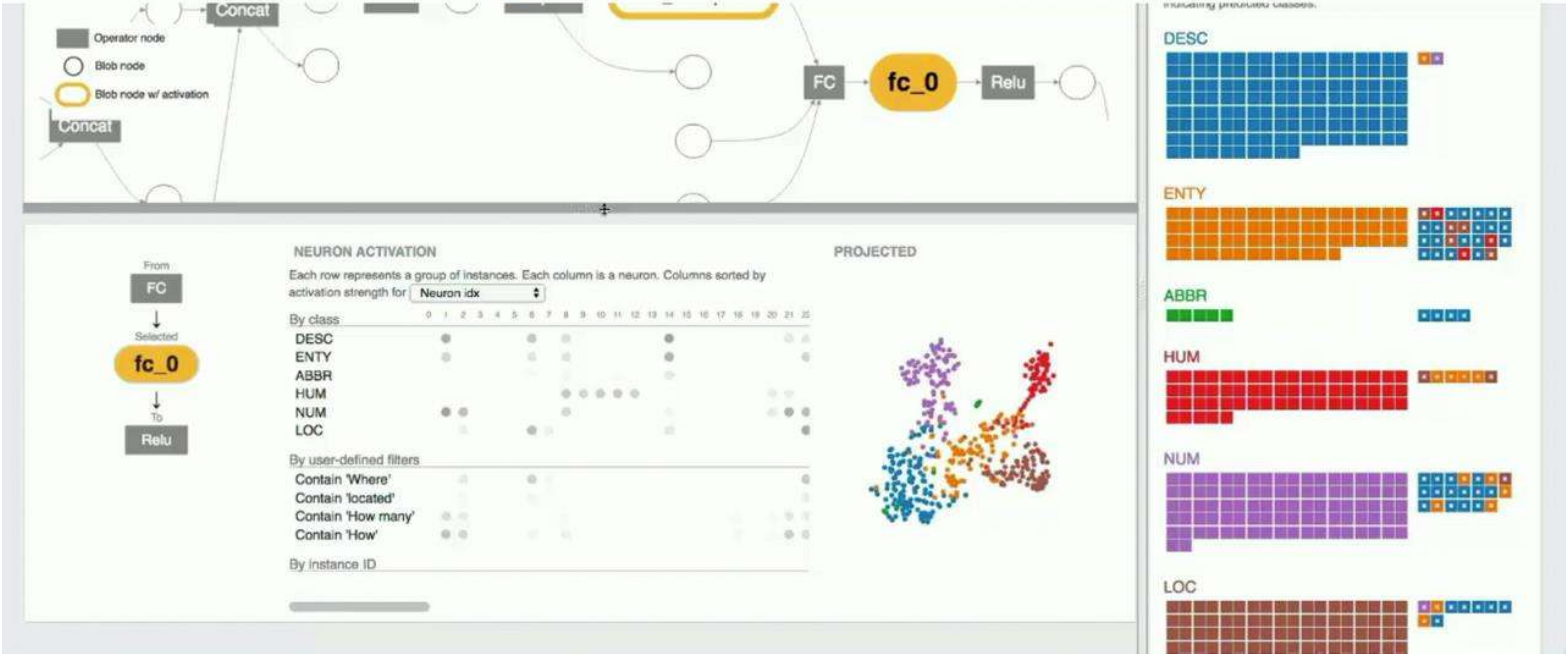
Observation: Two Analytics Patterns

Complementary

INSTANCE-LEVEL	**SUBSET**-LEVEL
How model responds to individual instances?	How model behaves at higher-level categorization (e.g., by topic)?
Useful for debugging	Useful for large datasets

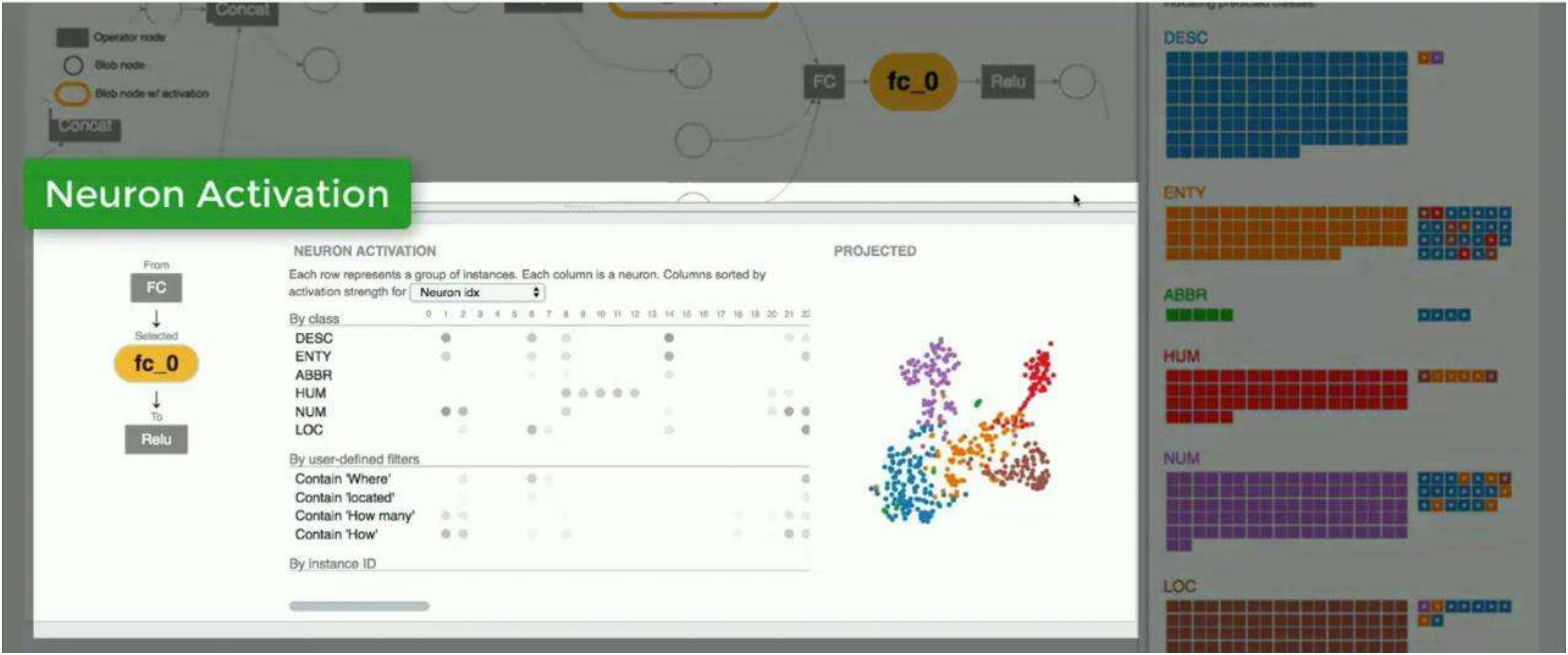
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets



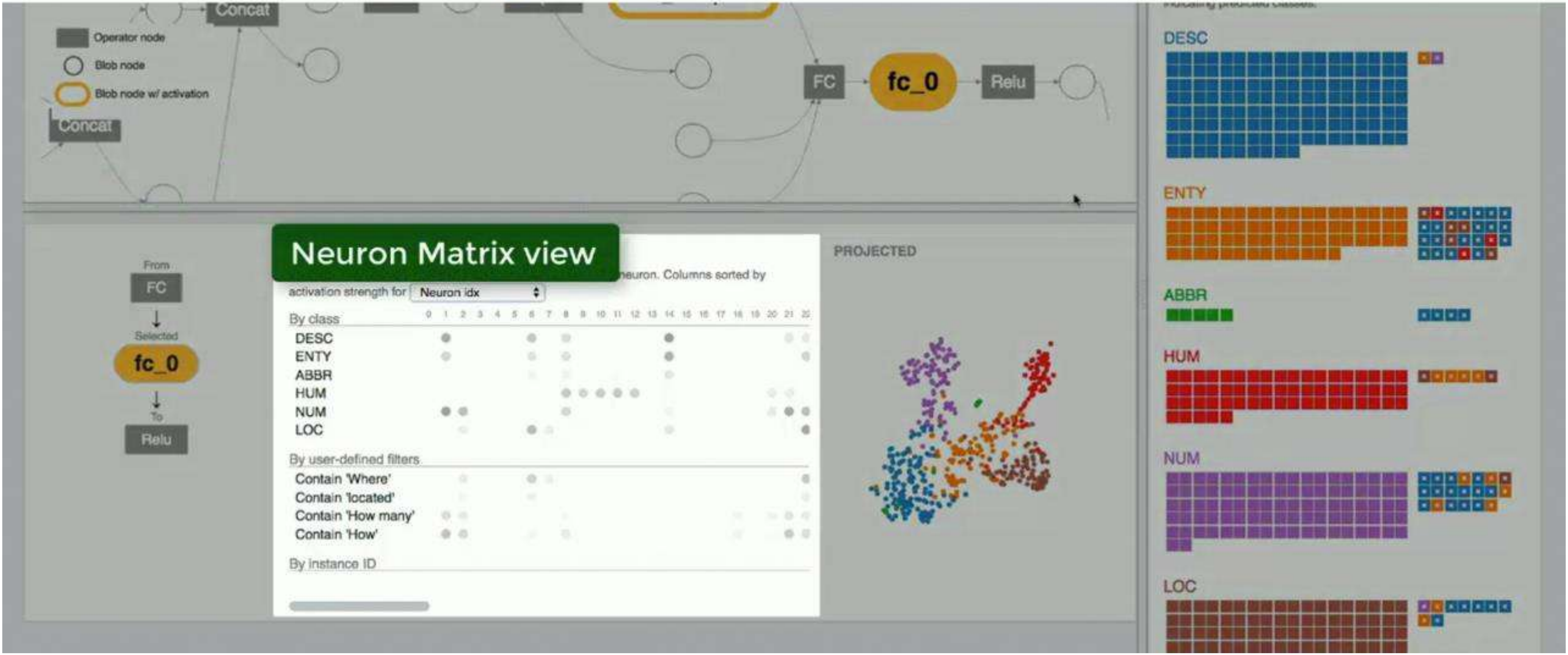
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets



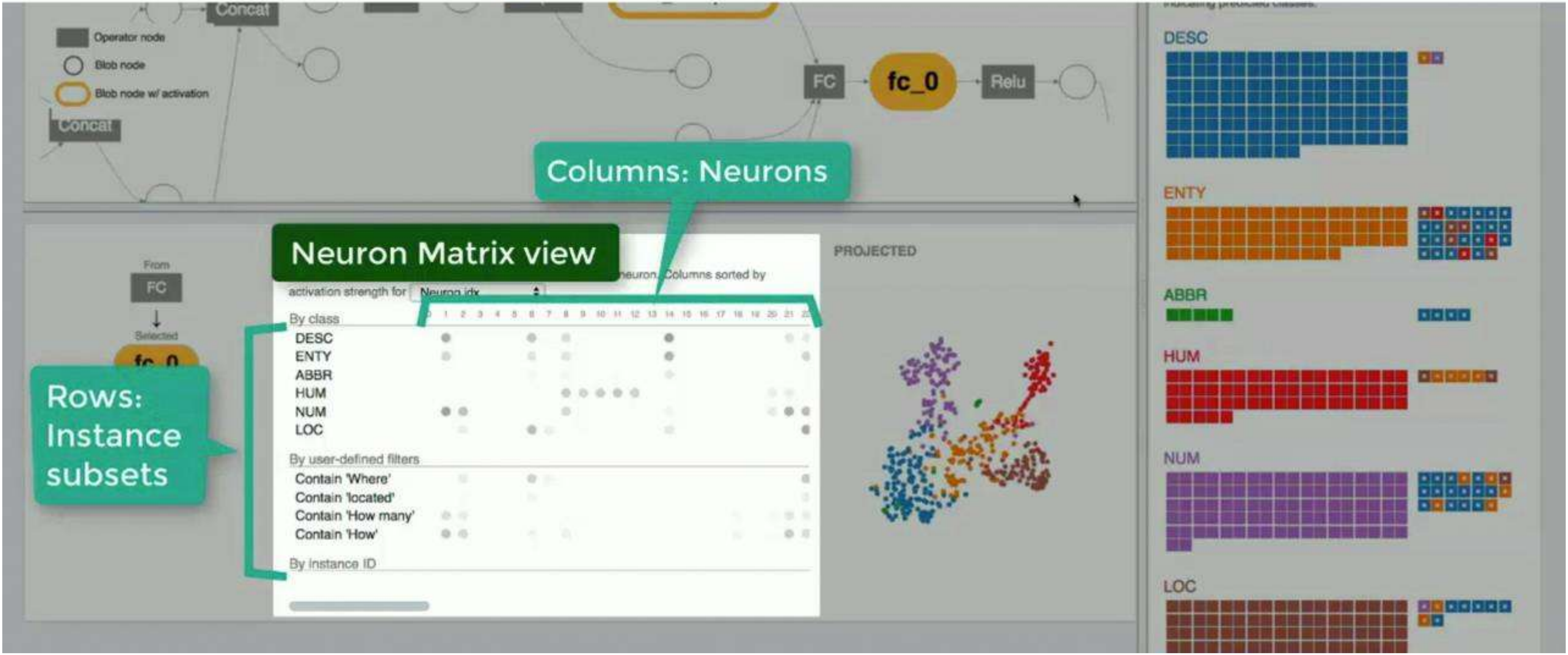
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets



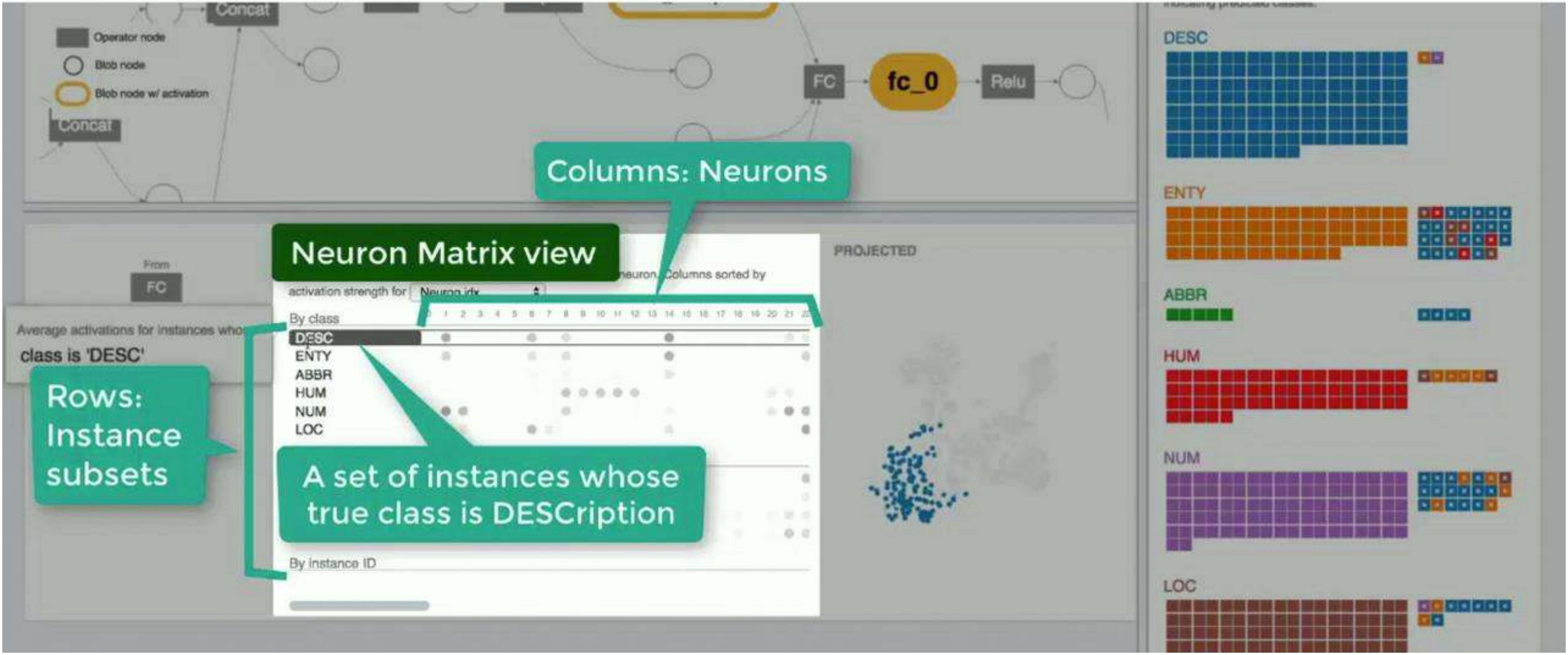
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets



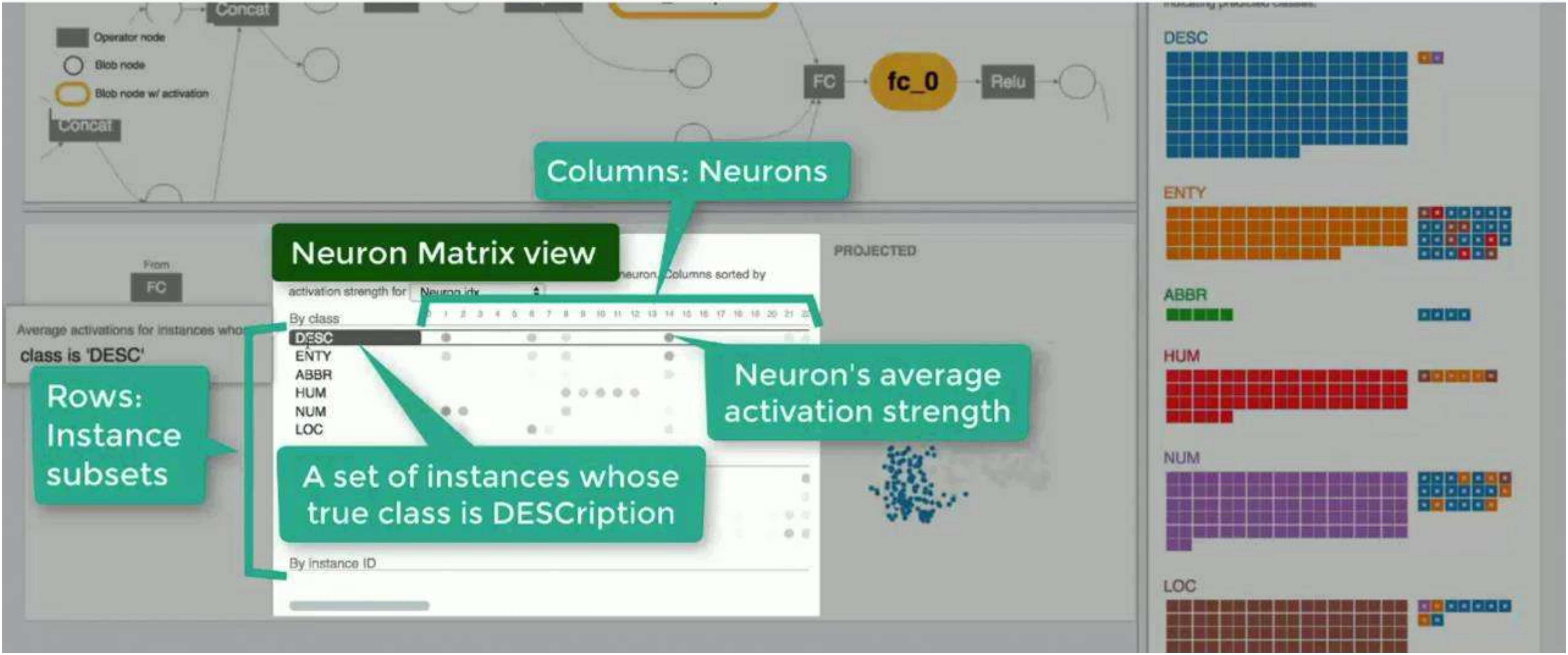
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets



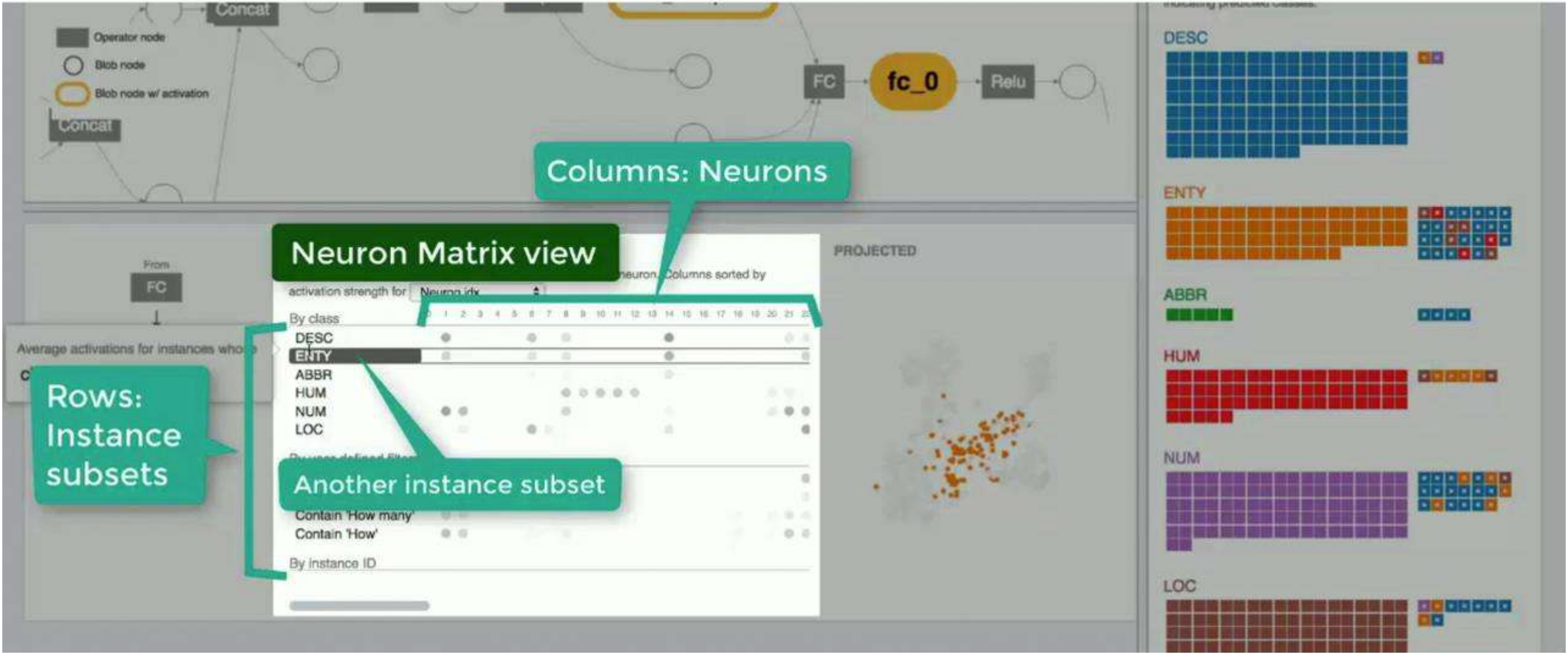
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets



ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets



ActiVis Key Ideas (3)

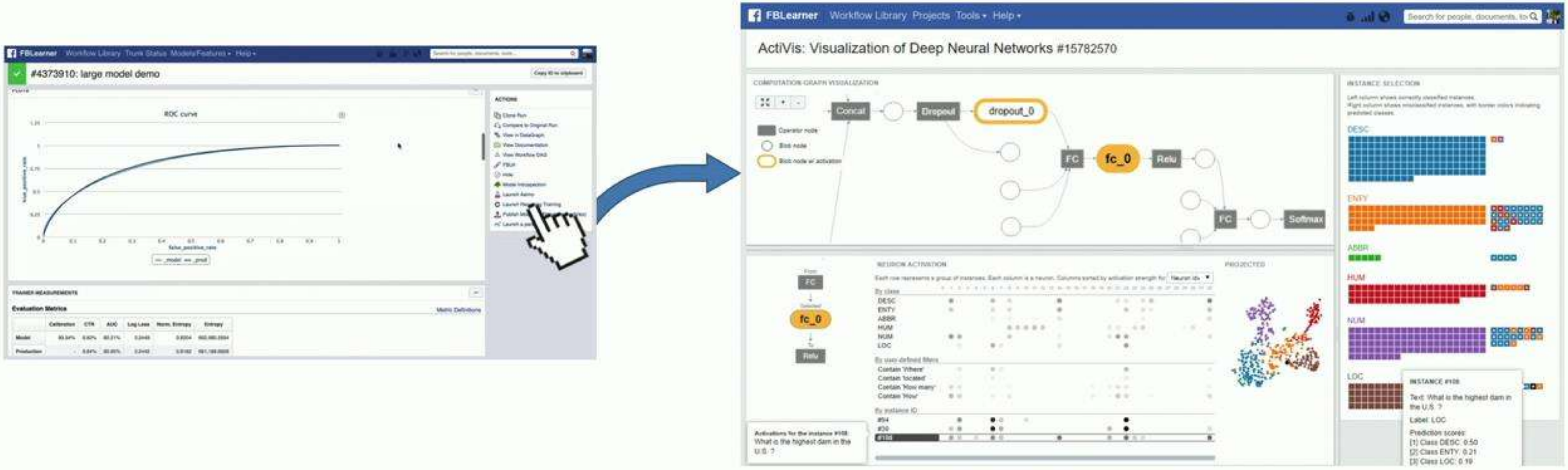
Scaling Up ActiVis for Facebook

1. User-guided Instance Sampling
2. Selective Pre-computation of Layers
3. Matrix Computation for Billion-Scale Instances

Deployed on *FBLearner*

Facebook's ML platform

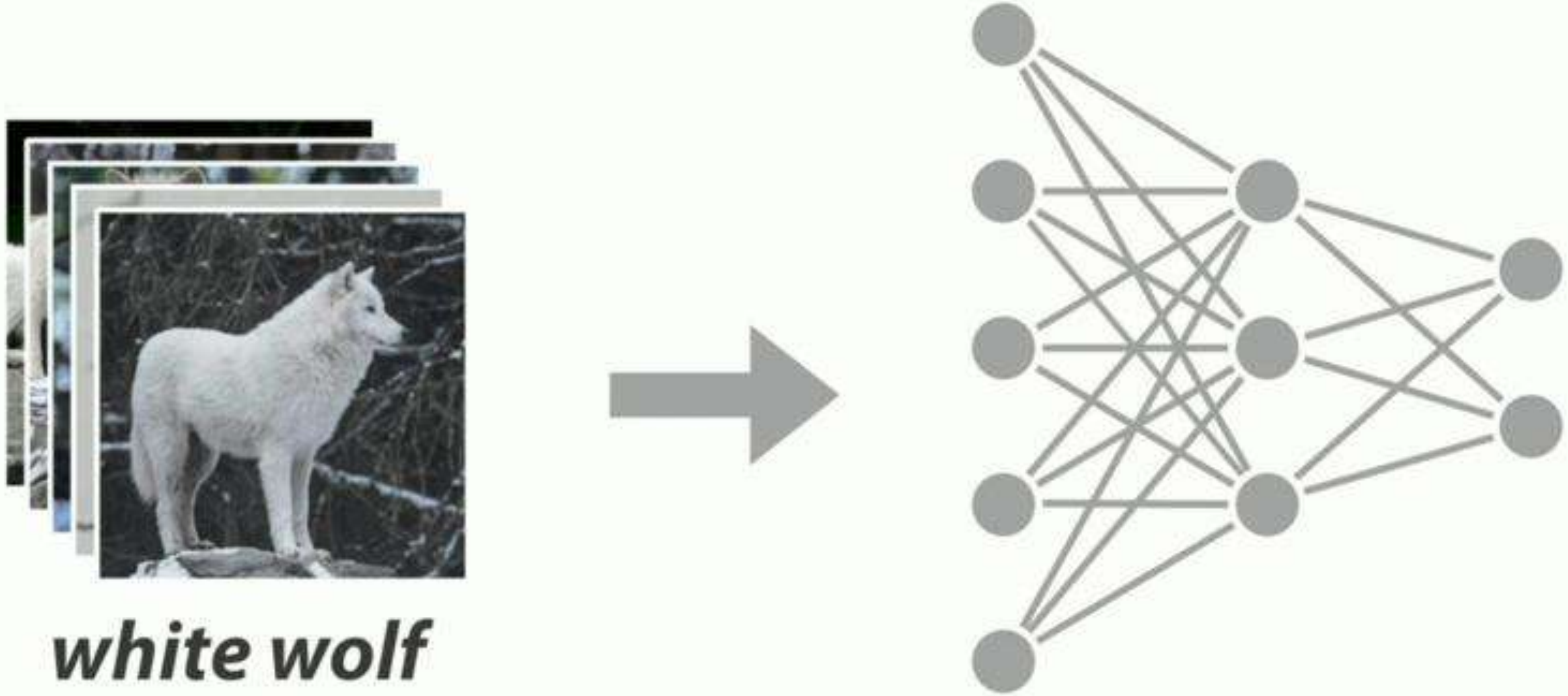
used by >25% of engineering team



SUMMIT

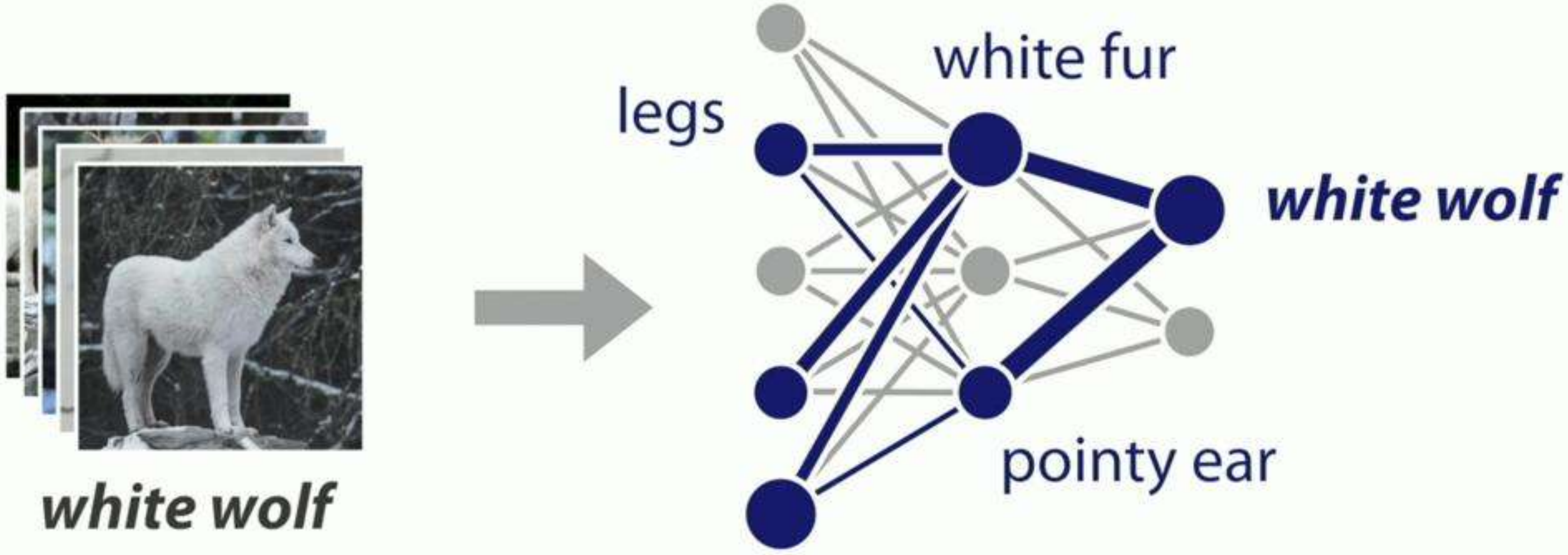
IEEE VIS 2019

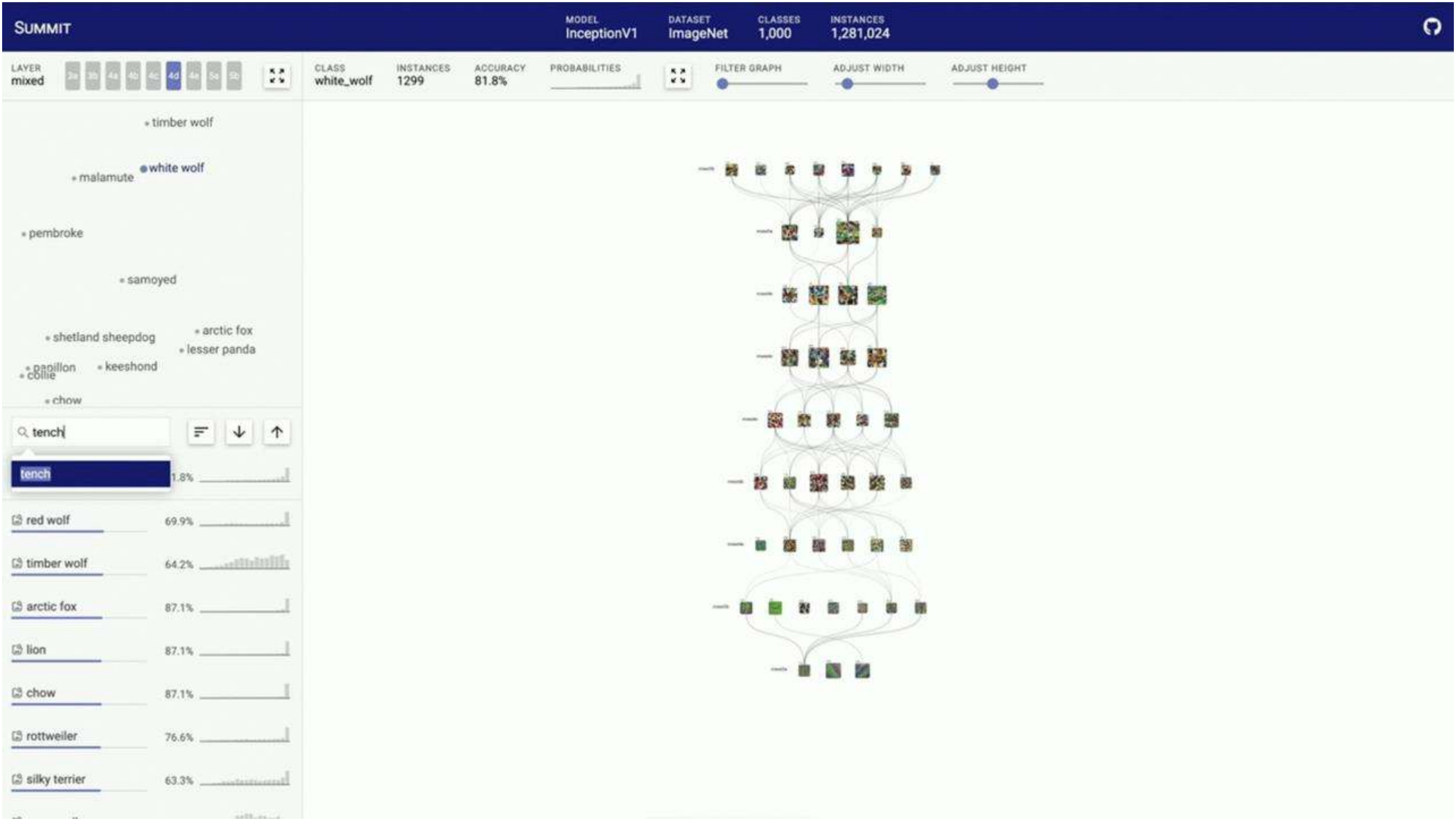
Scalably summarize and **interactively visualize** neural network feature representations for millions of images

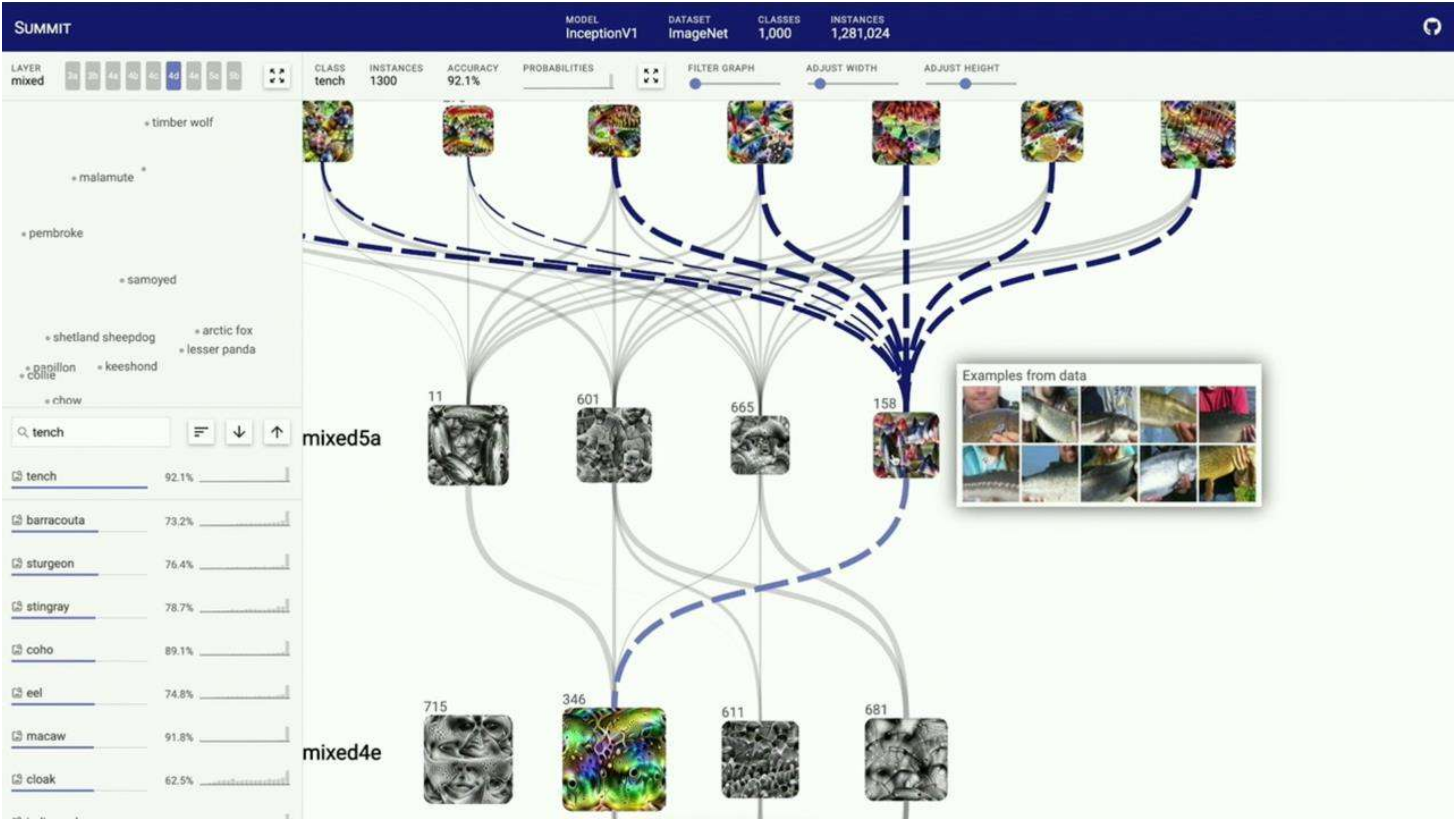


SUMMIT

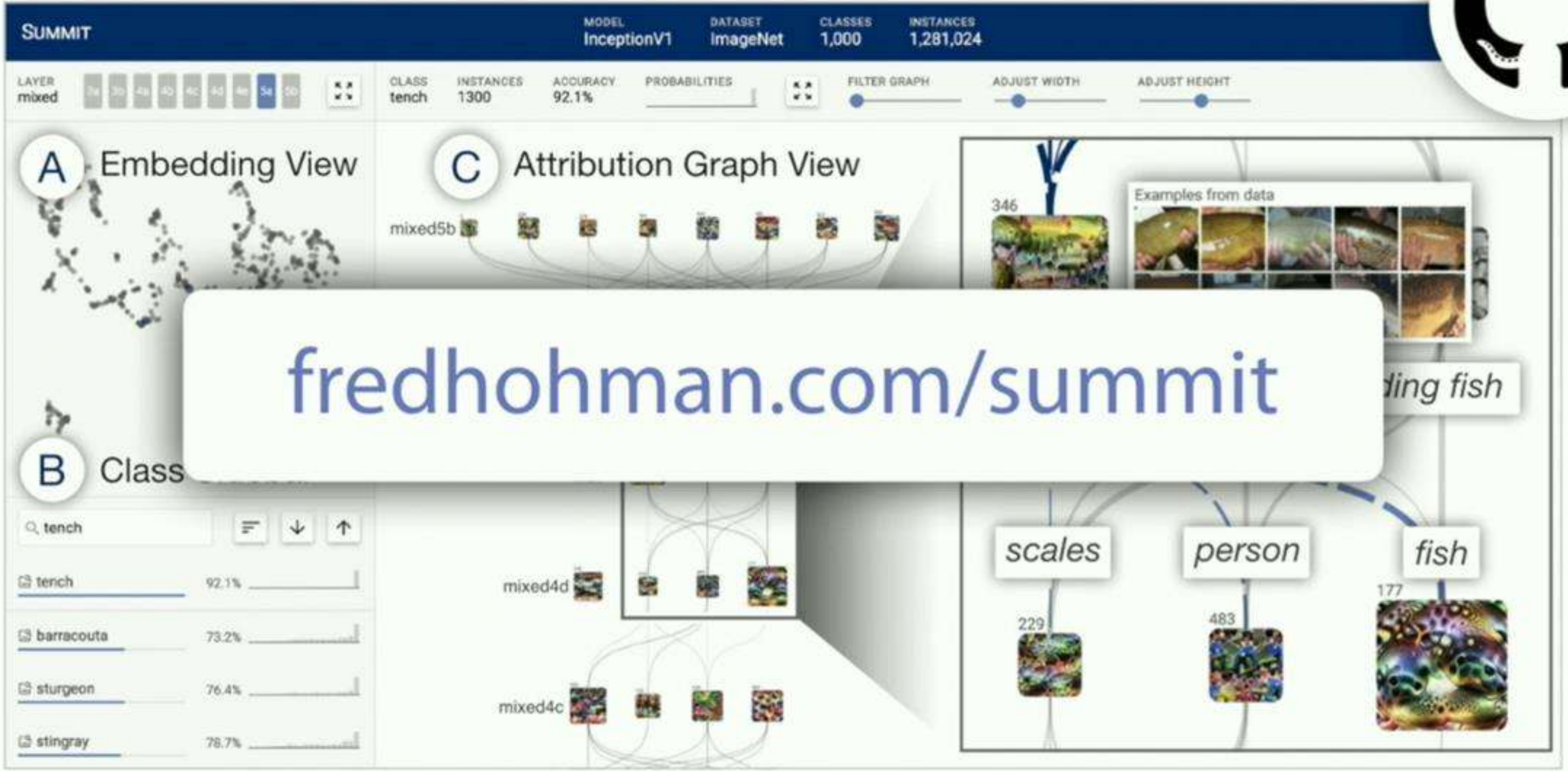
Scalably summarize and **interactively visualize** neural network feature representations for millions of images




SUMMIT
MODEL
InceptionV1
DATASET
ImageNet
CLASSES
1,000
INSTANCES
1,281,024
LAYER
mixed
4d
CLASS
white_wolf
INSTANCES
1299
ACCURACY
81.8%
PROBABILITIES
FILTER GRAPH
ADJUST WIDTH
ADJUST HEIGHT
timber wolf
white wolf
malamute
pembroke
samoyed
arctic fox
shetland sheepdog
lesser panda
papillon
collie
keeshond
chow
tench
tench
red wolf
69.9%
timber wolf
64.2%
arctic fox
87.1%
lion
87.1%
chow
87.1%
rottweiler
76.6%
silky terrier
63.3%


SUMMIT
MODEL InceptionV1
DATASET ImageNet
CLASSES 1,000
INSTANCES 1,281,024
LAYER mixed
CLASS tench
INSTANCES 1300
ACCURACY 92.1%
PROBABILITIES
FILTER GRAPH
ADJUST WIDTH
ADJUST HEIGHT
timber wolf
malamute
pembroke
samoyed
shetland sheepdog
arctic fox
lesser panda
papillon
collie
keeshond
chow
tench
tench 92.1%
barracouta 73.2%
sturgeon 76.4%
stingray 78.7%
coho 89.1%
eel 74.8%
macaw 91.8%
cloak 62.5%
mixed5a
11
601
665
158
Examples from data
mixed4e
715
346
611
681

See more in the paper, and try our open-source demo!



fredhohman.com/summit

SUMMIT

Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations

Fred Hohman, Haekyu Park, Caleb Robinson, Duen Horng (Polo) Chau

IEEE VIS 2019
Vancouver, Canada

Georgia Tech

Interpretable AI via Visual Analytics

Understanding Industry-Scale Models

ActiVis - Activation analysis by subsets

Interactive Learning of Complex Models

GAN Lab - Experimentation with GANs

Research Landscape

Survey, Gamut

GAN Lab

Understanding Complex Deep Generative Models using Interactive Visual Experimentation





Minsuk Kahng
Georgia Tech

Nikhil Thorat
Google

Polo Chau
Georgia Tech

Fernanda Viégas
Google

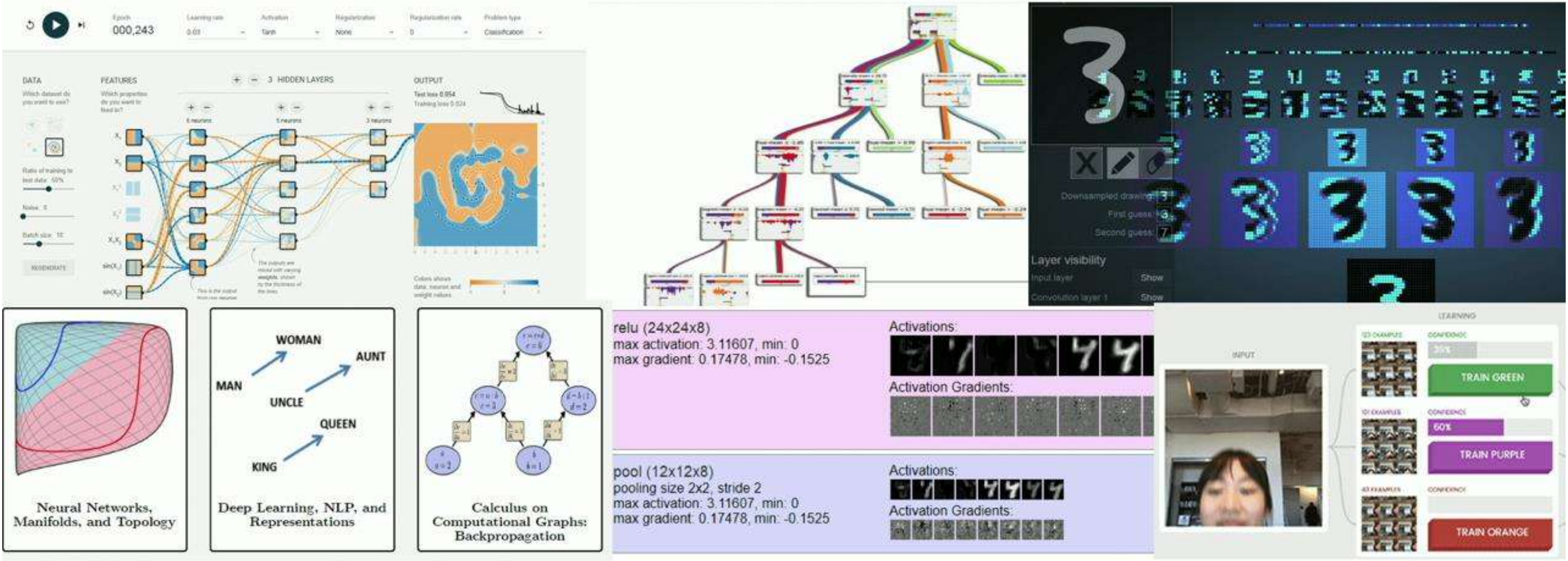
Martin Wattenberg
Google

Georgia Tech

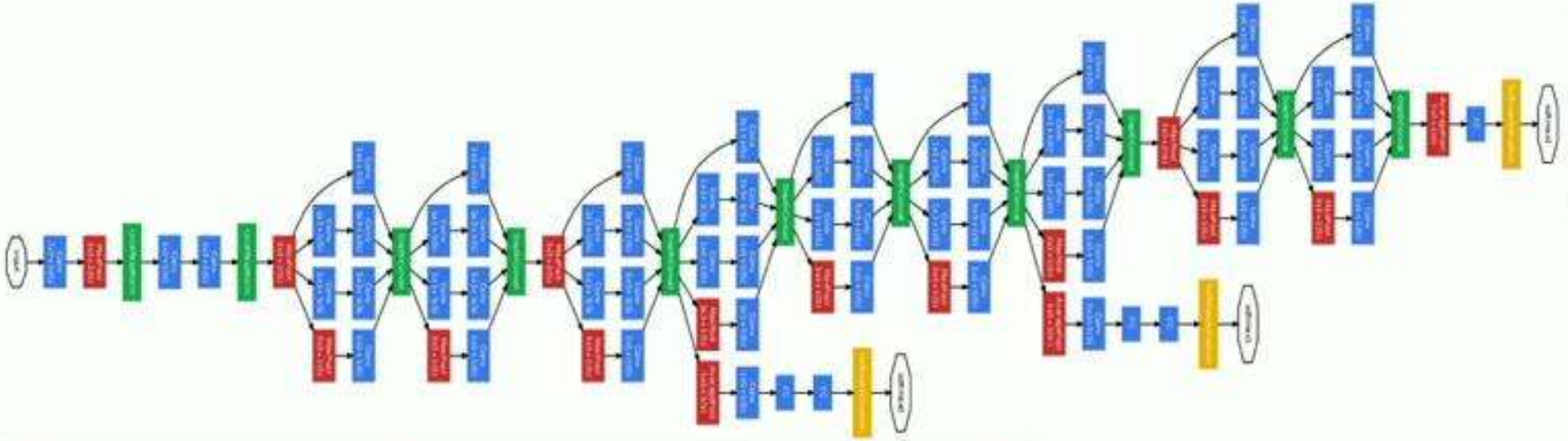
Google AI

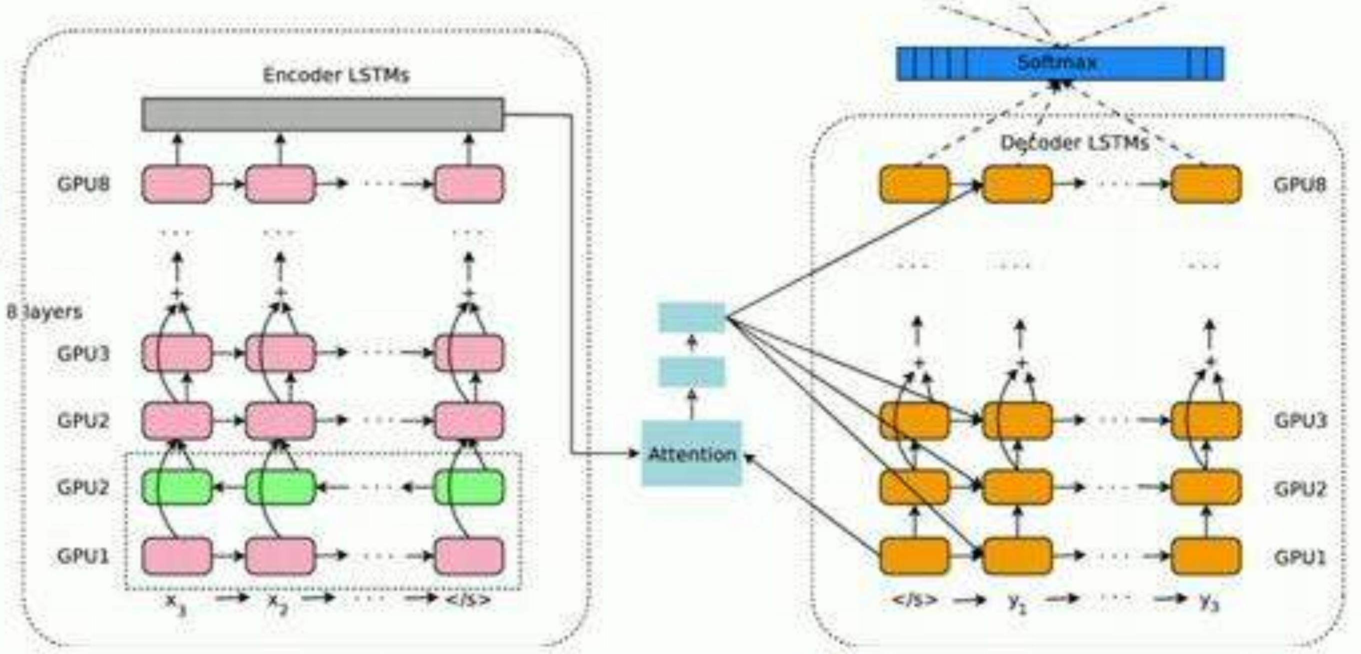
PAIR | People + AI Research Initiative

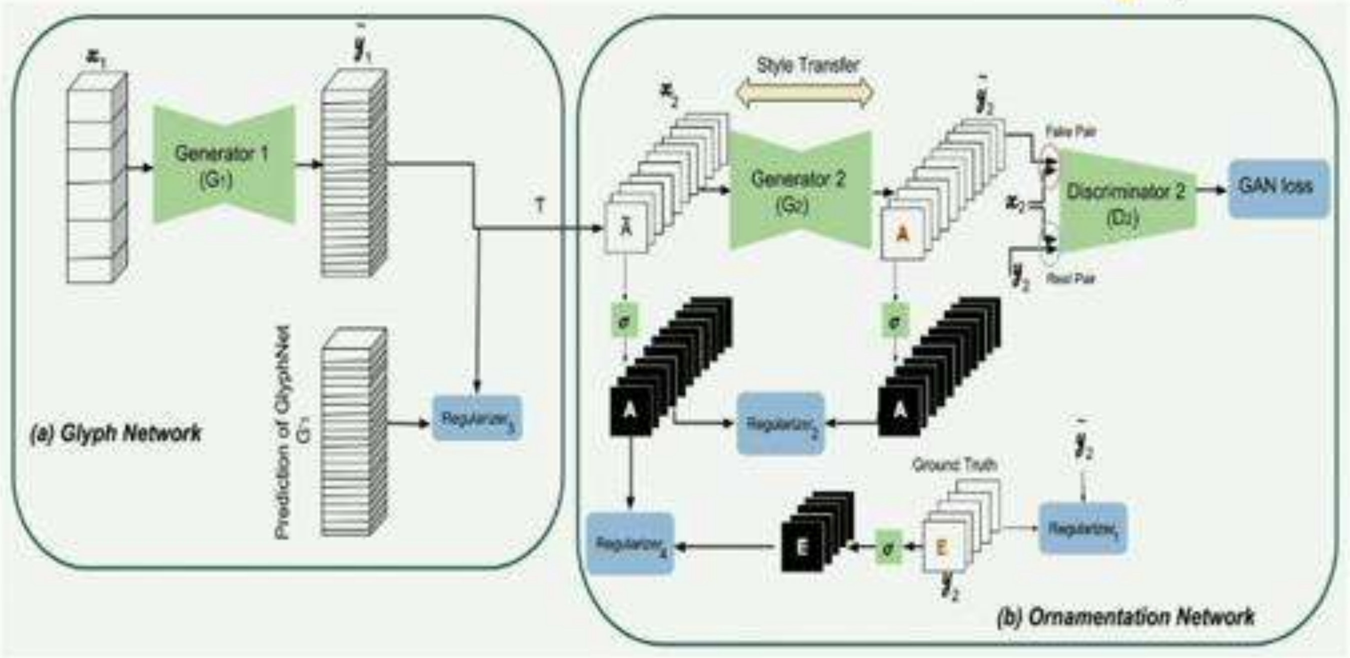
Visualization for ML Education

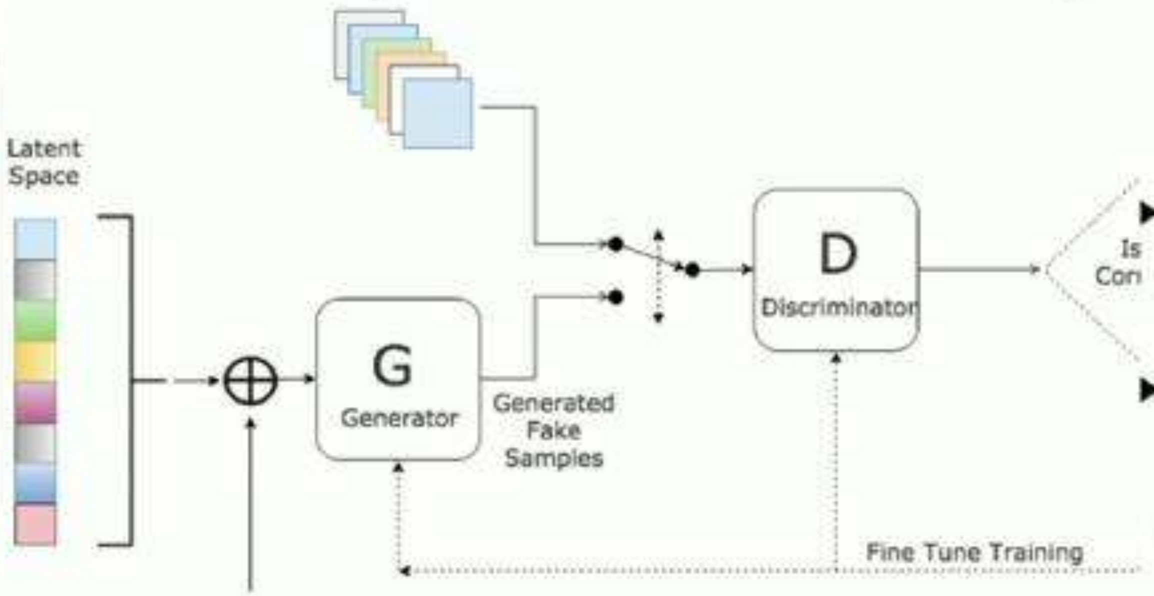


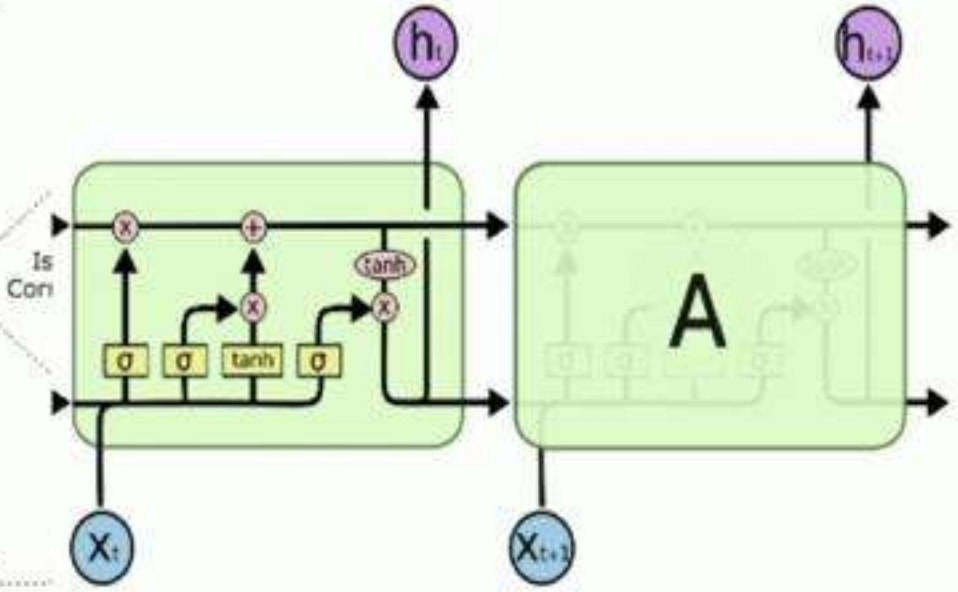
Modern deep models are complex











Generative Adversarial Networks (GANs)

"the most interesting idea in the last 10 years in ML"

- Yann LeCun

Face images generated by BEGAN [Berthelot et al., 2017]

Generative Adversarial Networks (GANs)

Hard to understand and train even for experts

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Discriminator

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

Generator

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

Why GANs are hard?

A GAN uses two *competing* neural networks

Why GANs are hard?

A GAN uses two *competing* neural networks

Generator
synthesizes outputs

Discriminator
spots fake

Why GANs are hard?

A GAN uses two *competing* neural networks

Generator
synthesizes outputs

Counterfeiter
makes fake bills

Discriminator
spots fake

Why GANs are hard?

A GAN uses two *competing* neural networks

Generator
synthesizes outputs

Counterfeiter
makes fake bills

Discriminator
spots fake

Police
spots fake bills

Why GANs are hard?

A GAN uses two *competing* neural networks

Generator
synthesizes outputs

Discriminator
spots fake

Counterfeiter
makes fake bills

Police
spots fake bills

Why GANs are hard?

A GAN uses two *competing* neural networks

Generator
synthesizes outputs

Discriminator
spots fake

Counterfeiter
makes fake bills

Police
spots fake bills

Why GANs are hard?

A GAN uses two *competing* neural networks

Generator
synthesizes outputs

Discriminator
spots fake

Counterfeiter
makes fake bills

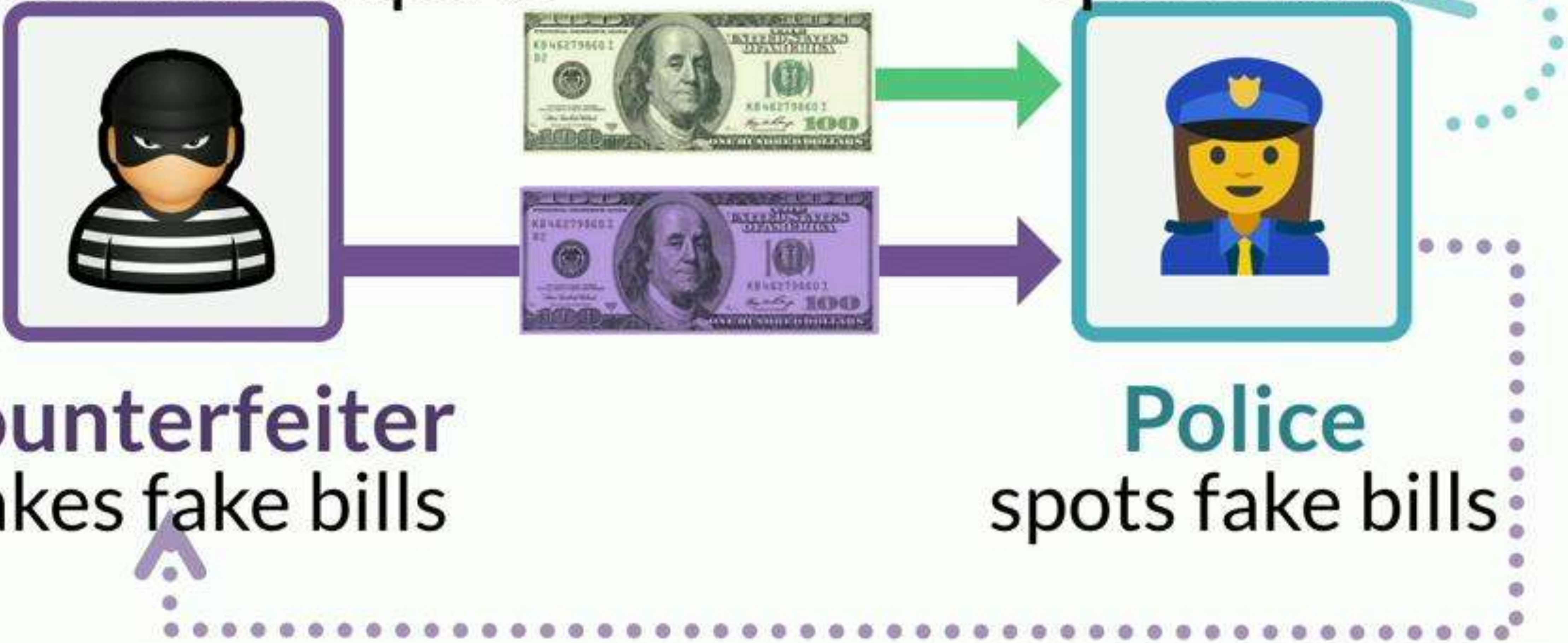
Police
spots fake bills

Why GANs are hard?

A GAN uses two *competing* neural networks

Generator
synthesizes outputs

Discriminator
spots fake



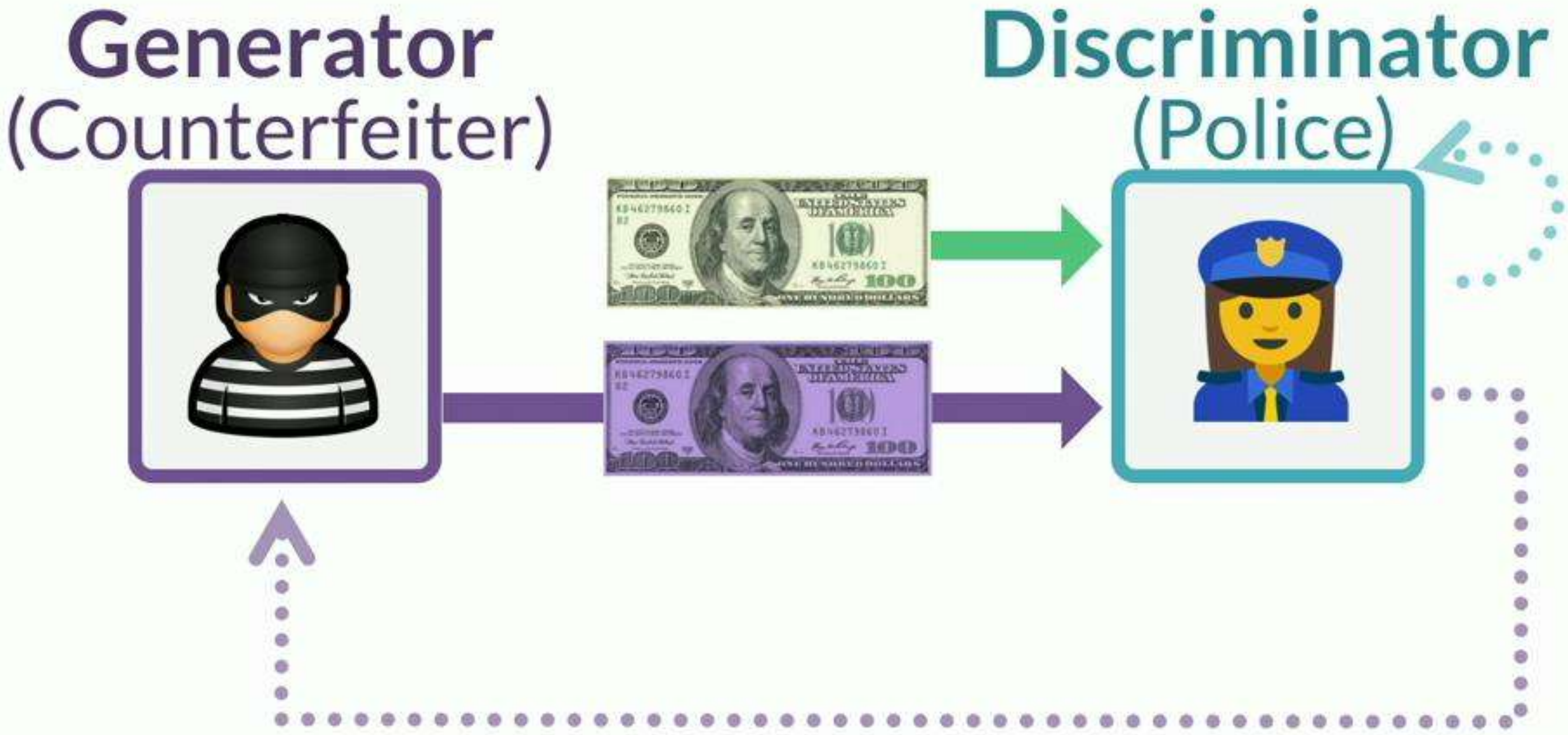
Counterfeiter
makes fake bills

Police
spots fake bills

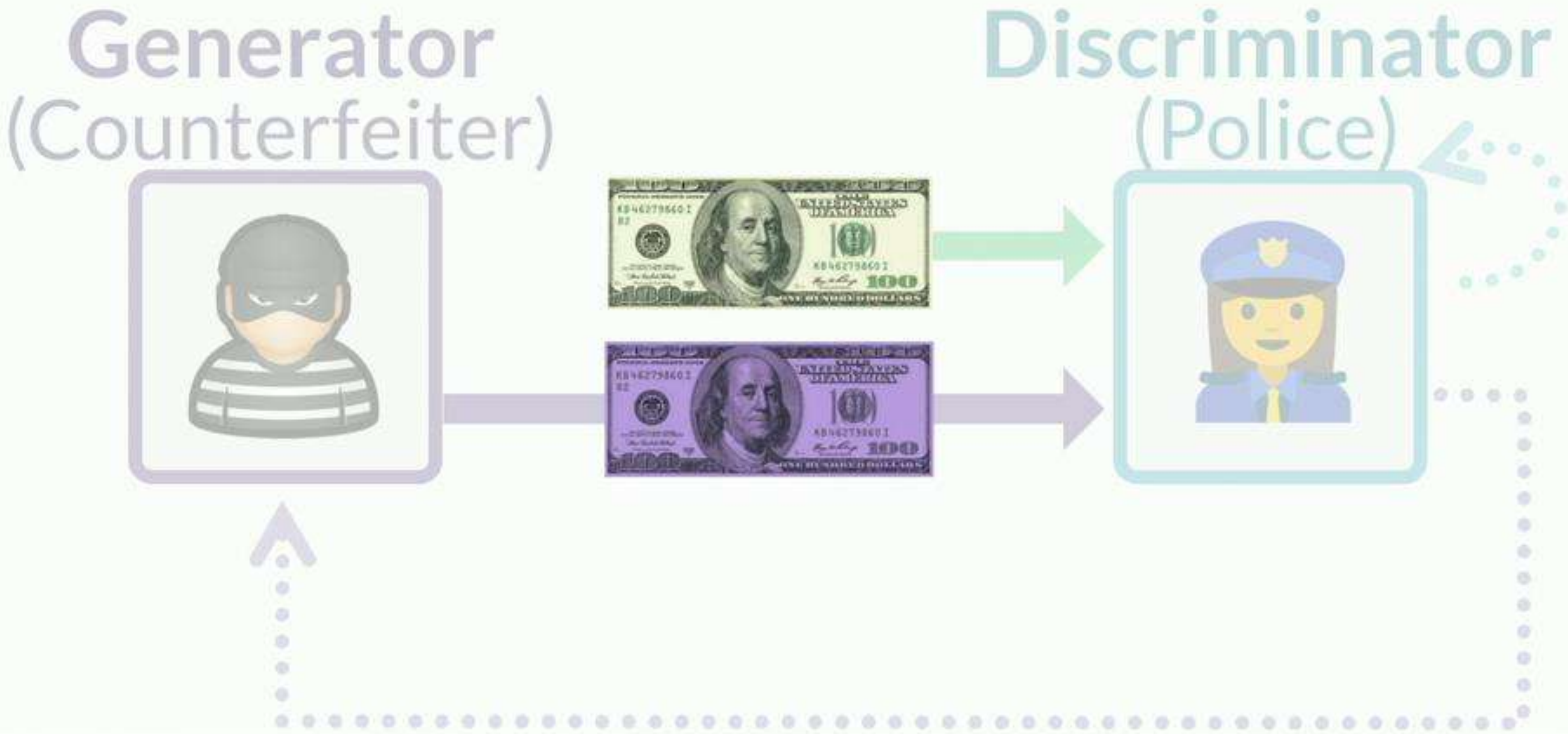
GAN Lab Research Challenges

Can we design an interactive tool for GANs?

1. Conceptual understanding of GANs
2. Interactive model training
3. Easily accessible by anyone

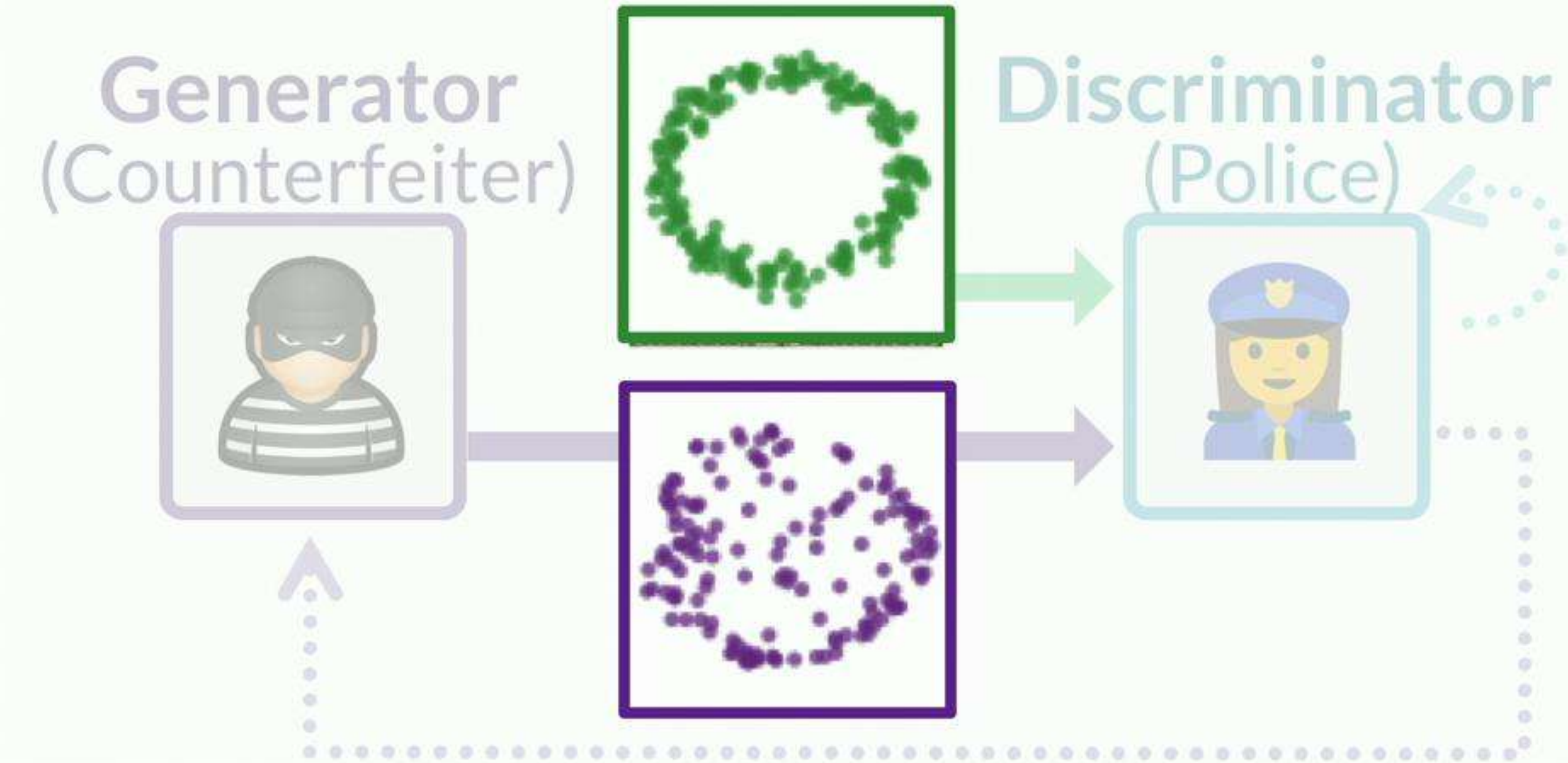

Generator
(Counterfeiter)
Discriminator
(Police)

What type of *data* to visualize?



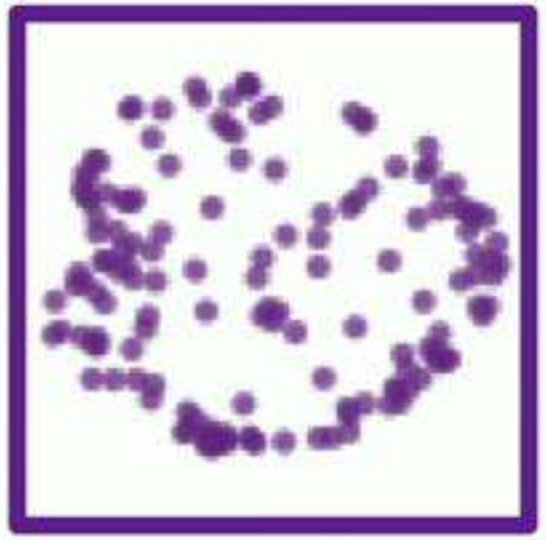
What type of *data* to visualize?

2D distribution, instead of high-dimensional images



What type of *data* to visualize?

2D distribution, instead of high-dimensional images



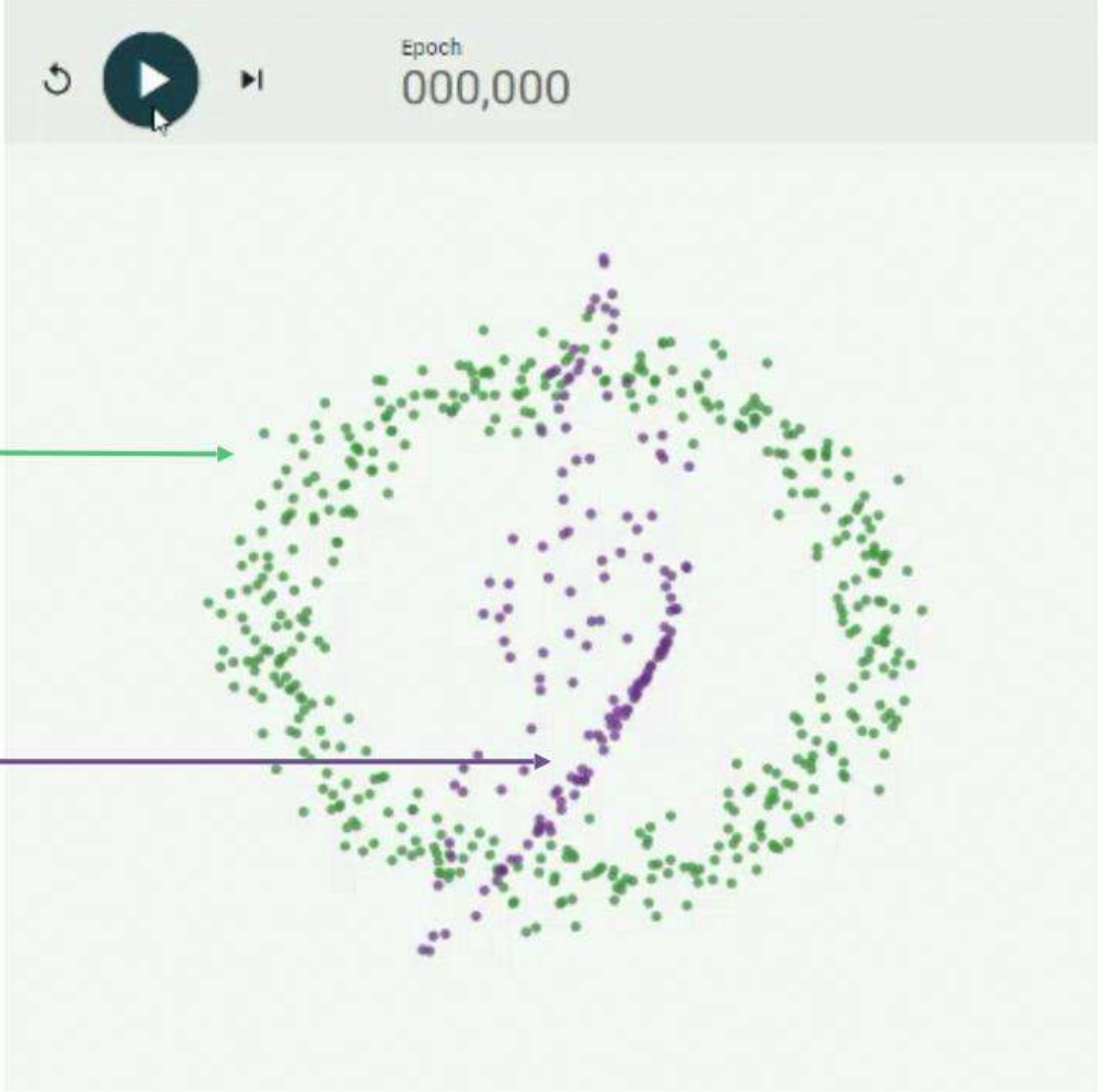
Why 2D data points?

1. To focus on GAN's main concepts
2. To easily visualize data distribution

VER. 0.1

Real
(green)

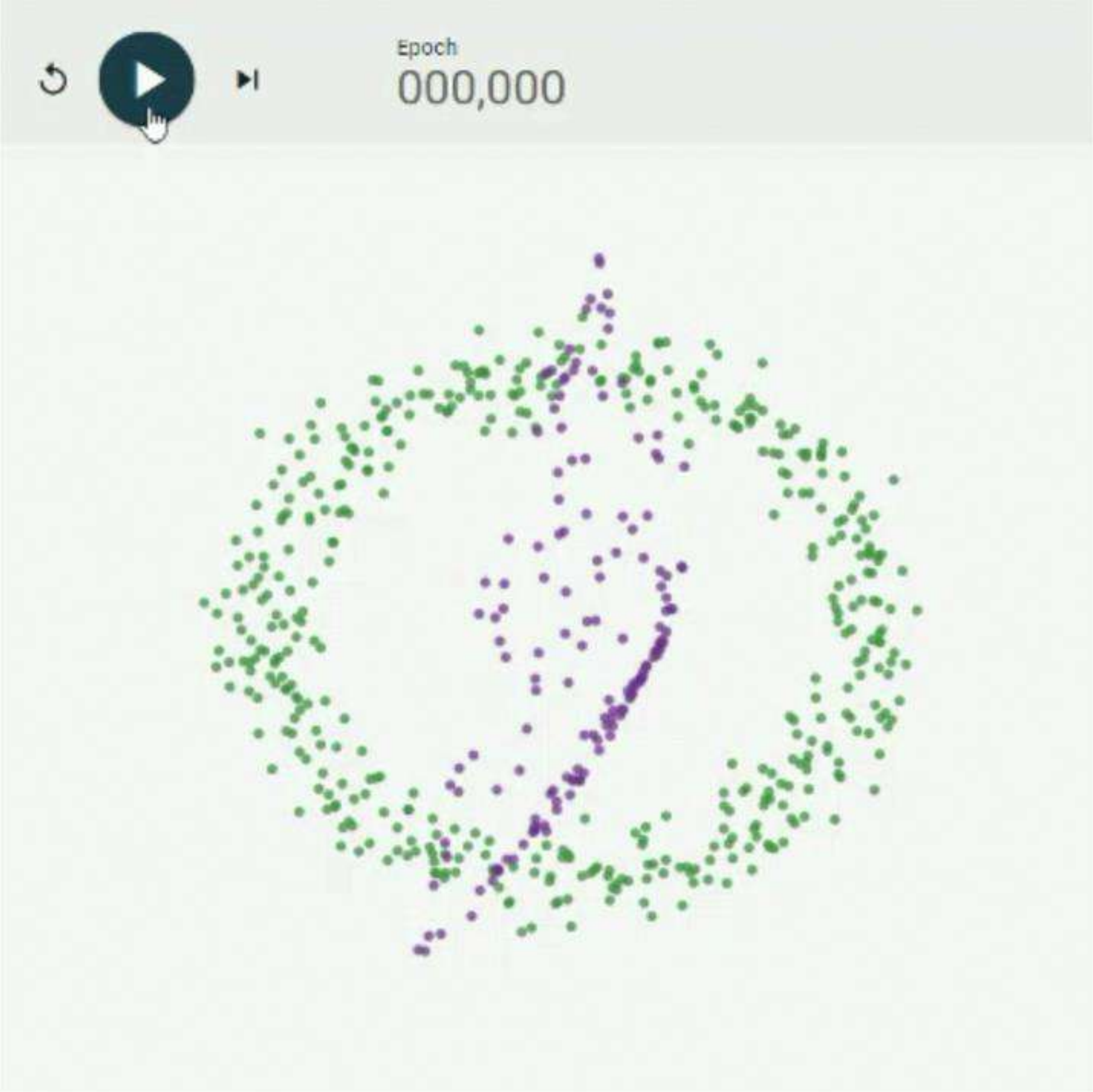
Generated
(purple)



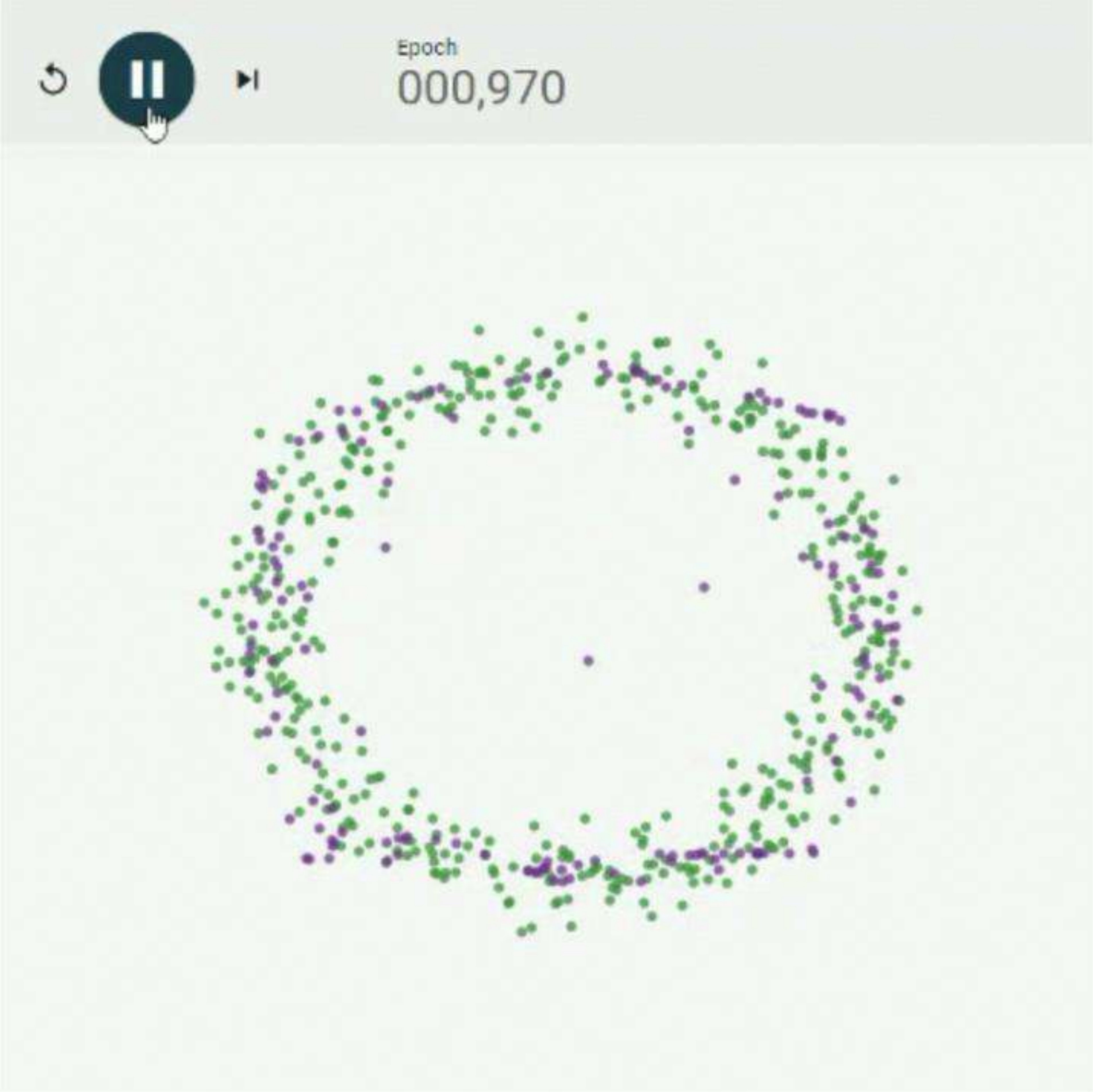
VER. 0.1

Real
(green)

Generated
(purple)



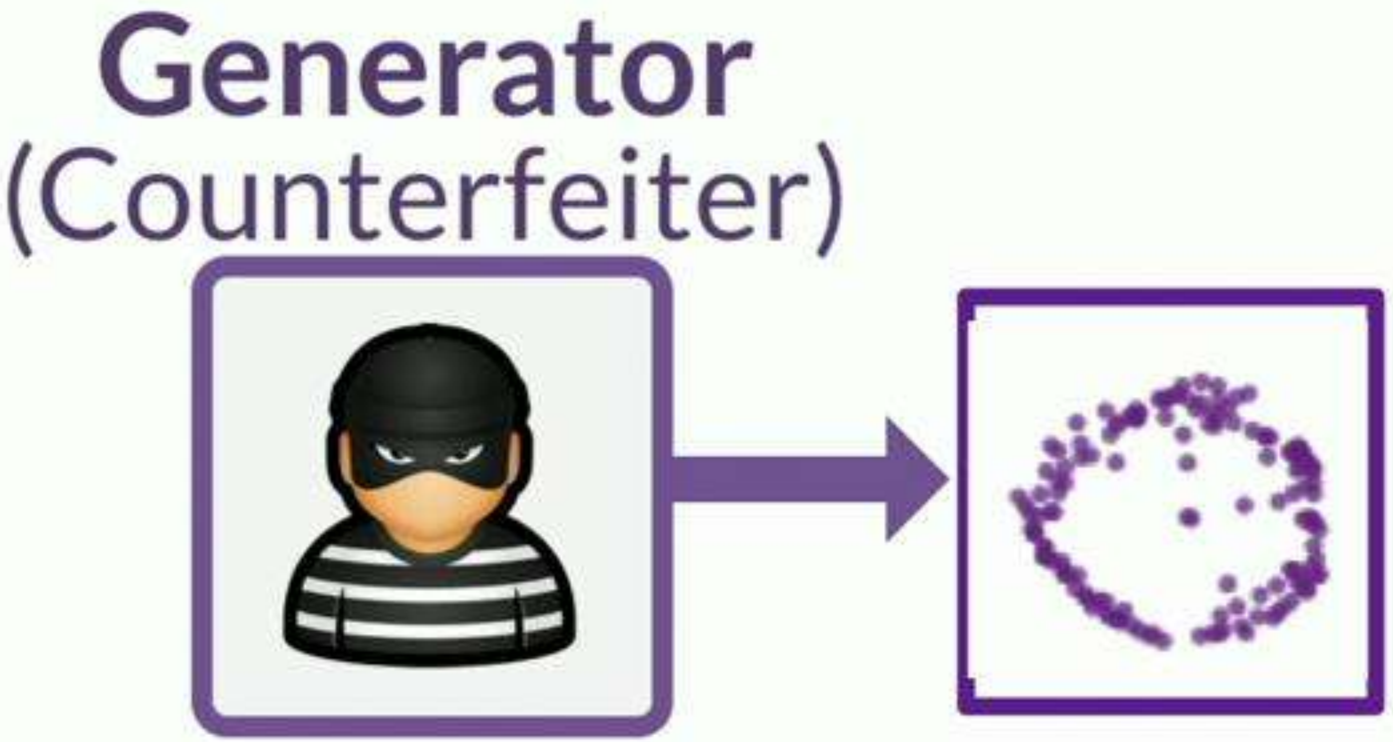
VER. 0.1



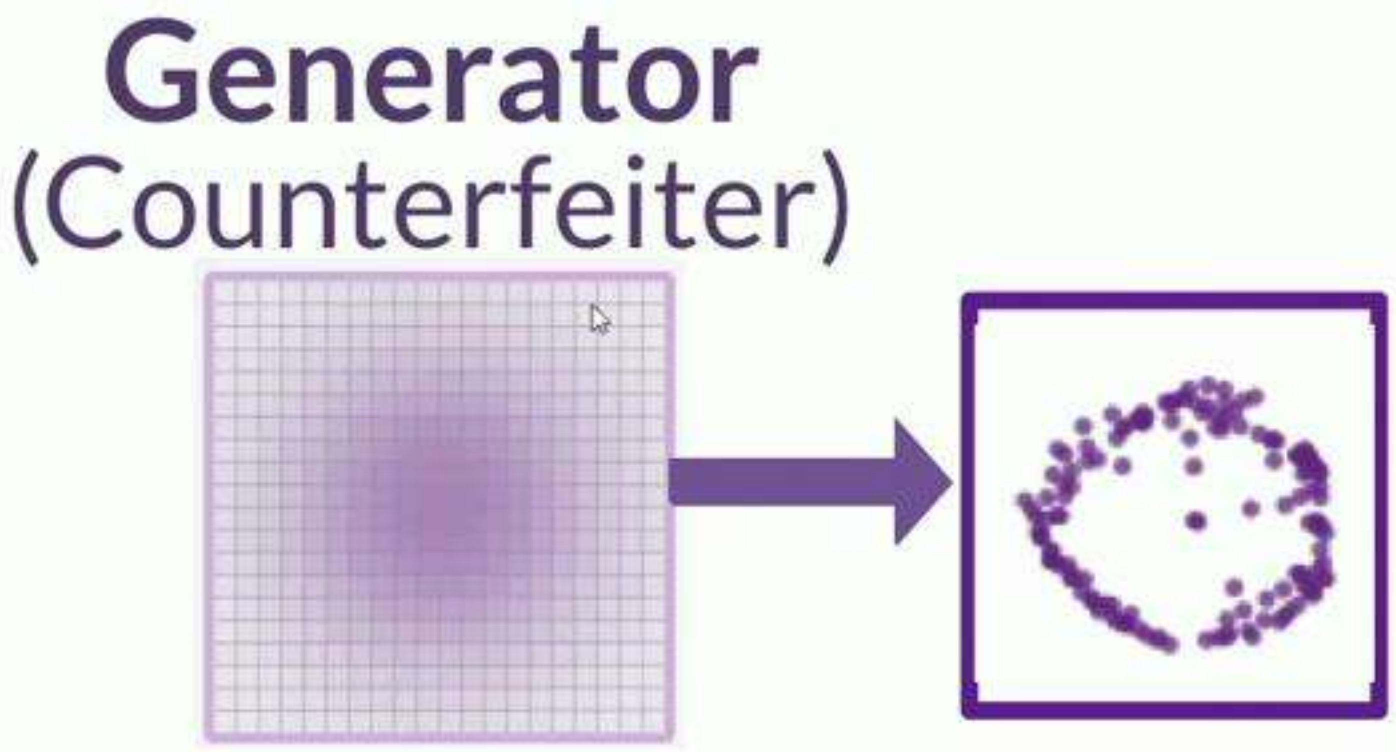
Real
(green)

Generated
(purple)

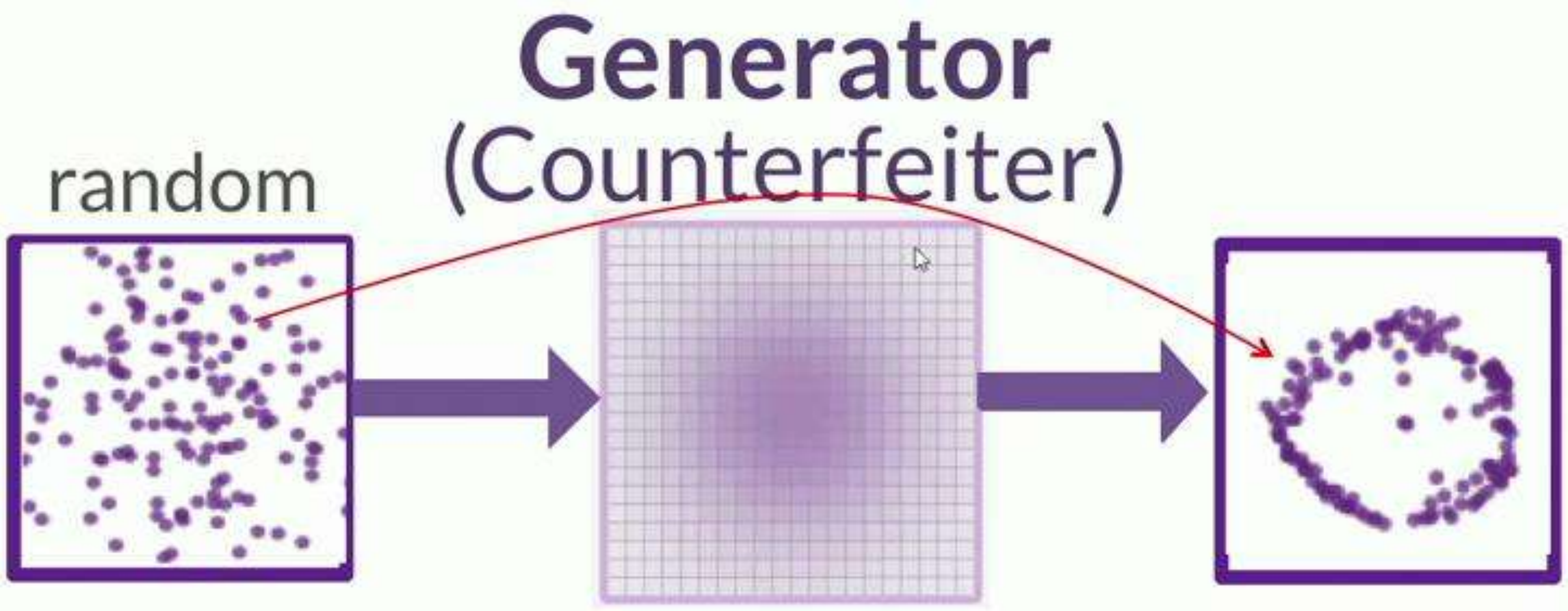
How to visually explain the *generator*?



How to visually explain the *generator*?

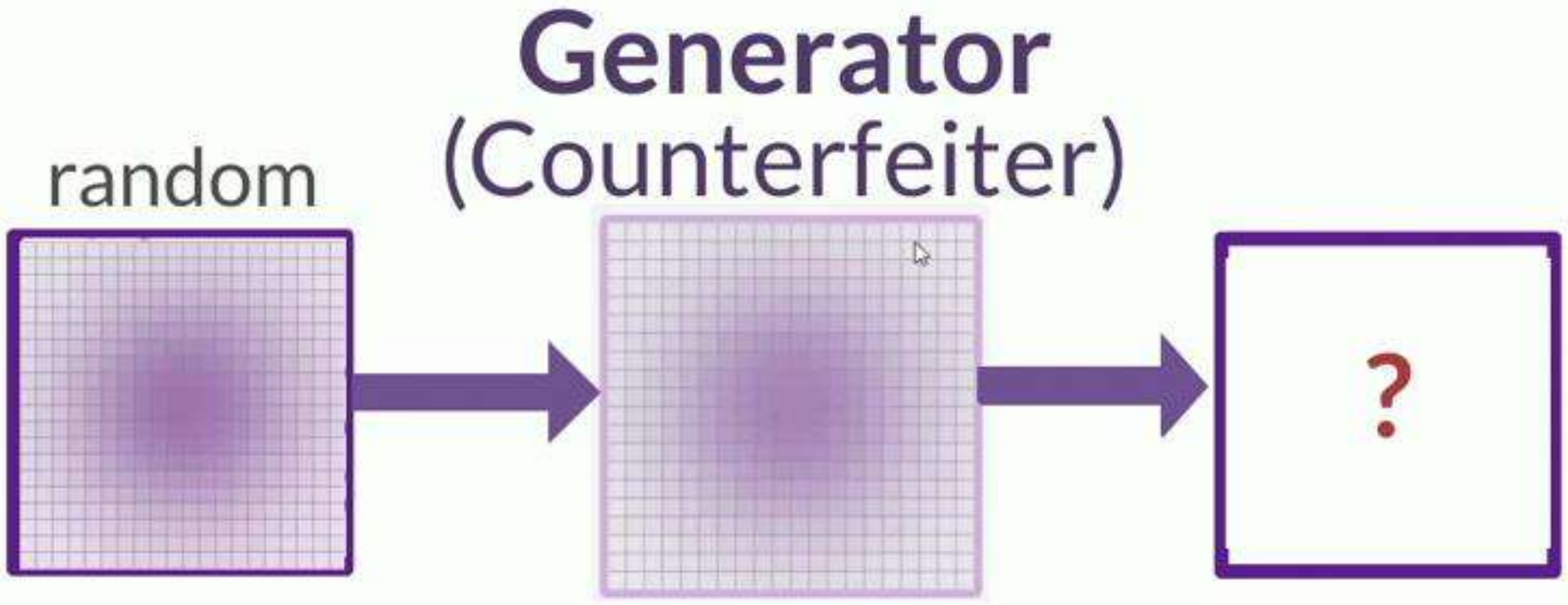


How to visually explain the *generator*?



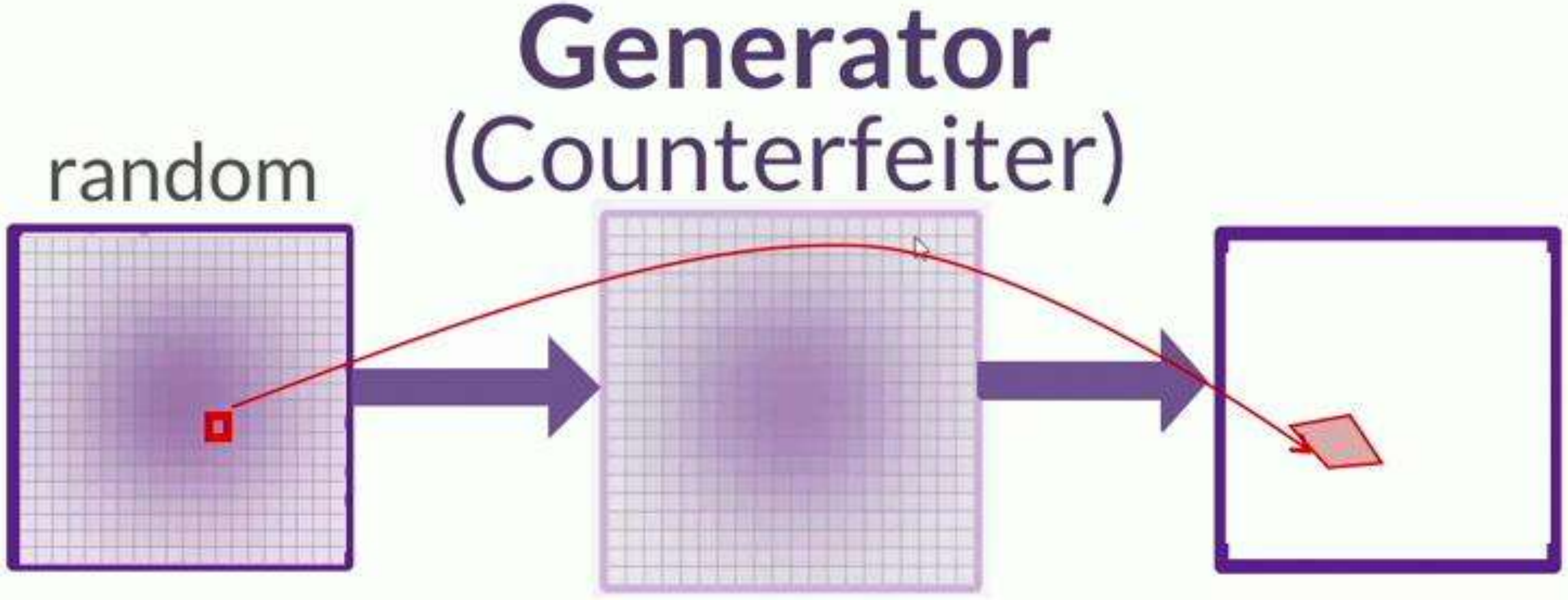
map an input point into a new position

How to visually explain the *generator*?



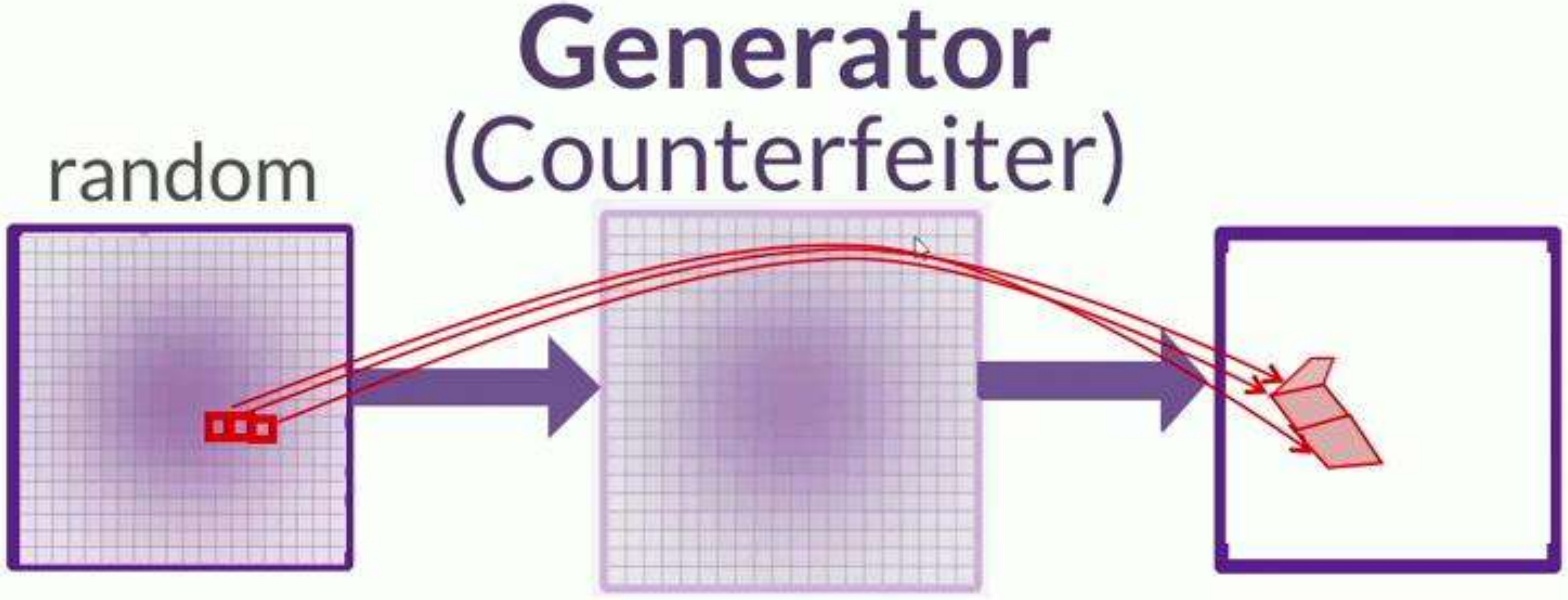
map an input point into a new position

How to visually explain the *generator*?



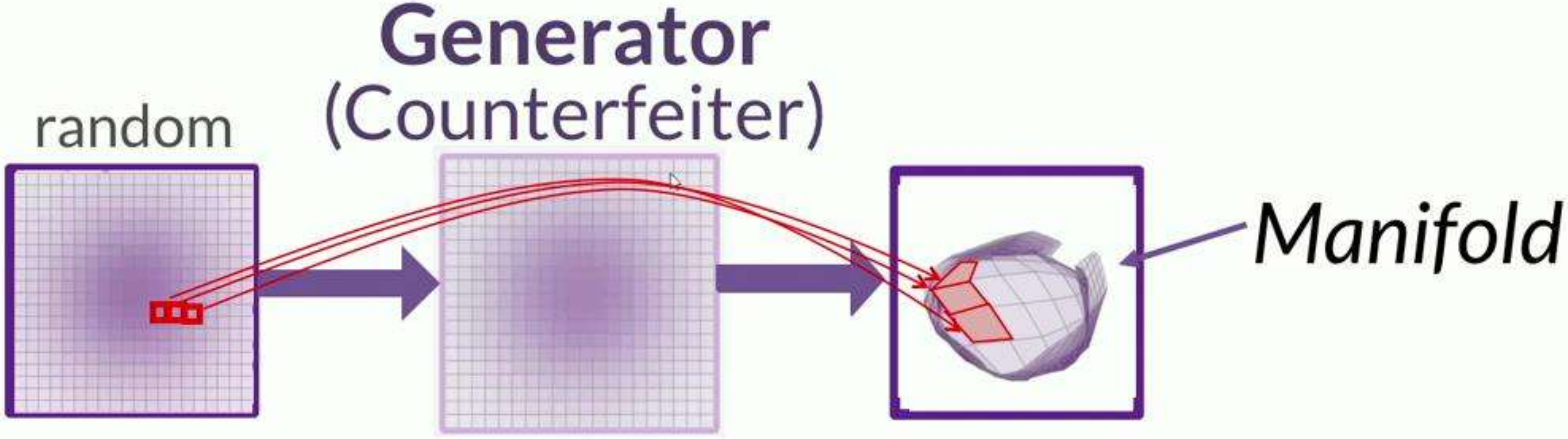
map an input point
into a new position

How to visually explain the *generator*?



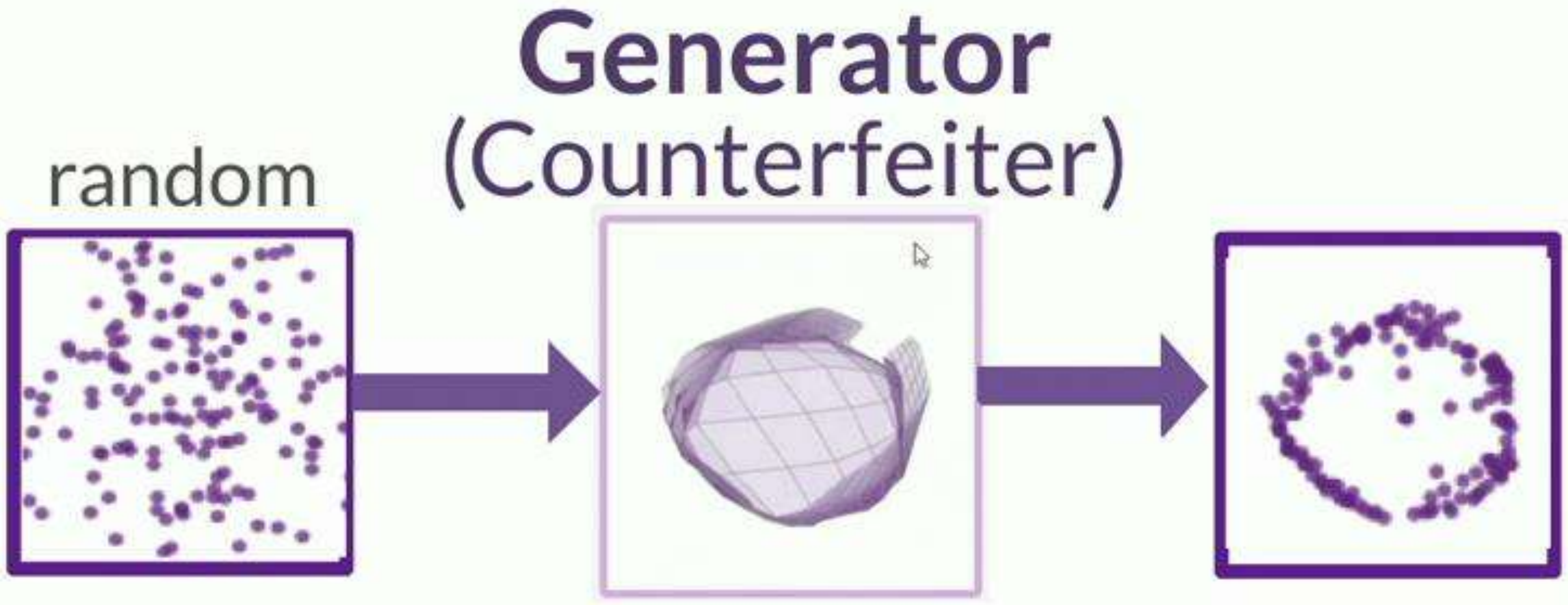
map an input point into a new position

How to visually explain the *generator*?



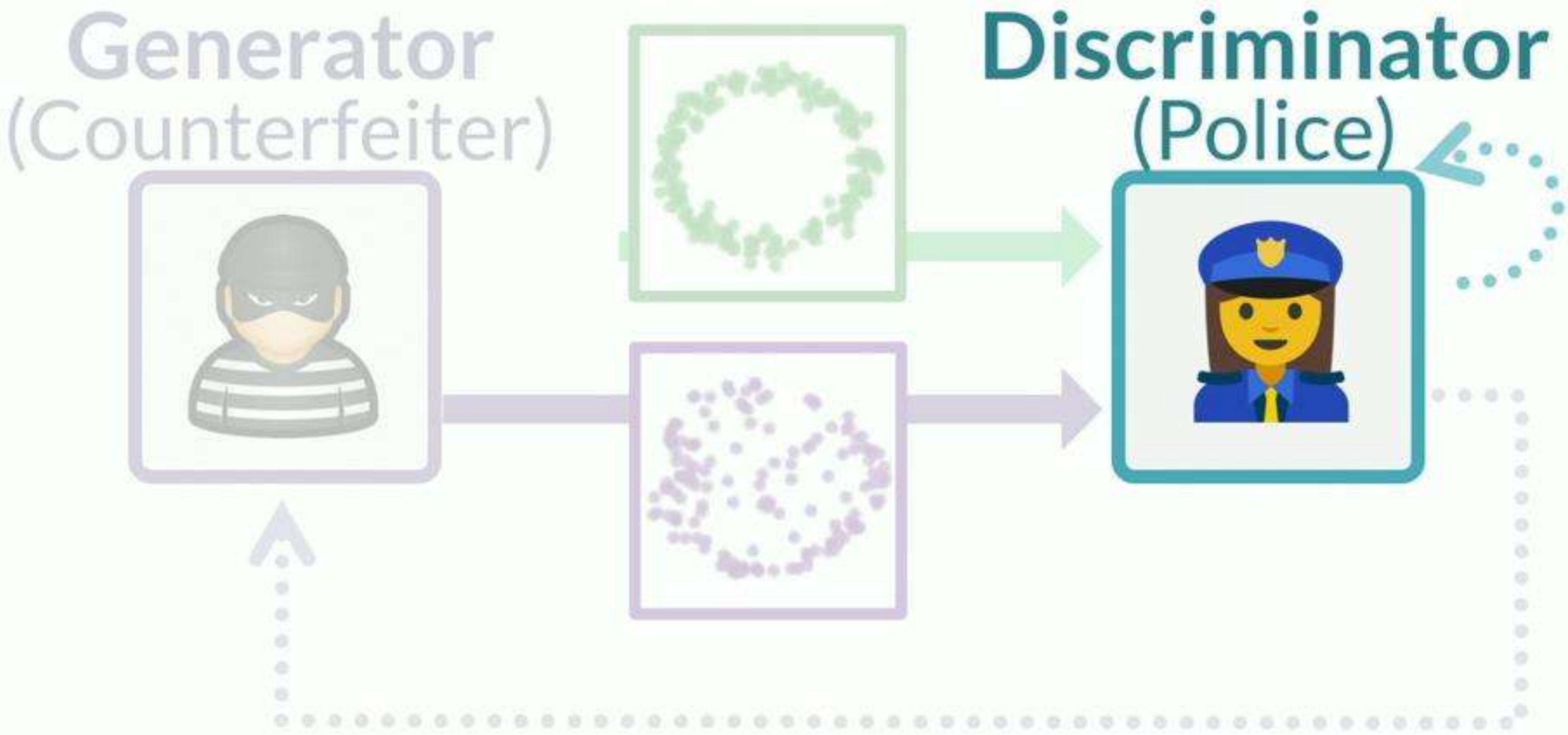
map an input point
into a new position

How to visually explain the *generator*?



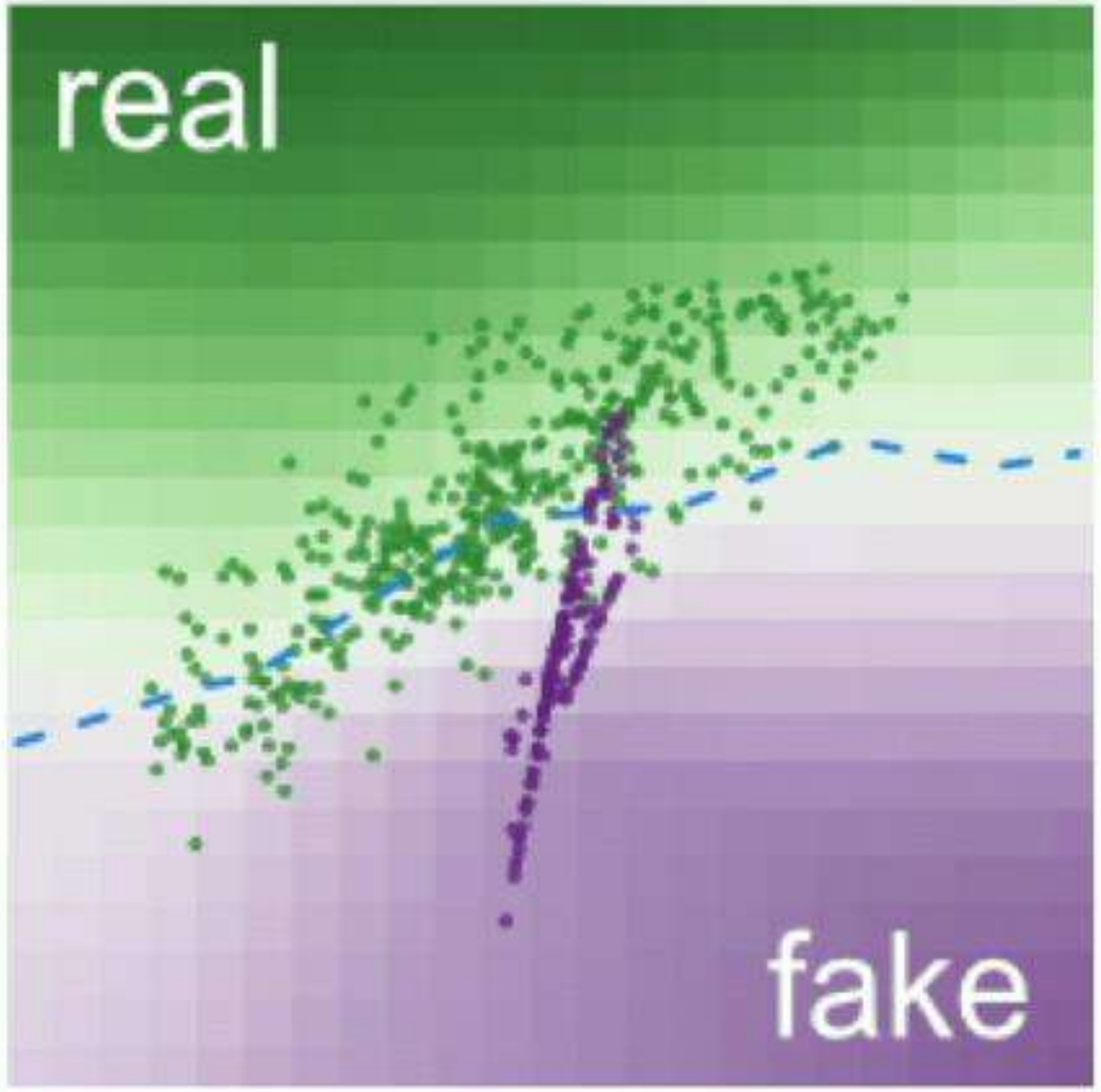
map an input point into a new position

How to visualize the *discriminator*?



How to visualize the *discriminator*?

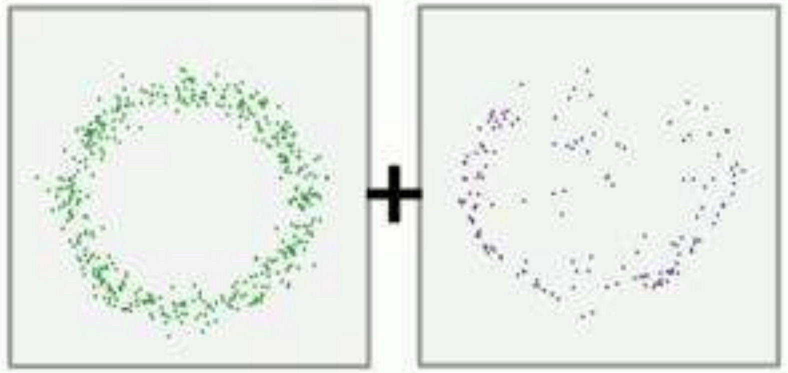
2D heatmap, to represent binary classification



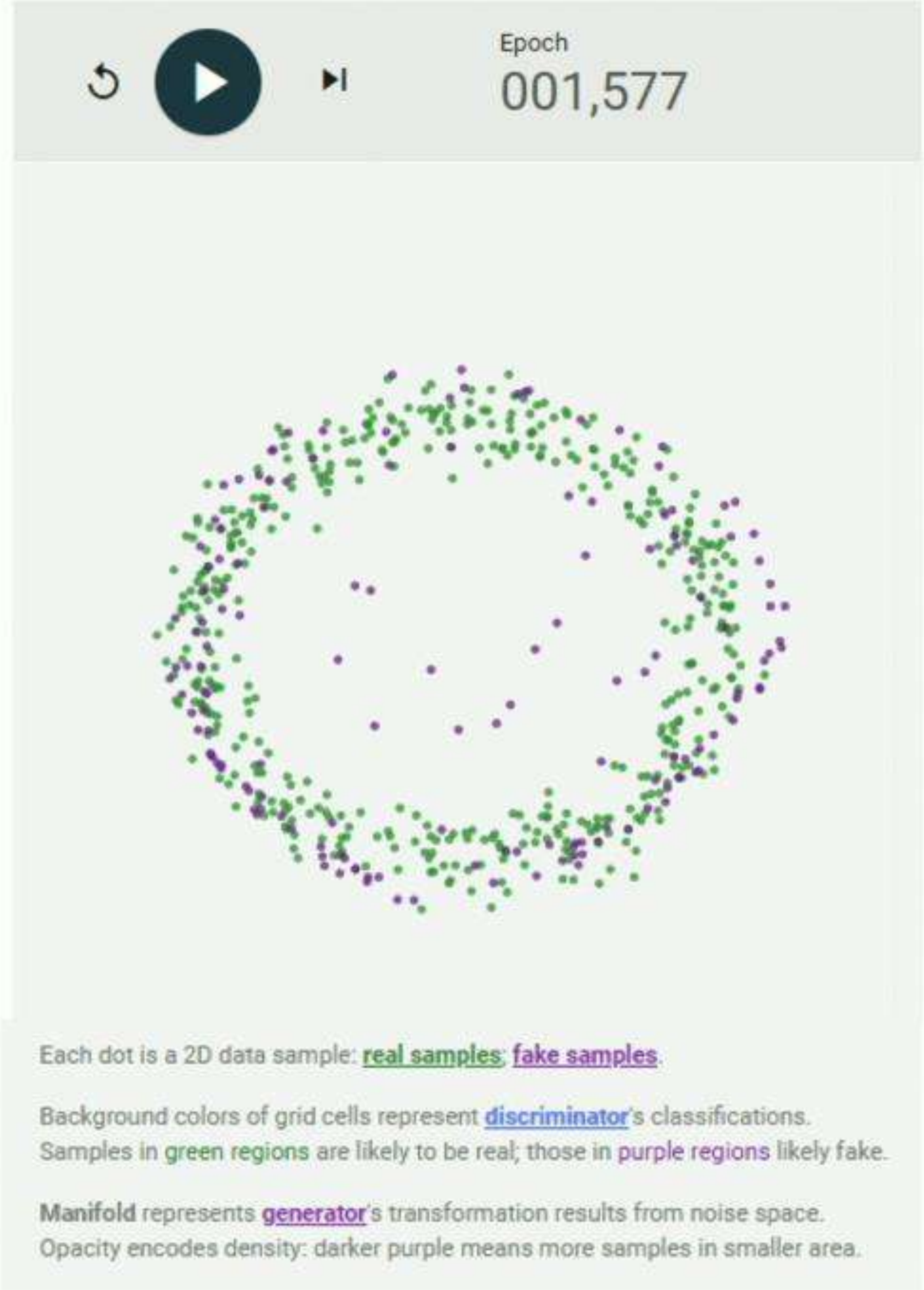
Data points in this region are likely real.

Data points are likely fake.

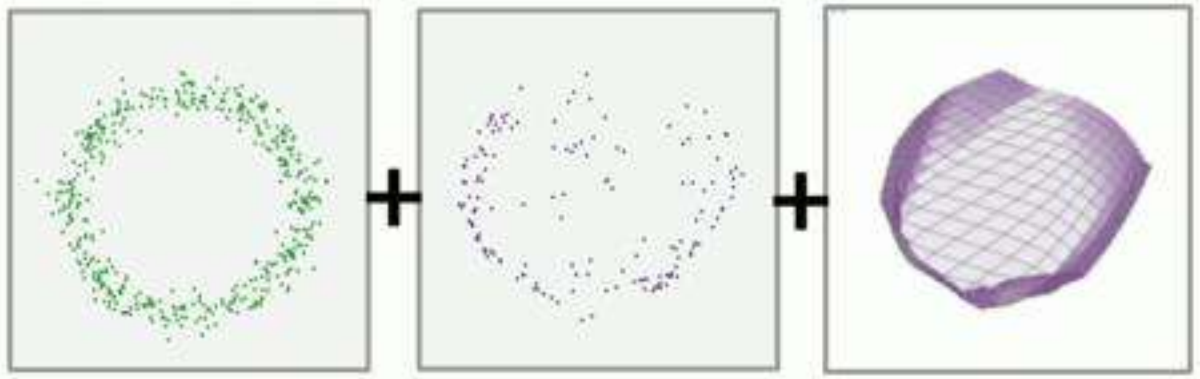
VER. 0.5



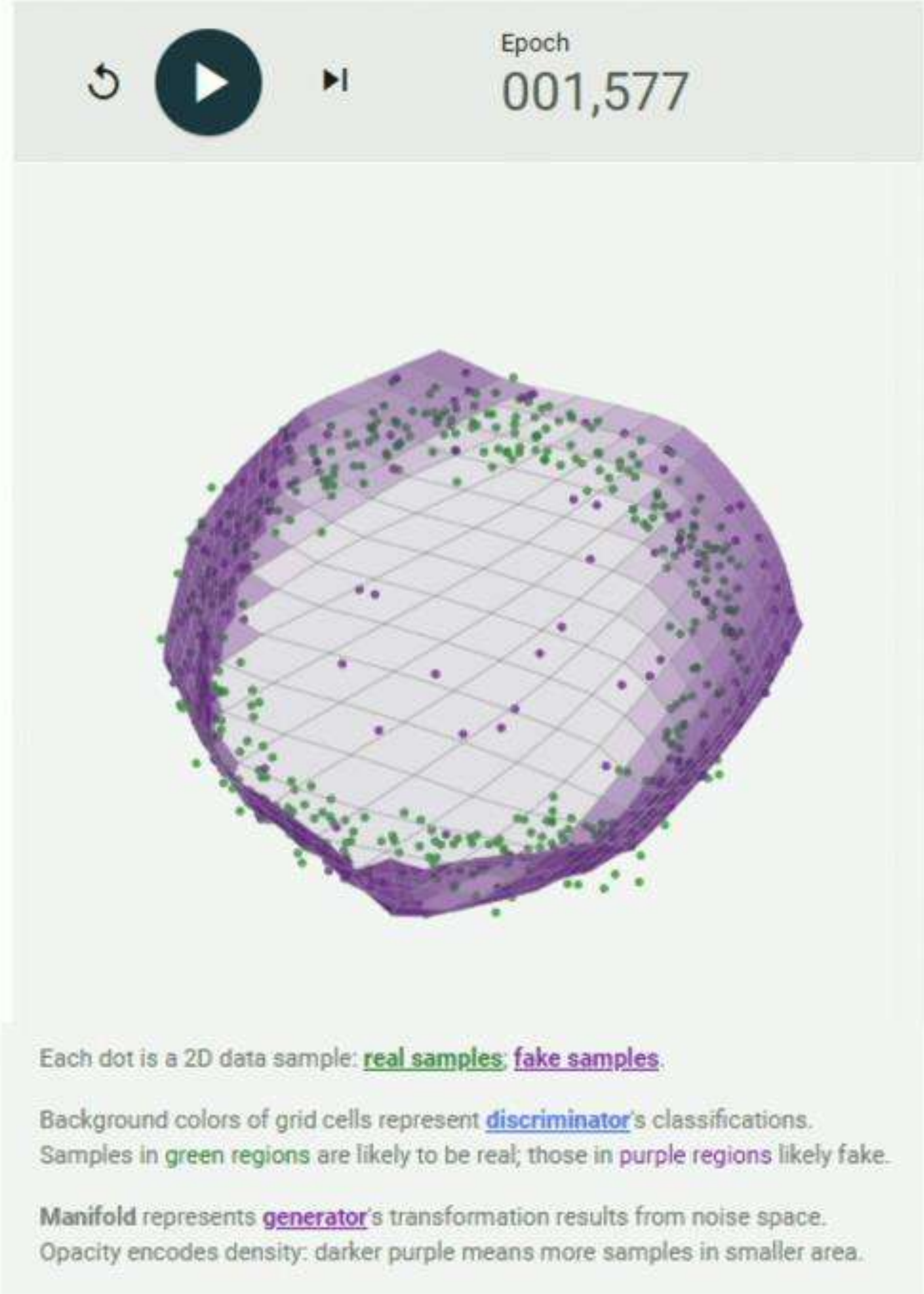
real data + fake data



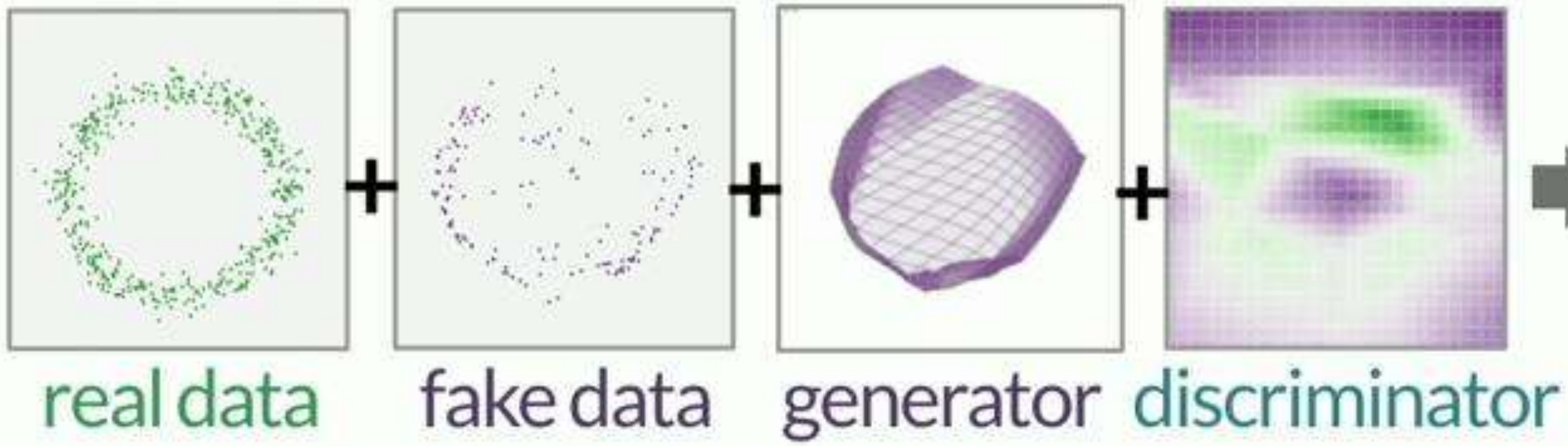
VER. 0.5

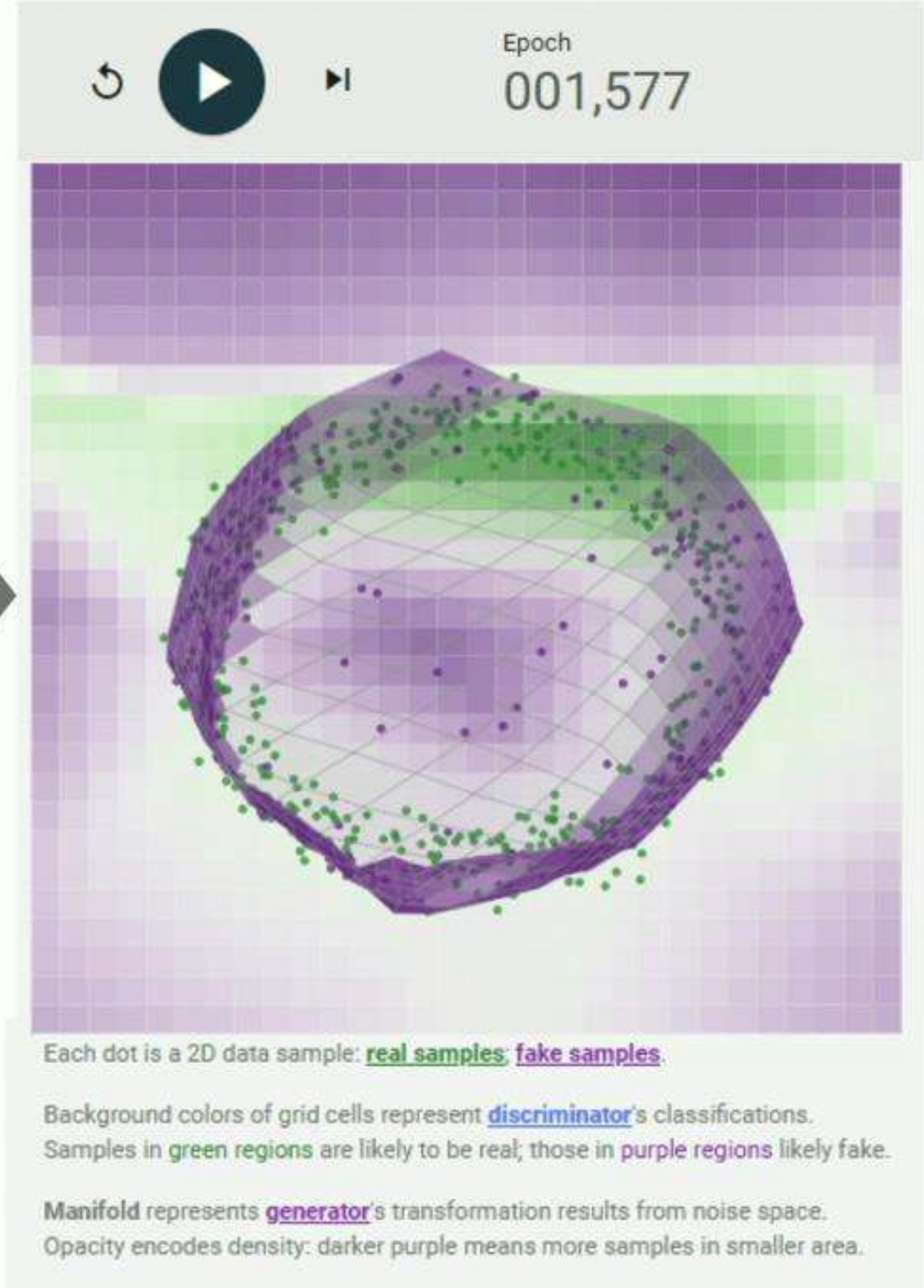


real data　fake data　generator

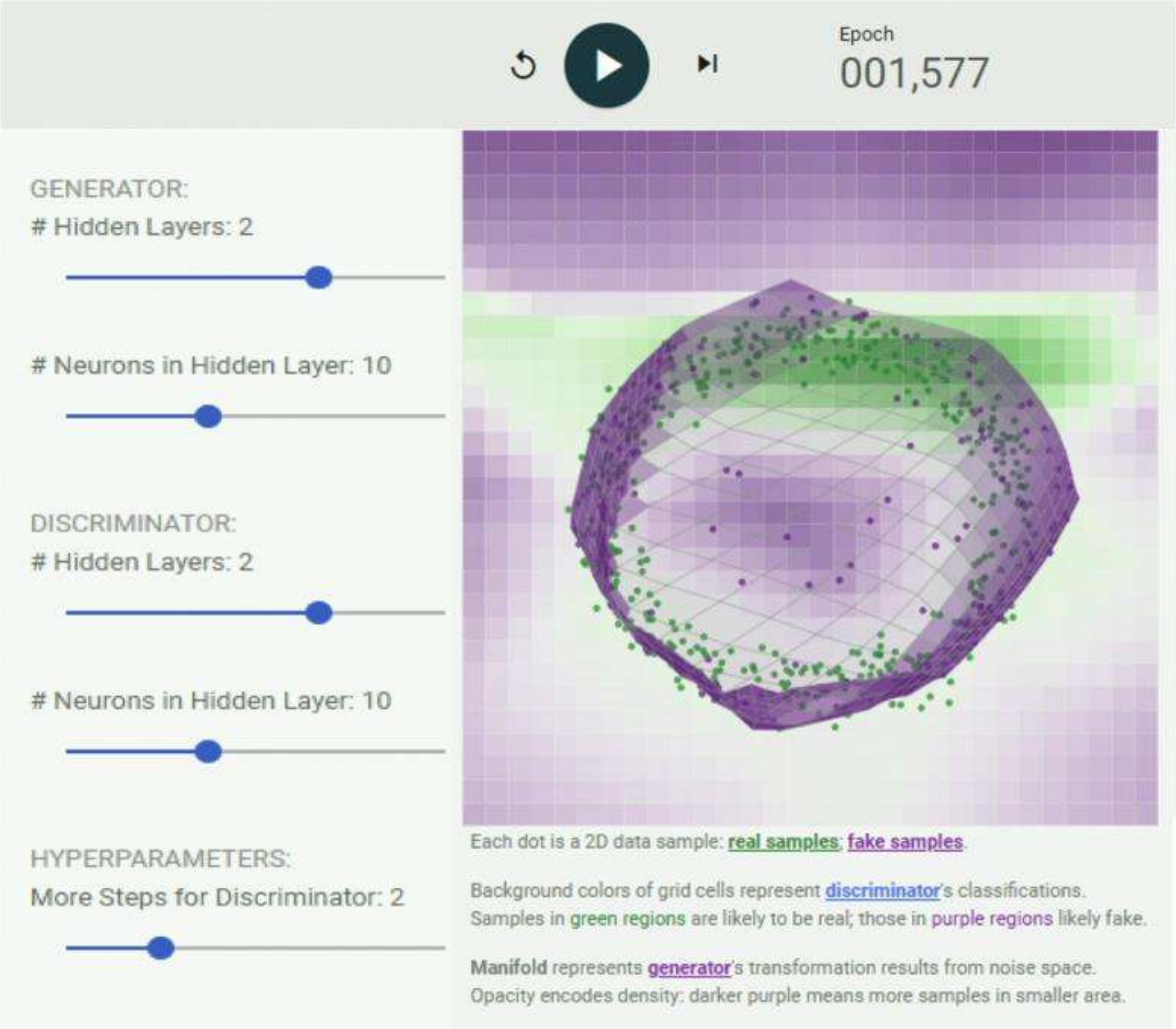


VER. 0.5

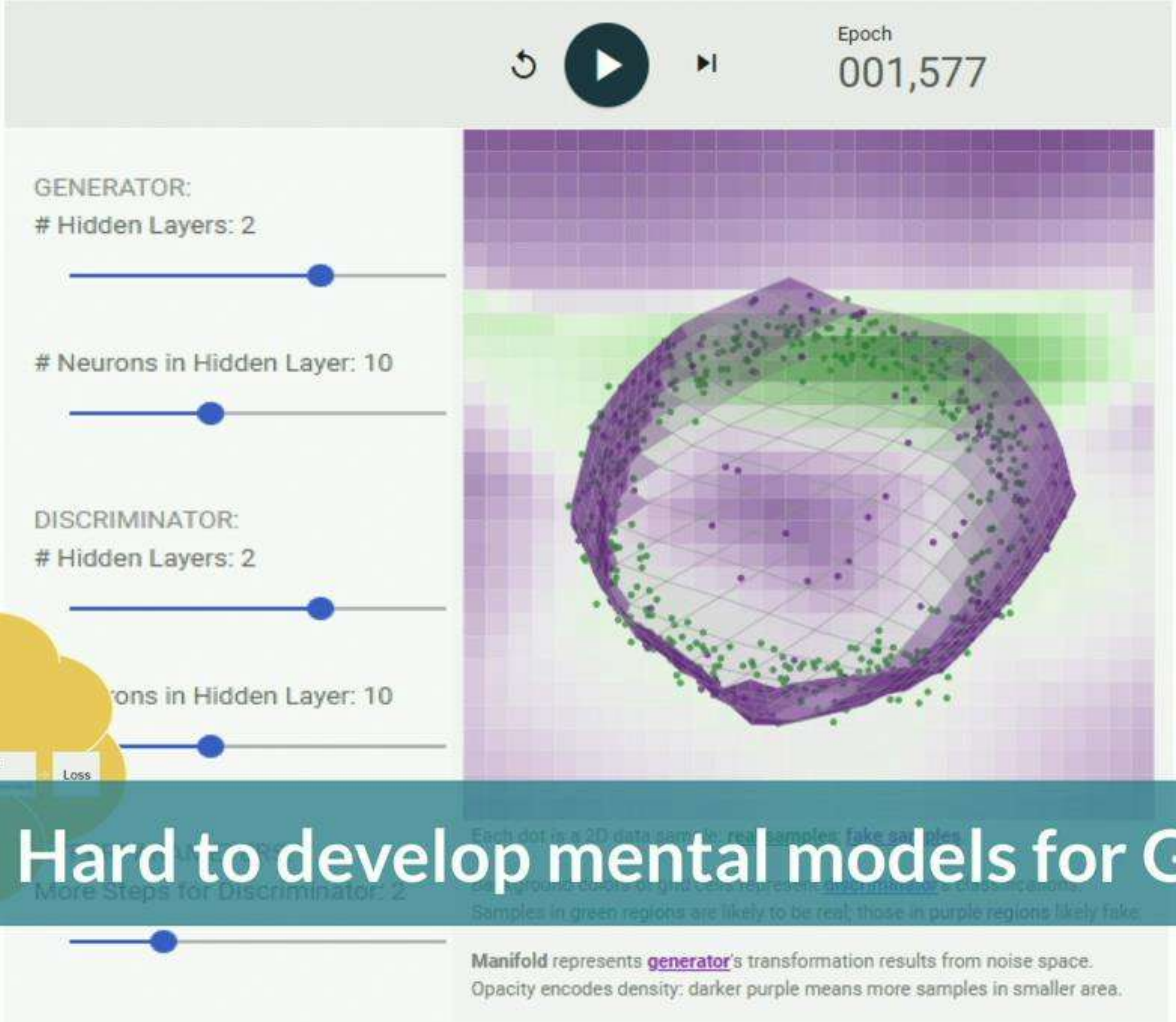

real data
+
fake data
+
generator
+
discriminator

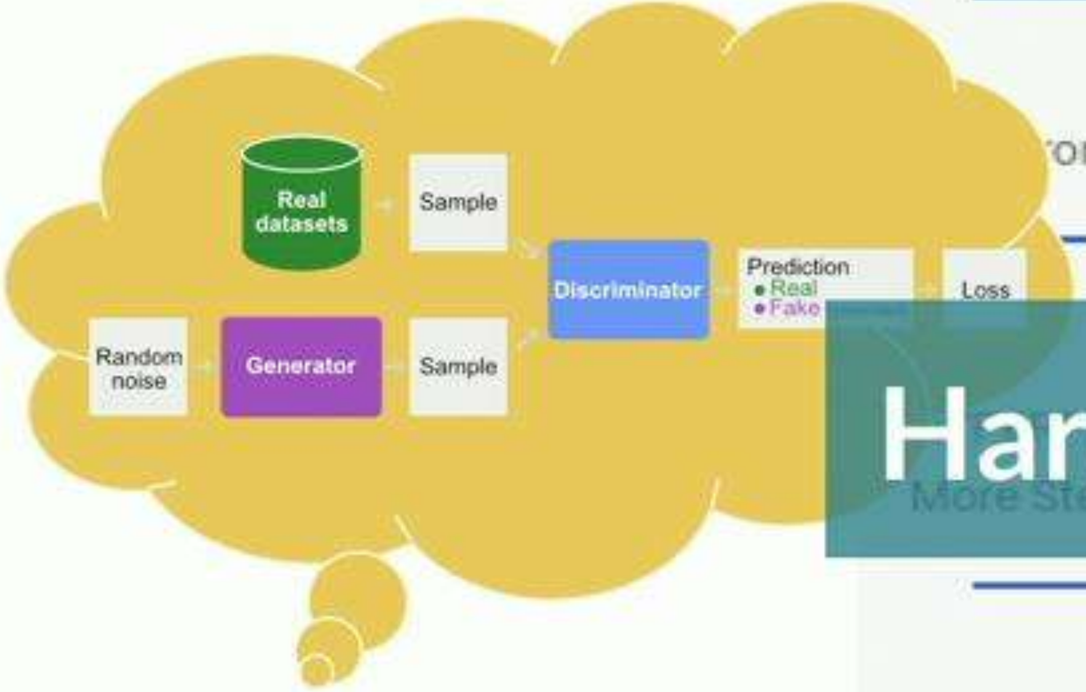

Epoch
001,577
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.

VER. 0.5


Epoch
001,577
GENERATOR:
Hidden Layers: 2
Neurons in Hidden Layer: 10
DISCRIMINATOR:
Hidden Layers: 2
Neurons in Hidden Layer: 10
HYPERPARAMETERS:
More Steps for Discriminator: 2
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.

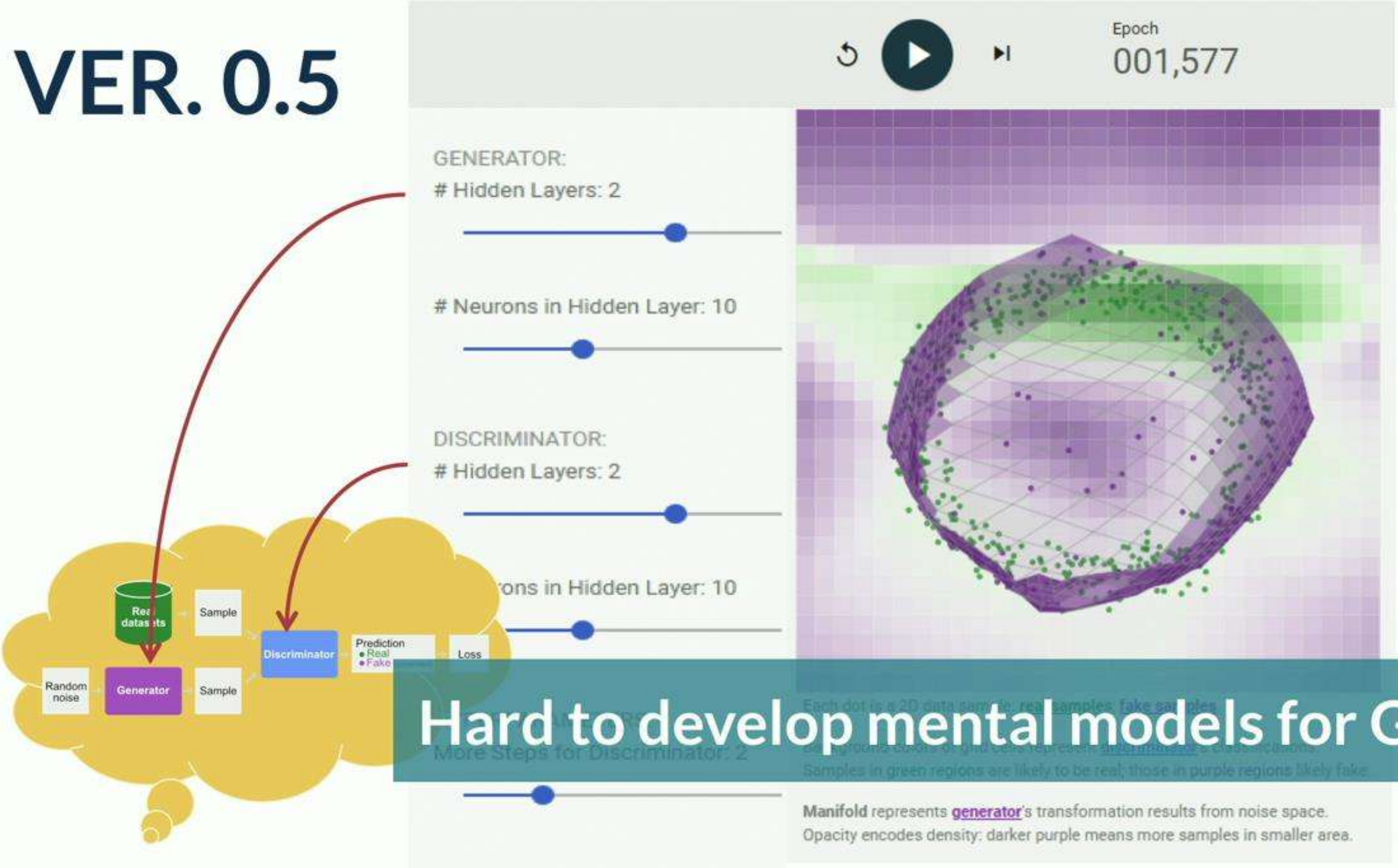
VER. 0.5



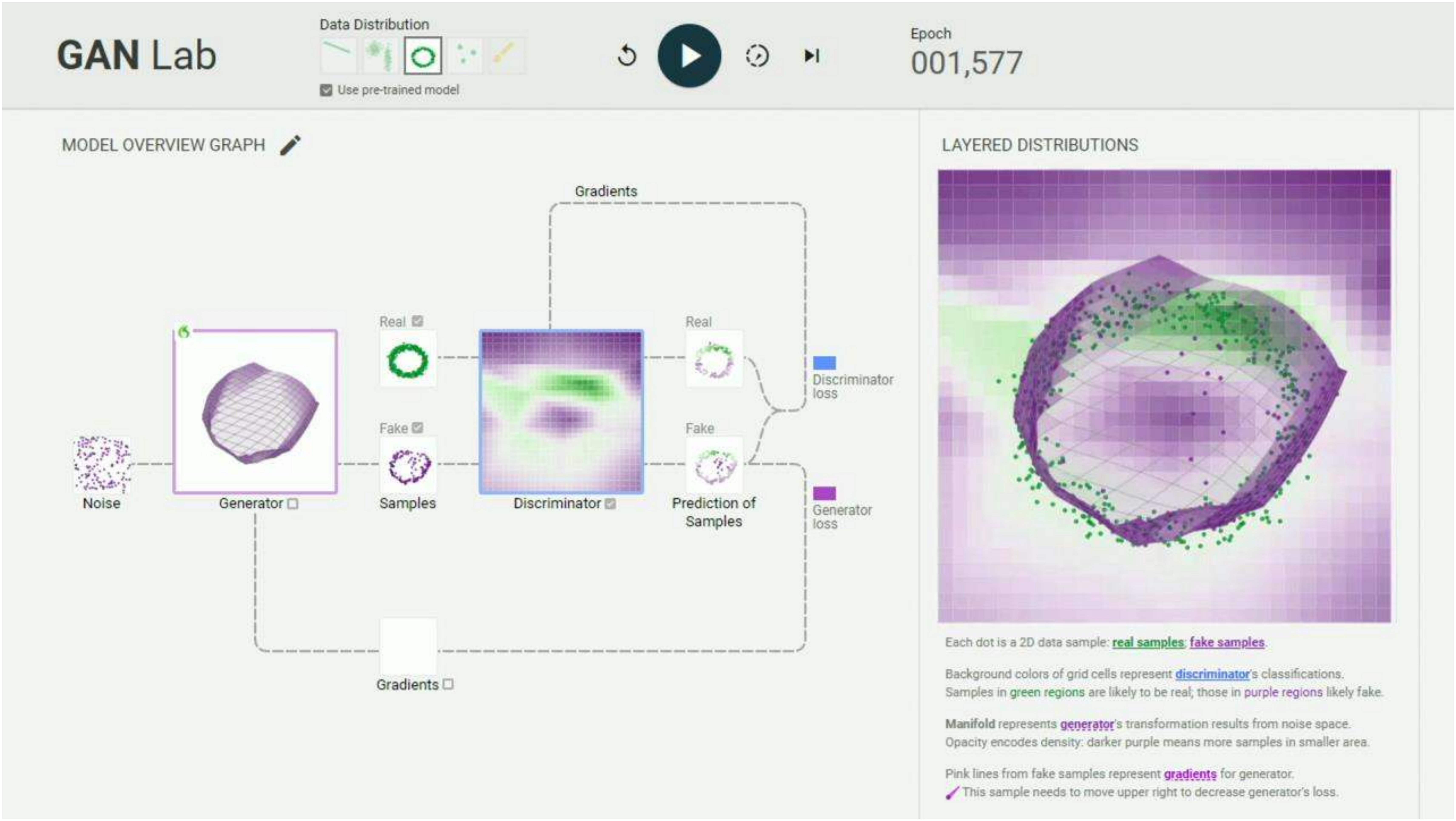


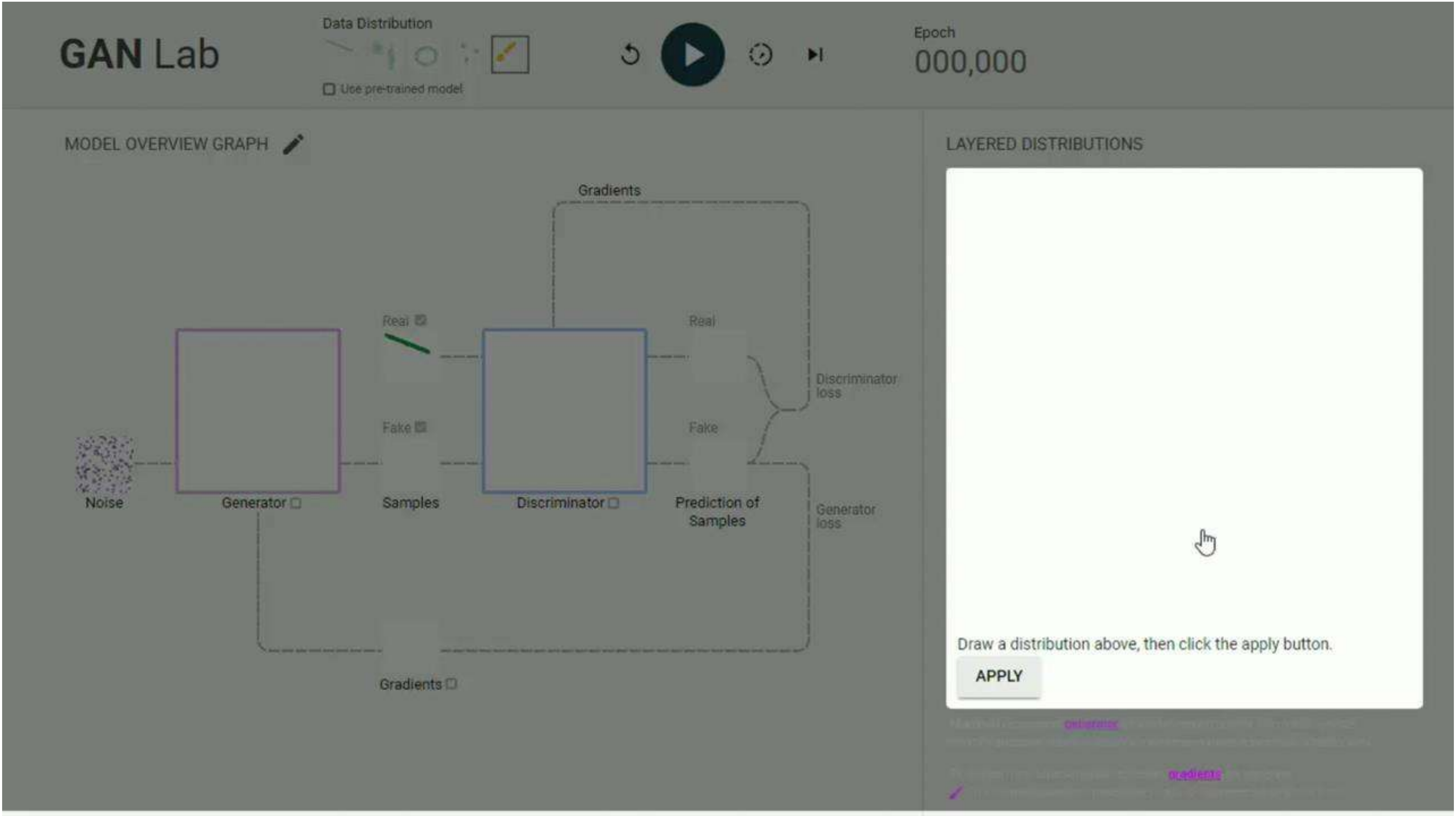
Hard to develop mental models for GANs

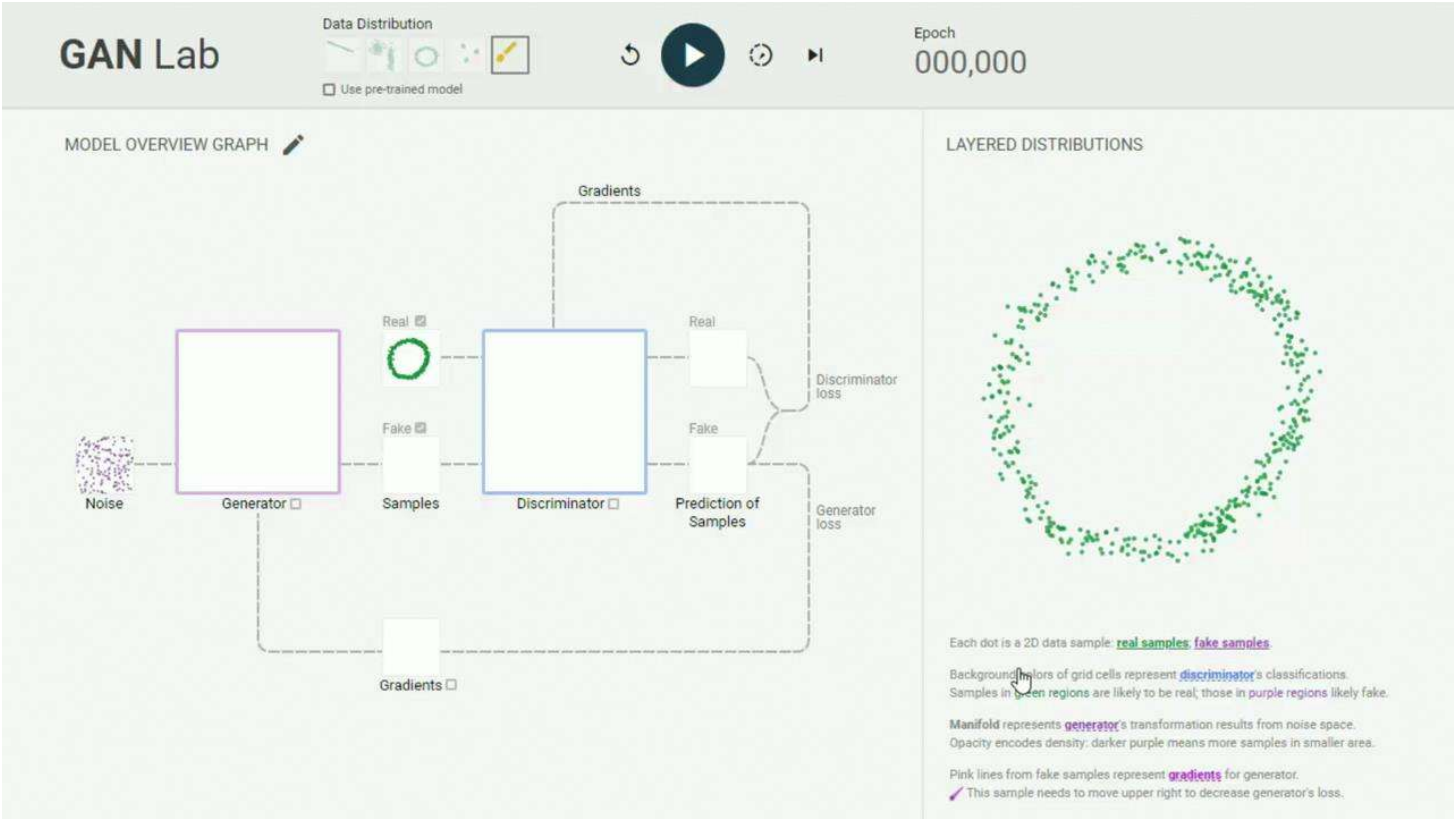
VER. 0.5

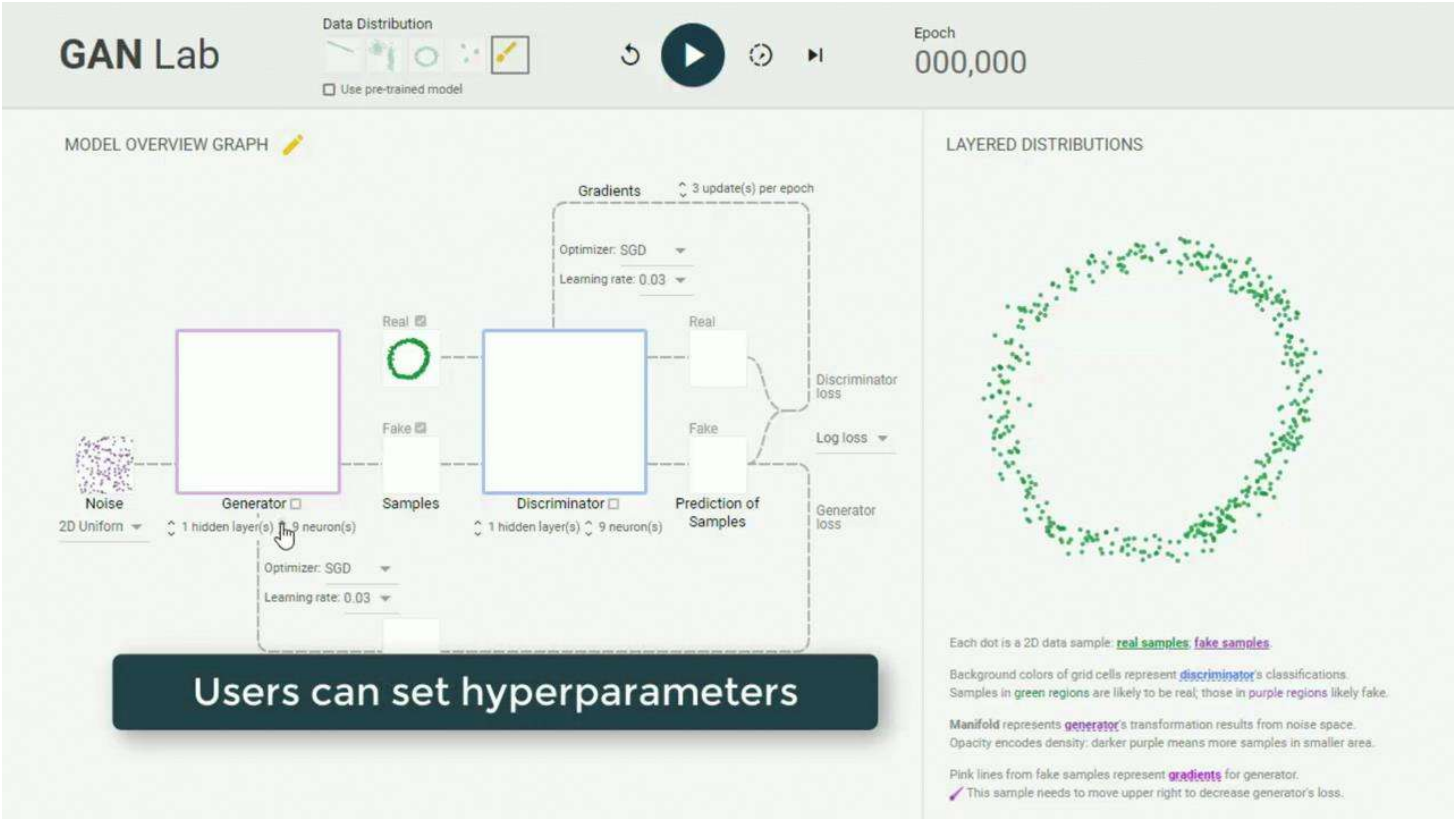


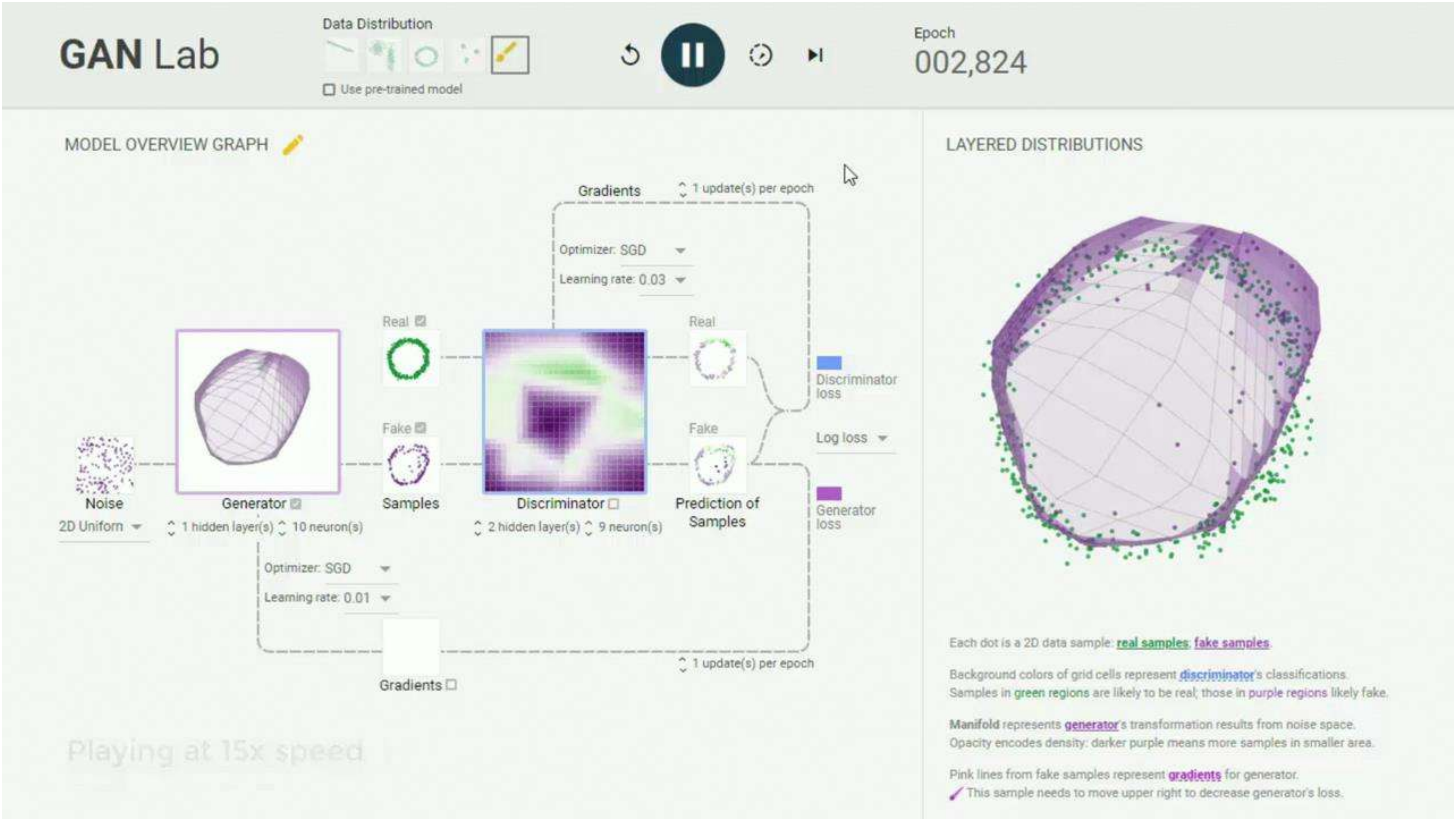
Hard to develop mental models for GANs


GAN Lab
Data Distribution
Use pre-trained model
Epoch
001,577
MODEL OVERVIEW GRAPH
Gradients
Real
Fake
Noise
Generator
Samples
Discriminator
Prediction of Samples
Discriminator loss
Generator loss
Gradients
LAYERED DISTRIBUTIONS
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.


GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,000
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
Real
Fake
Real
Fake
Discriminator loss
Generator loss
Noise
Generator
Samples
Discriminator
Prediction of Samples
Gradients
Draw a distribution above, then click the apply button.
APPLY


GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,000
MODEL OVERVIEW GRAPH
Gradients
Real
Fake
Noise
Generator
Samples
Discriminator
Real
Fake
Prediction of Samples
Discriminator loss
Generator loss
Gradients
LAYERED DISTRIBUTIONS
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.


GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,000
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.03
Real
Fake
Discriminator loss
Log loss
Generator loss
Noise
2D Uniform
Generator
1 hidden layer(s)
9 neuron(s)
Samples
Discriminator
1 hidden layer(s)
9 neuron(s)
Prediction of Samples
Optimizer: SGD
Learning rate: 0.03
Users can set hyperparameters
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

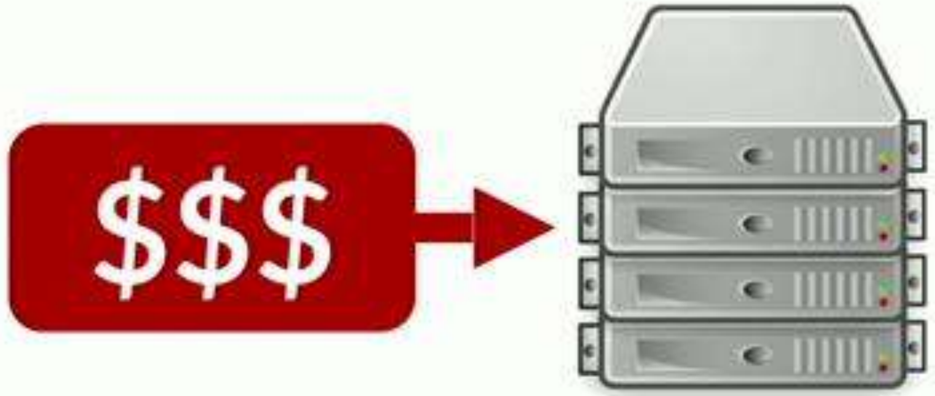


GAN Lab
Data Distribution
Use pre-trained model
Epoch
002,824
MODEL OVERVIEW GRAPH
Gradients
1 update(s) per epoch
Optimizer: SGD
Learning rate: 0.03
Real
Fake
Noise
2D Uniform
Generator
1 hidden layer(s)
10 neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Discriminator loss
Log loss
Generator loss
Optimizer: SGD
Learning rate: 0.01
Gradients
1 update(s) per epoch
Playing at 15x speed
LAYERED DISTRIBUTIONS
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

GAN Lab broadens education access

Conventional Deep Learning Visualization

Visualization
in JavaScript



Model Training
in Python with GPU

GAN Lab broadens education access

Everything done in browser, powered by TensorFlow.js

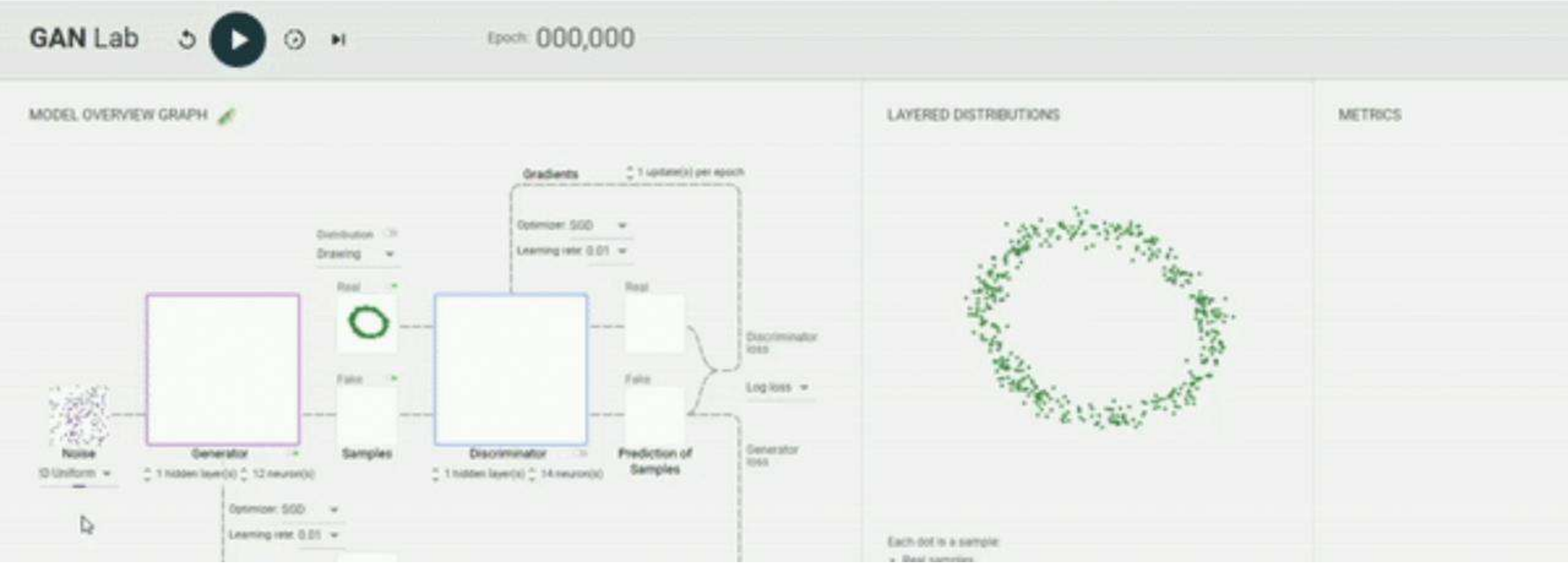
Visualization
in JavaScript

Model Training
also in JavaScript

Accelerated by WebGL

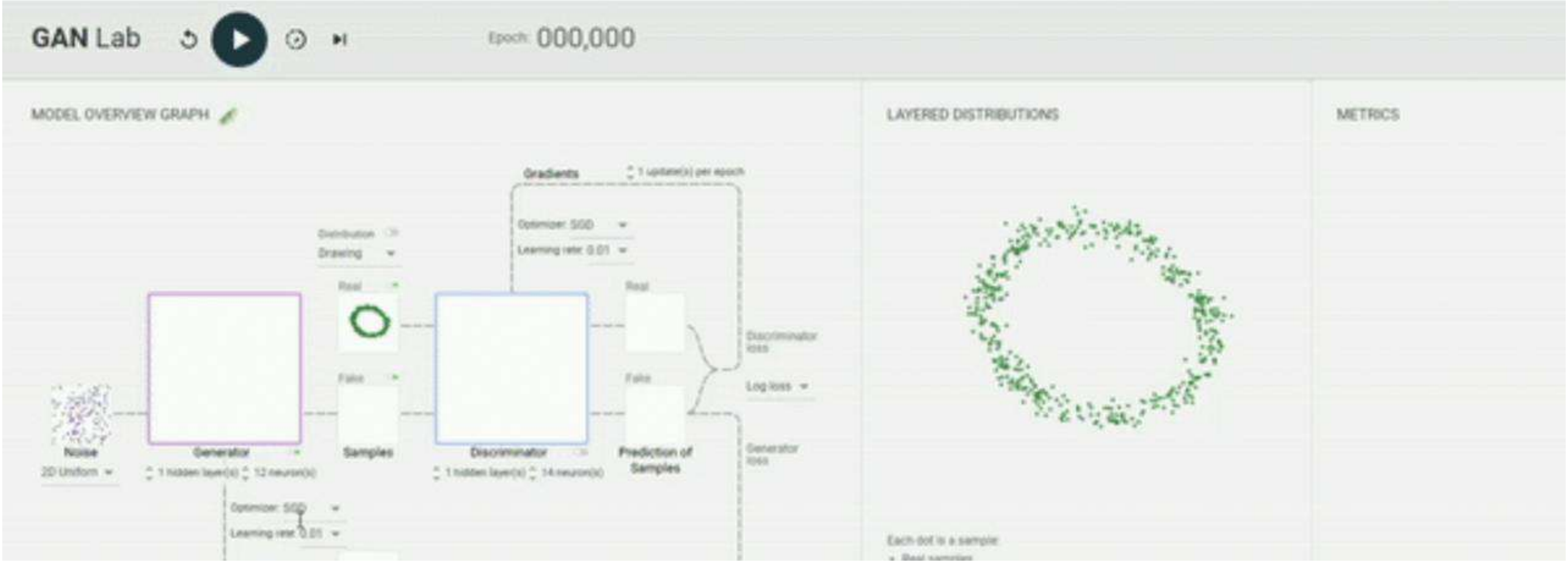
GAN Lab is Live! Try at bit.ly/gan-lab

30K visitors, 135 countries 1.9K Likes 800+ Retweets



GAN Lab is Live! Try at bit.ly/gan-lab

30K visitors, 135 countries ♥ 1.9K Likes 800+ Retweets



Interpretable AI via Visual Analytics

Understanding Industry-Scale Models

ActiVis - Activation analysis by subsets

Interactive Learning of Complex Models

GAN Lab - Experimentation with GANs

Research Landscape

Survey, Gamut

TVCG 2018

Visual Analytics in Deep Learning

An Interrogative Survey for the Next Frontiers


Fred
Hohman
Georgia Tech


Minsuk
Kahng
Georgia Tech


Robert
Pienta
Symantec


Polo
Chau
Georgia Tech

Georgia Tech

Symantec

Visual Analytics in Deep Learning | Interrogative Survey Overview

§4 WHY

Why would one want to use visualization in deep learning?

Interpretability & Explainability
Debugging & Improving Models
Comparing & Selecting Models
Teaching Deep Learning Concepts

§6 WHAT

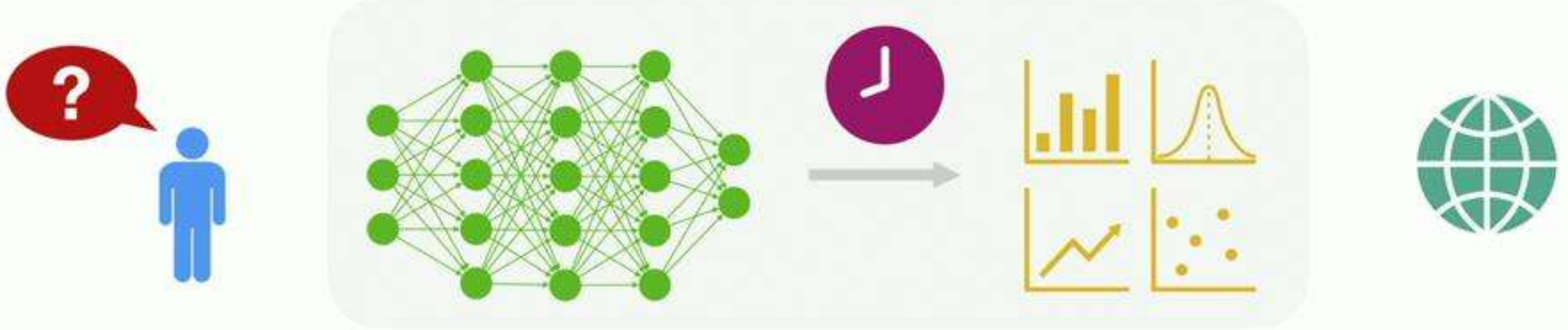
What data, features, and relationships in deep learning can be visualized?

Computational Graph & Network Architecture
Learned Model Parameters
Individual Computational Units
Neurons In High-dimensional Space
Aggregated Information

§8 WHEN

When in the deep learning process is visualization used?

During Training
After Training



§5 WHO

Who would use and benefit from visualizing deep learning?

Model Developers & Builders
Model Users
Non-experts

§7 HOW

How can we visualize deep learning data, features, and relationships?

Node-link Diagrams for Network Architecture
Dimensionality Reduction & Scatter Plots
Line Charts for Temporal Metrics
Instance-based Analysis & Exploration
Interactive Experimentation

§9 WHERE

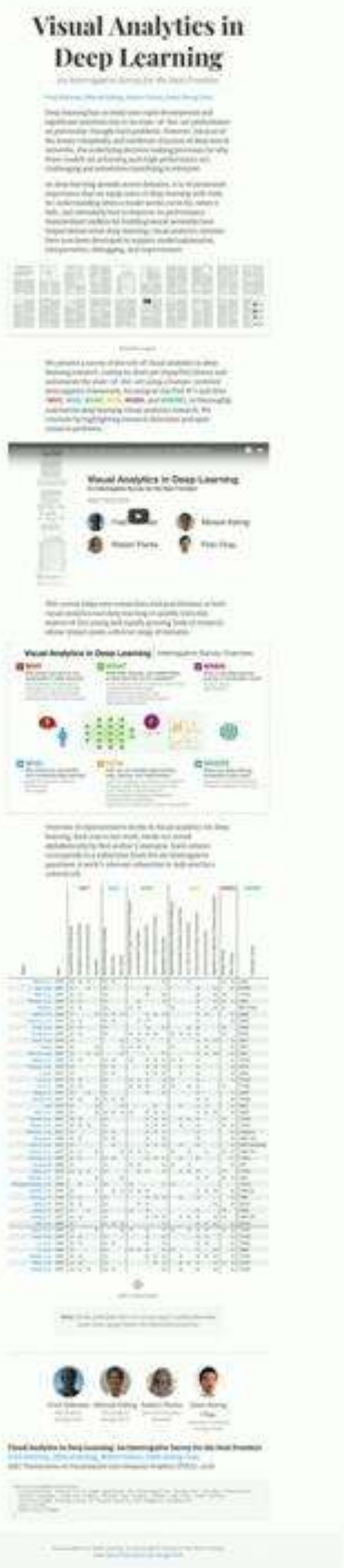
Where has deep learning visualization been used?

Application Domains & Models
A Vibrant Research Community

Key Takeaways

bit.ly/**va-dl-survey**

1. Most tools aimed at expert users
2. Instance-based analysis
3. Inherently interdisciplinary
4. Lacks actionability
5. Evaluation is hard
6. State-of-the-art models not robust



Gamut

A Design Probe to Understand How Data Scientists Understand Machine Learning Models

CHI 2019

Fred Hohman
@fredhohman
Georgia Tech

Andrew Head
UC Berkeley

Rich Caruana
Microsoft Research

Rob DeLine
Microsoft Research

Steven Drucker
Microsoft Research

What is interpretability?

Human understanding
of a system's…

internals
e.g., components
[Gilpin, 2018]

operations
e.g., math
[Biran, 2017]

data mapping
e.g., input to output
[Montavon, 2017]

representation
in an explanation
[Ribeiro, 2016]

What is interpretability?

Human understanding

of a system's…

internals
e.g., components
[Gilpin, 2018]

operations
e.g., math
[Biran, 2017]

data mapping
e.g., input to output
[Montavon, 2017]

representation
in an explanation
[Ribeiro, 2016]

No formal, agreed upon definition
[Lipton, 2016]

Gamut Contributions

1. **Capabilities**
   of interpretability

2. **Design Probe**
   embodying capabilities

3. **Evaluation & Investigation**
   of probe & emerging practice of interpretability w/ real users

From formative research

Explainable ML Interface Questions

Why does this house cost that much?

What is the difference between these two?

What if I added…

What are similar homes?

Where is it wrong?

What is most important?

From formative research

Explainable ML Interface Capabilities

Why does this house cost that much?

What is the difference between these two?

What if I added…

What are similar homes?

Where is it wrong?

What is most important?

From formative research

Explainable ML Interface Capabilities

C1 Why does this house cost that much?
Local instance explanations

C2 What is the difference between these two?
Instance explanation comparisons

C3 What if I added…
Counterfactuals

C4 What are similar homes?
Nearest neighbors

C5 Where is it wrong?
Regions of error

C6 What is most important?
Feature importance

From formative research

Explainable ML Interface Capabilities

C1 Why does this house cost that much?
Local instance explanations

C2 What is the difference between these two?
Instance explanation comparisons

C3 What if I added…
Counterfactuals

C4 What are similar homes?
Nearest neighbors

C5 Where is it wrong?
Regions of error

C6 What is most important?
Feature importance

Definitions + examples in the paper!

Takeaways

🔬 **Consider interpretability capabilities for your interfaces**
Interpretability is ***not a singular, rigid concept***

🌎 **Tailor explanations for specific audiences**
Balance ***simplicity*** and ***completeness***

⚡ **Design and integrate effective interaction**
Interaction key to ***realizing interpretability*** & solidify model understanding
[Weld & Bansal, 2018]

Gamut

A Design Probe to Understand How Data Scientists Understand Machine Learning Models

bit.ly/**gamut-chi**

paper video blog slides

Fred Hohman
@fredhohman
Georgia Tech

Andrew Head
UC Berkeley

Rich Caruana
Microsoft Research

Rob DeLine
Microsoft Research

Steven Drucker
Microsoft Research

Secure AI

Attack & Defense (DNN)

ShapeShifter

SHIELD

Do-it-yourself Adversarial ML

ADAGIO

MLsploit

Interpretable AI

Understand Industry Models

ActiVis

ML Education

GAN Lab

Research landscape

Survey, Gamut

Thanks!

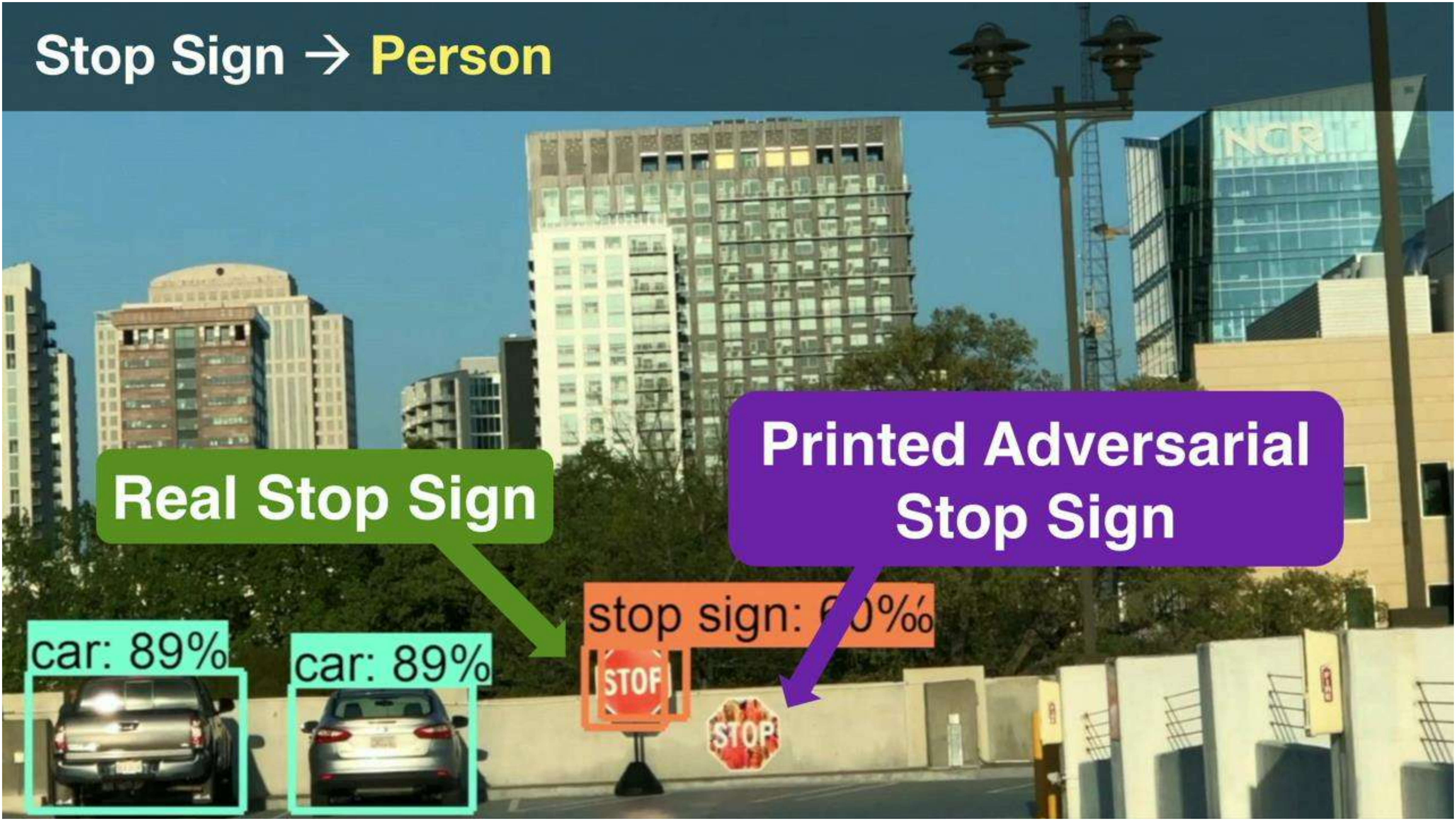
TOWARDS

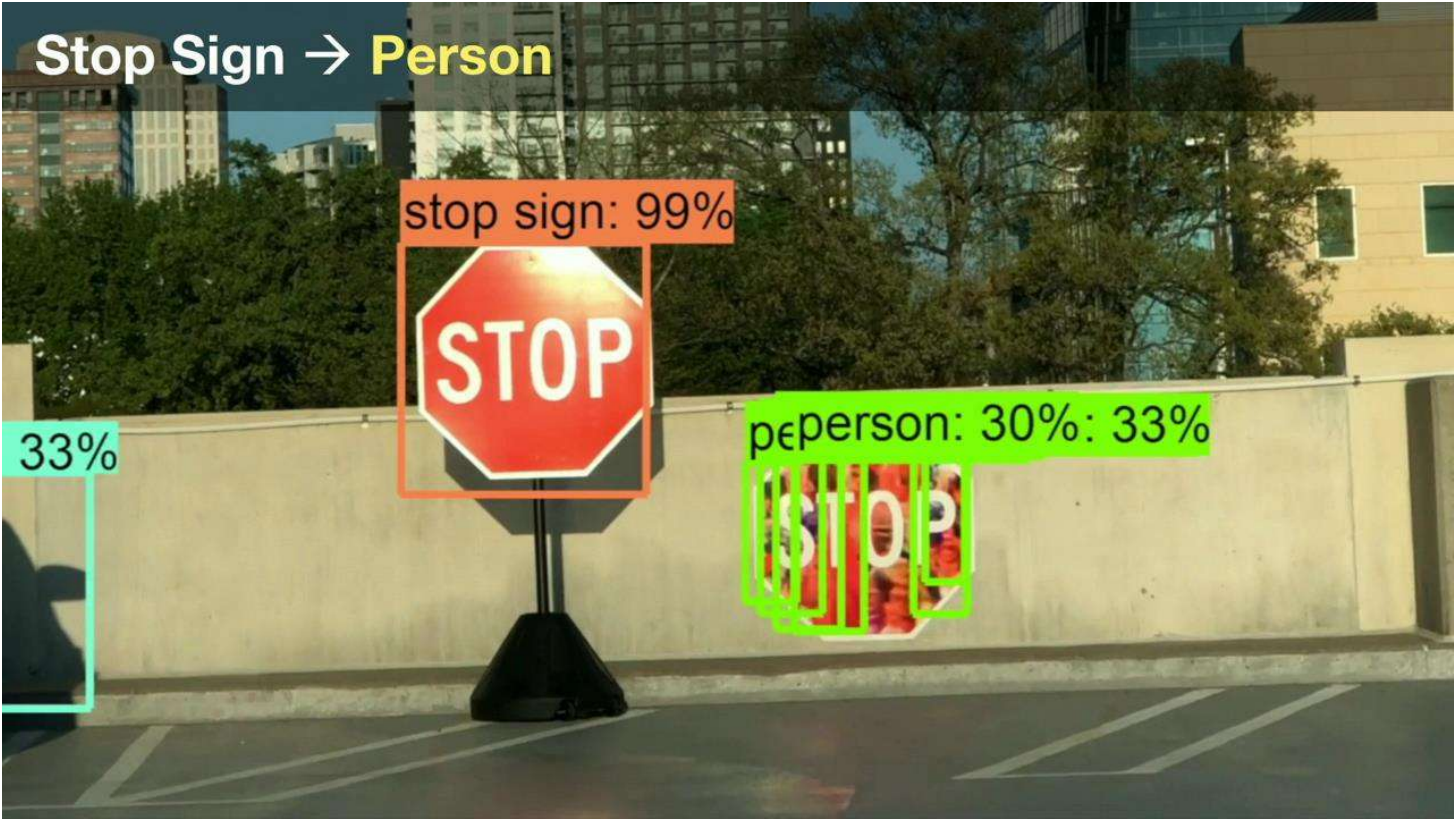
SECURE & INTERPRETABLE AI

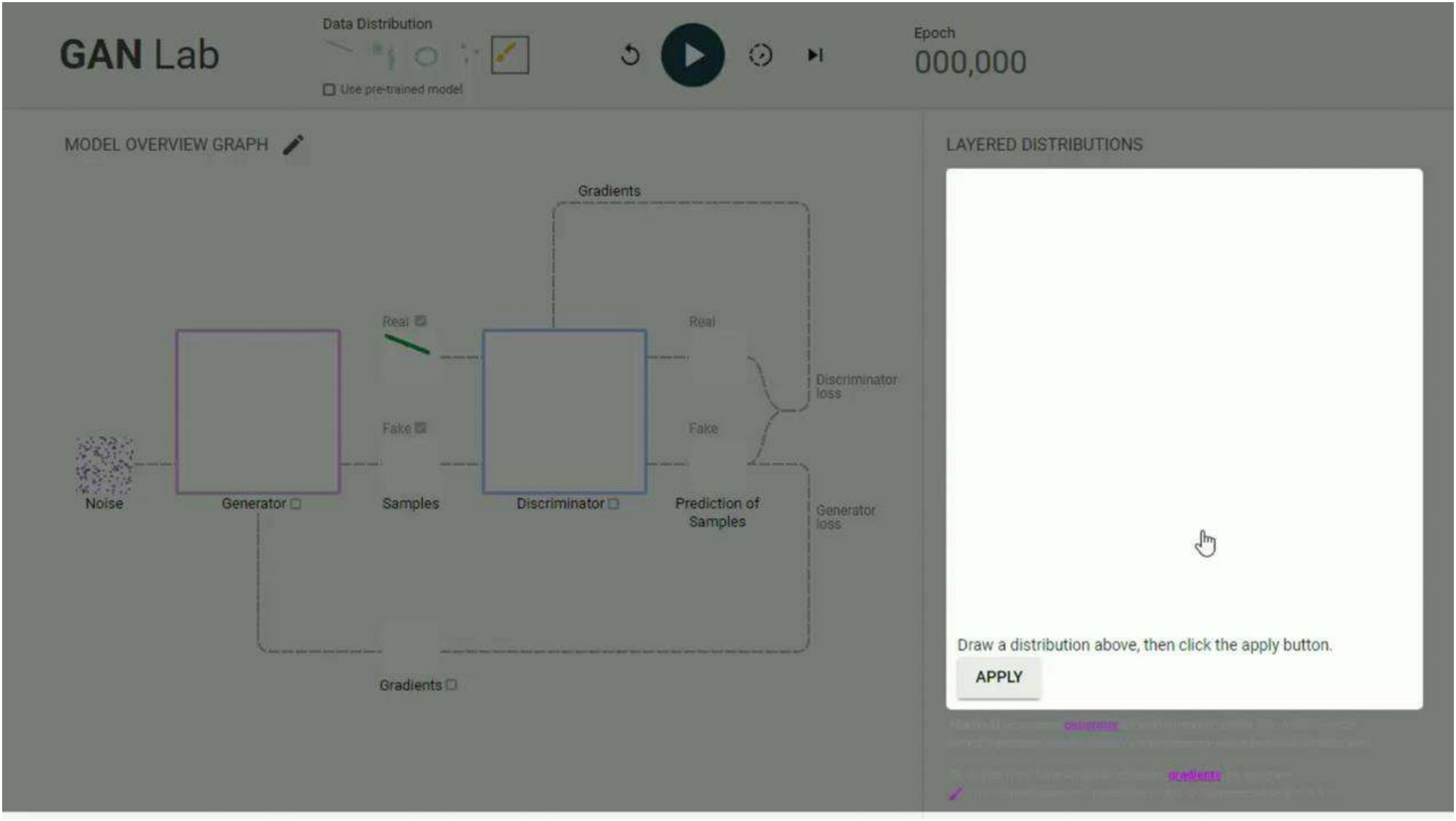
SCALABLE METHODS, INTERACTIVE VISUALIZATIONS, PRACTICAL TOOLS

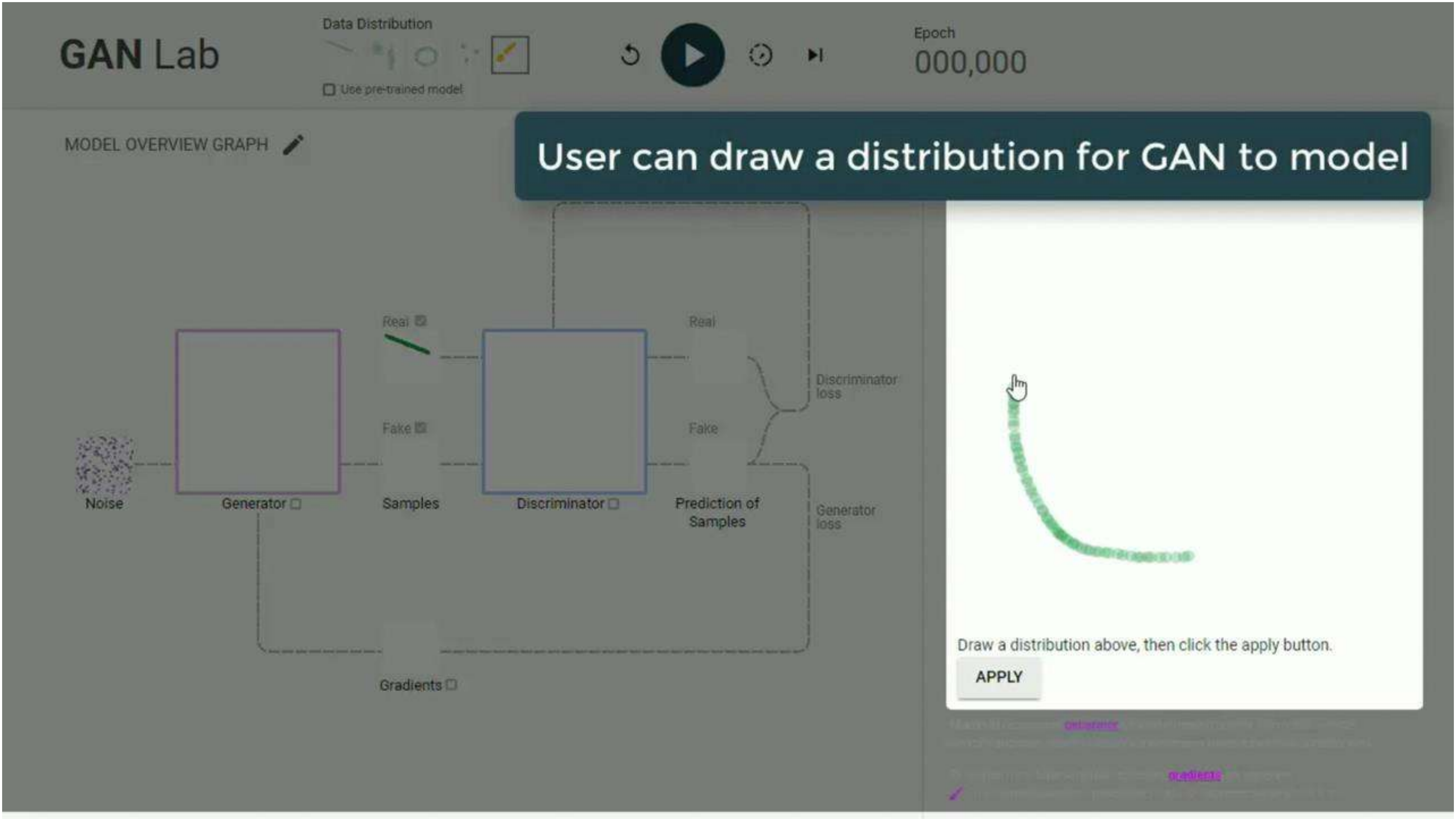
Polo Chau
Associate Professor
Associate Director, MS Analytics
Georgia Tech

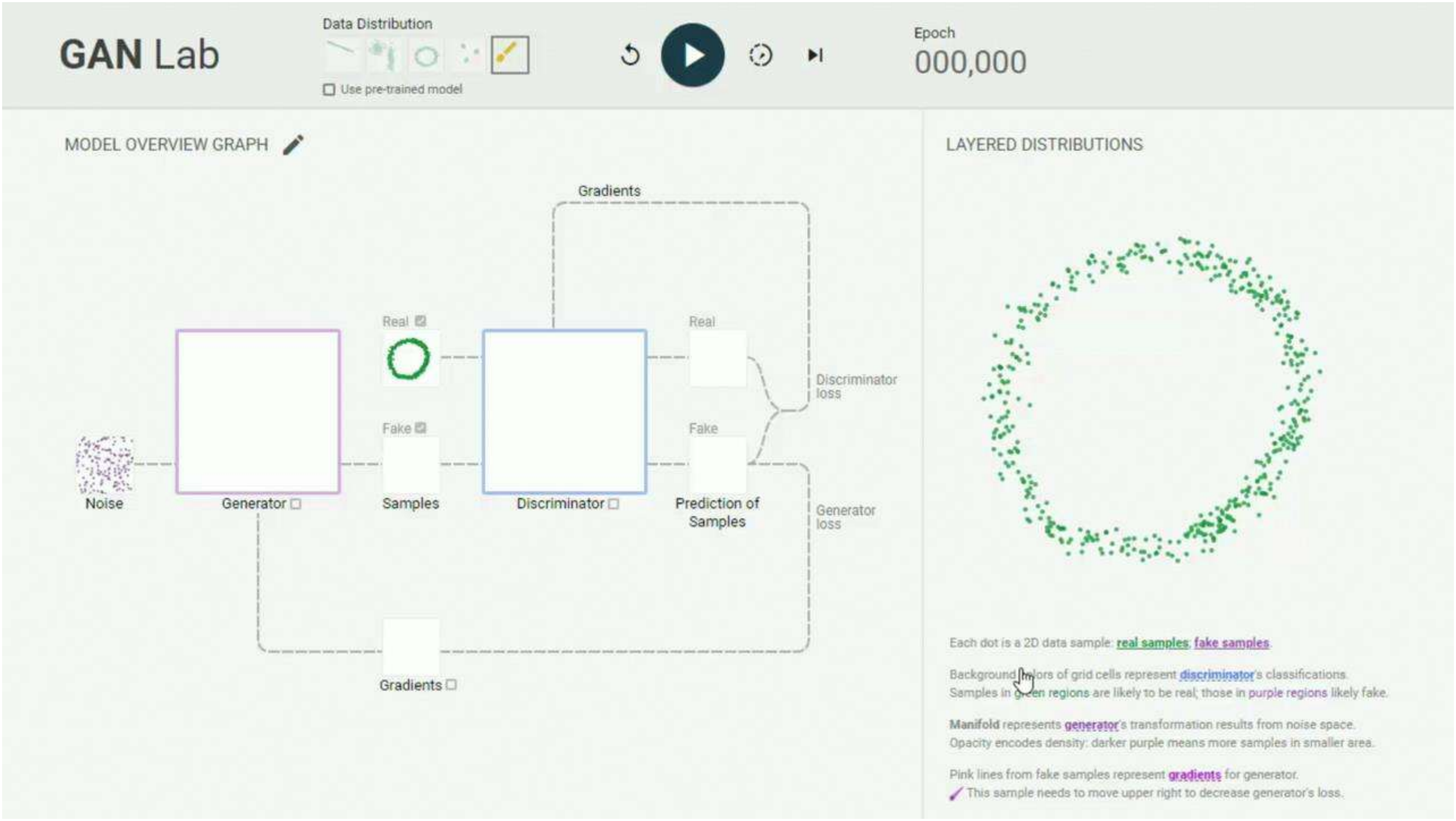
poloclub.github.io

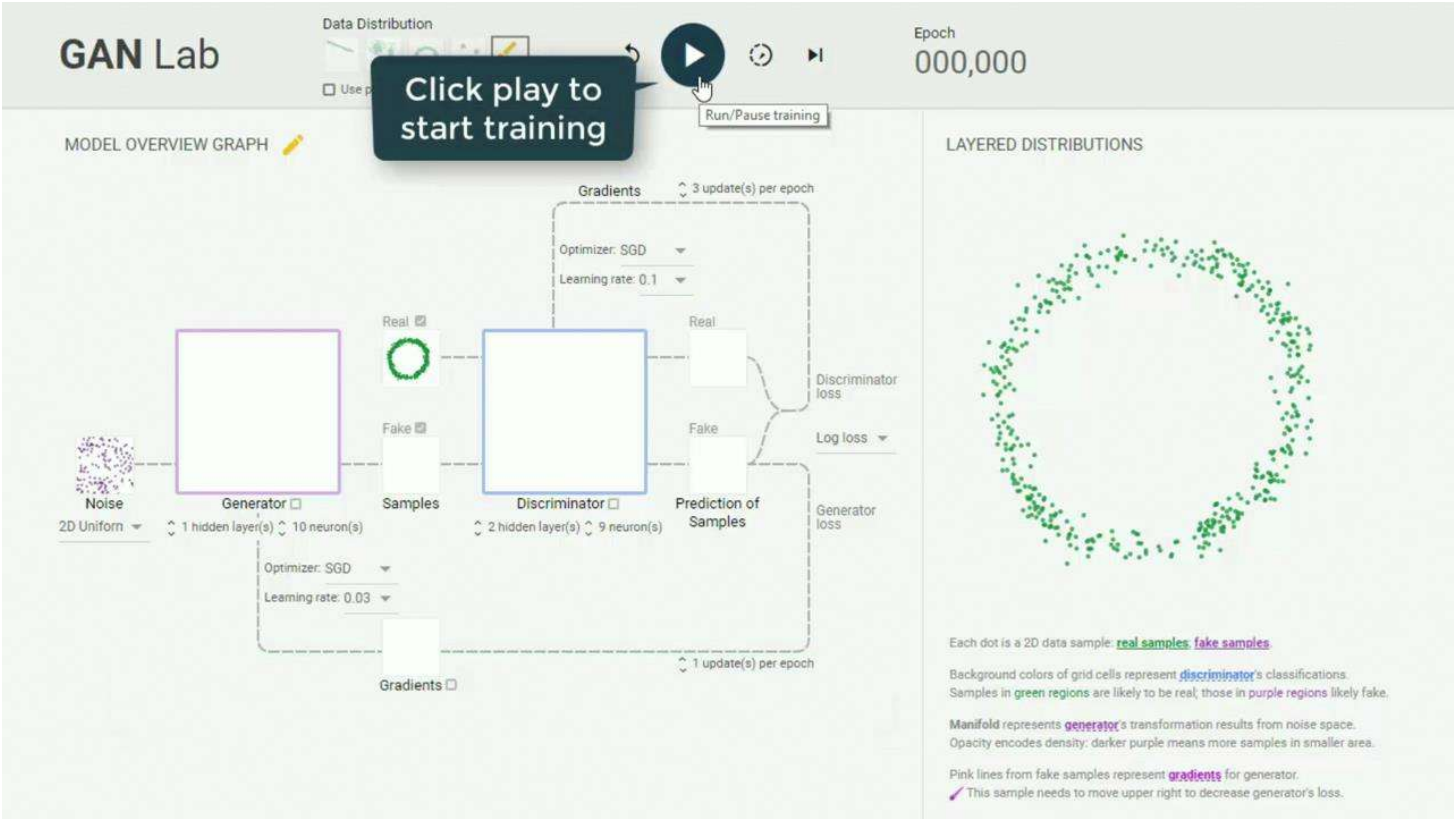

Stop Sign → Person
NCR
Printed Adversarial
Stop Sign
Real Stop Sign
stop sign: 60%
car: 89%
car: 89%
STOP
STOP


Stop Sign → Person
stop sign: 99%
STOP
33%
person: 30%: 33%
STOP

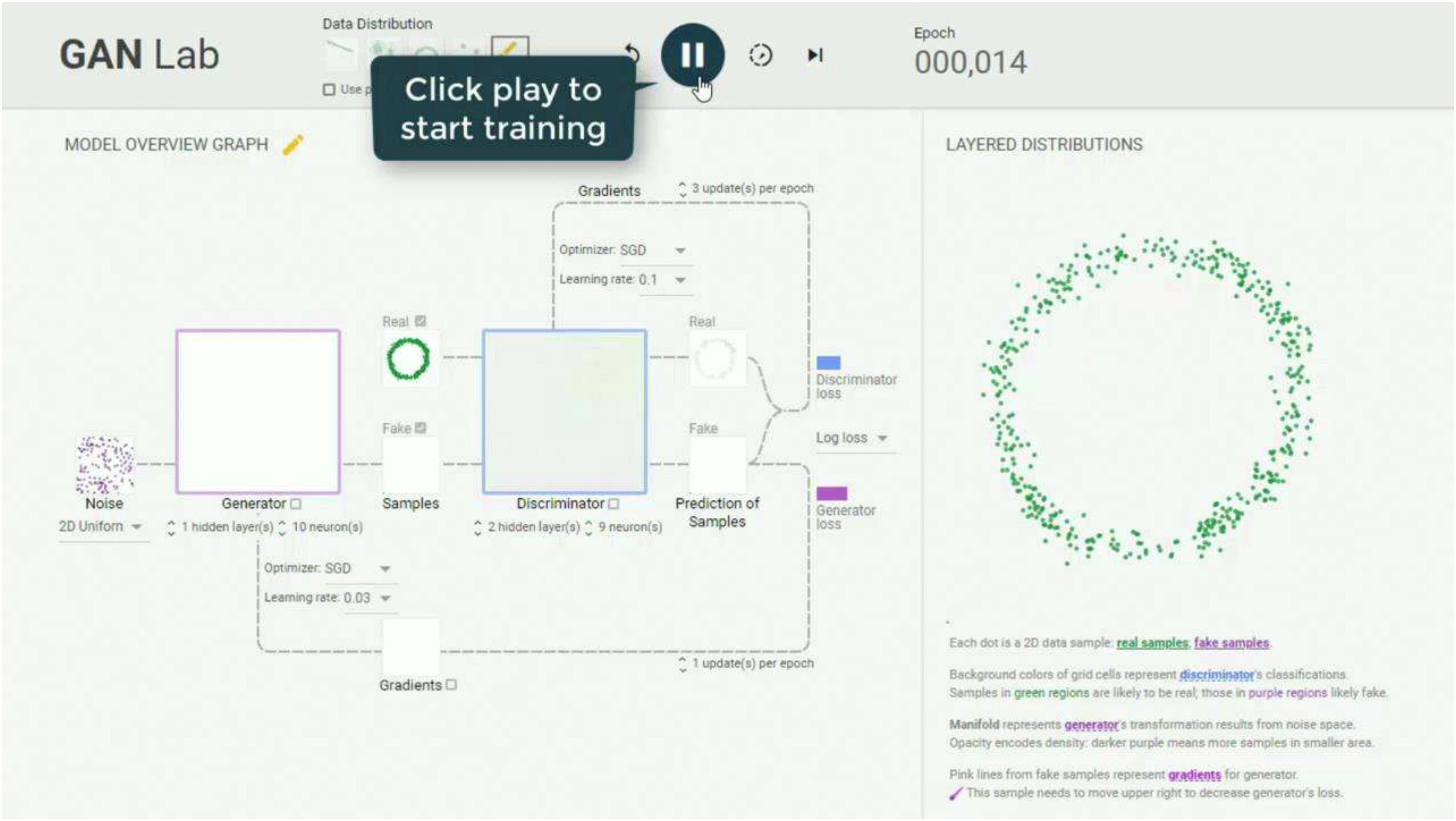

GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,000
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
Real
Fake
Real
Fake
Discriminator loss
Generator loss
Noise
Generator
Samples
Discriminator
Prediction of Samples
Gradients
Draw a distribution above, then click the apply button.
APPLY

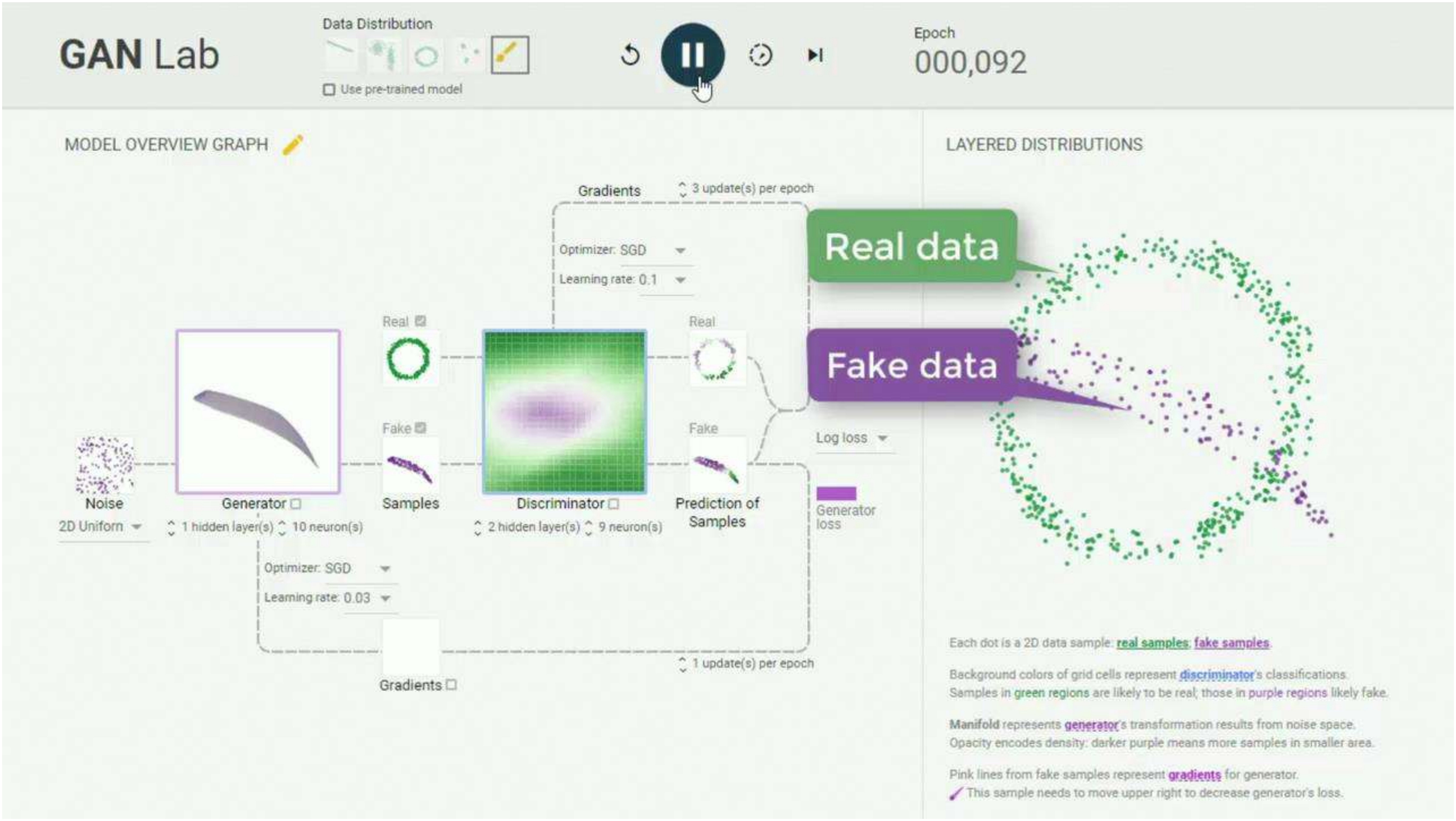

GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,000
MODEL OVERVIEW GRAPH
User can draw a distribution for GAN to model
Real
Fake
Noise
Generator
Samples
Discriminator
Prediction of Samples
Discriminator loss
Generator loss
Gradients
Draw a distribution above, then click the apply button.
APPLY

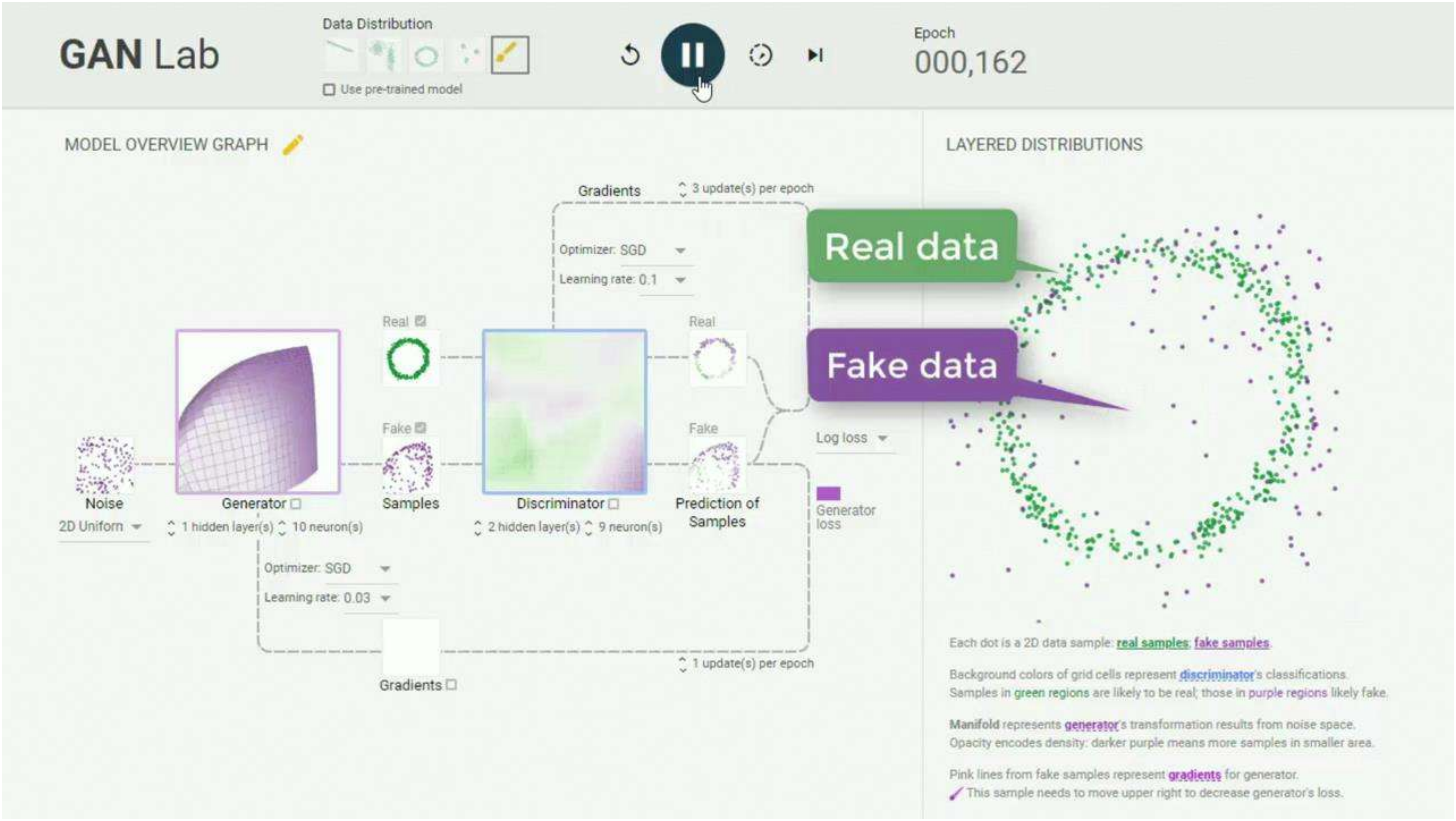

GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,000
MODEL OVERVIEW GRAPH
Gradients
Real
Fake
Noise
Generator
Samples
Discriminator
Prediction of Samples
Discriminator loss
Generator loss
Gradients
LAYERED DISTRIBUTIONS
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

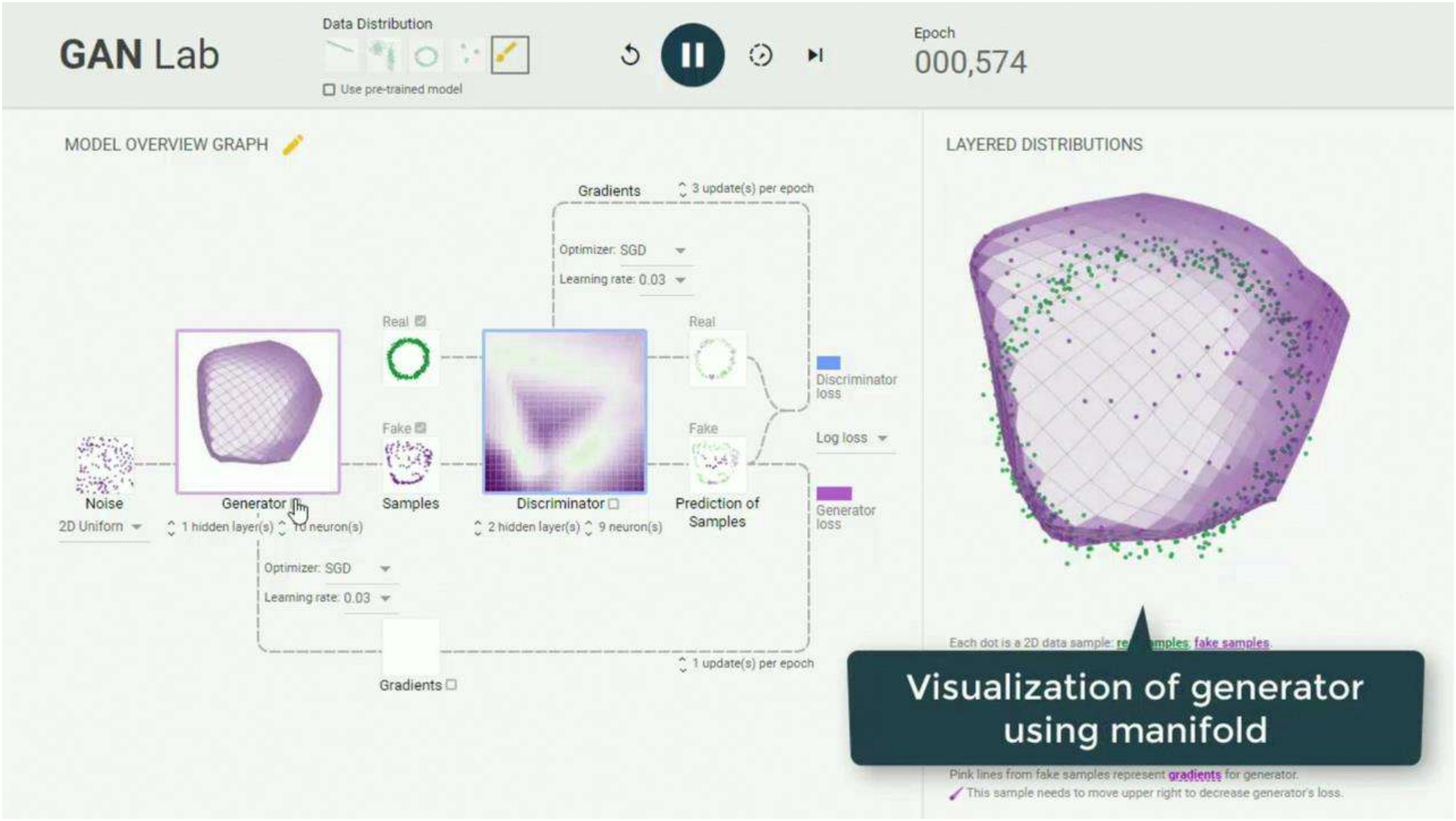


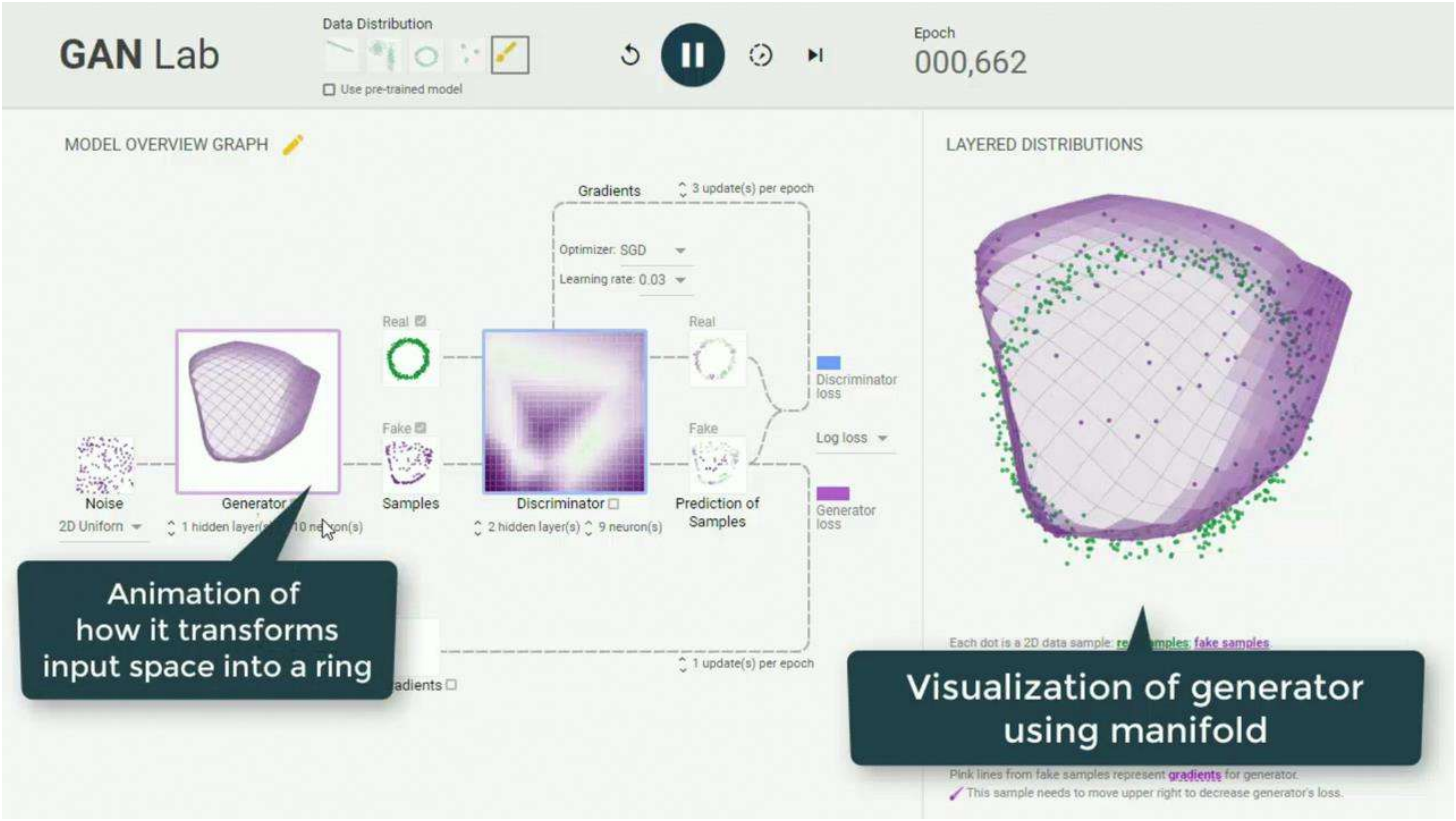
GAN Lab
Data Distribution
Use p
Click play to start training
Run/Pause training
Epoch
000,000
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.1
Real
Fake
Noise
2D Uniform
Generator
1 hidden layer(s)
10 neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Discriminator loss
Log loss
Generator loss
Optimizer: SGD
Learning rate: 0.03
1 update(s) per epoch
Gradients
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

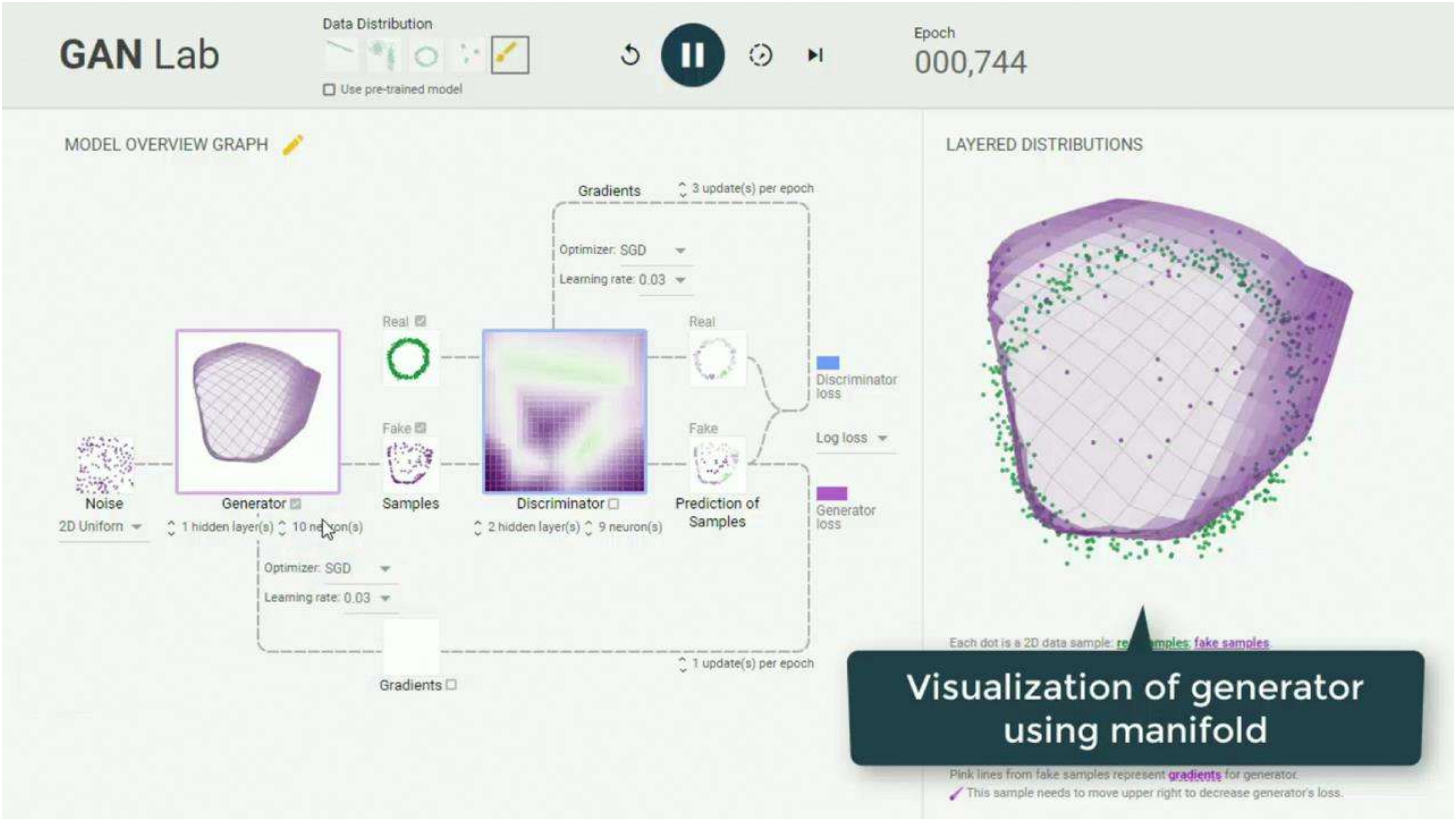

GAN Lab
Data Distribution
Epoch
000,014
Click play to start training
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.1
Real
Fake
Real
Fake
Discriminator loss
Log loss
Generator loss
Noise
2D Uniform
Generator
1 hidden layer(s)
10 neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Optimizer: SGD
Learning rate: 0.03
1 update(s) per epoch
Gradients
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

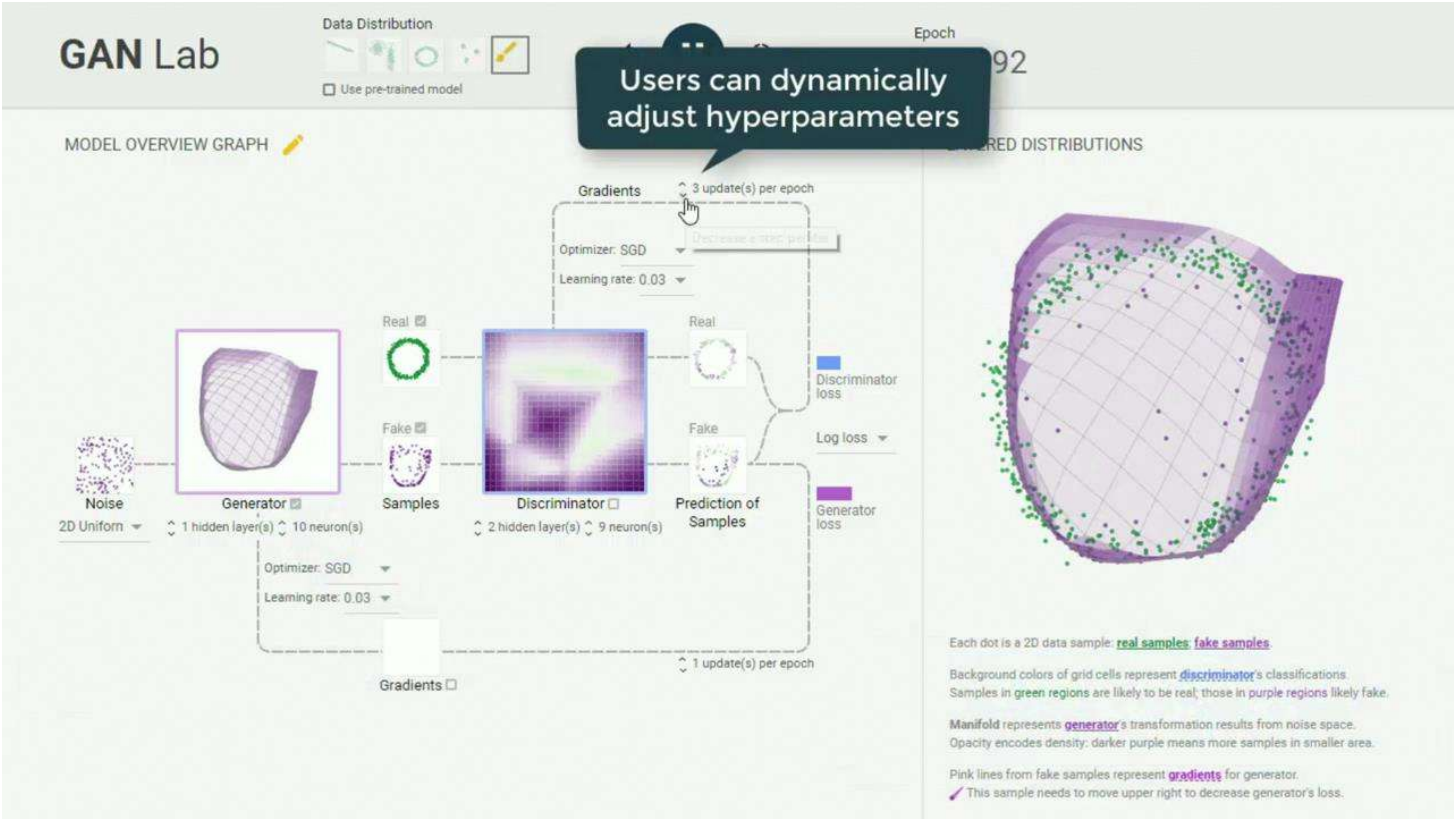

GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,092
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.1
Real data
Fake data
Real
Fake
Real
Fake
Noise
2D Uniform
Generator
1 hidden layer(s)
10 neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Log loss
Generator loss
Optimizer: SGD
Learning rate: 0.03
Gradients
1 update(s) per epoch
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.


GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,162
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.1
Real data
Fake data
Real
Fake
Noise
2D Uniform
Generator
1 hidden layer(s)
10 neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Log loss
Generator loss
Optimizer: SGD
Learning rate: 0.03
1 update(s) per epoch
Gradients
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

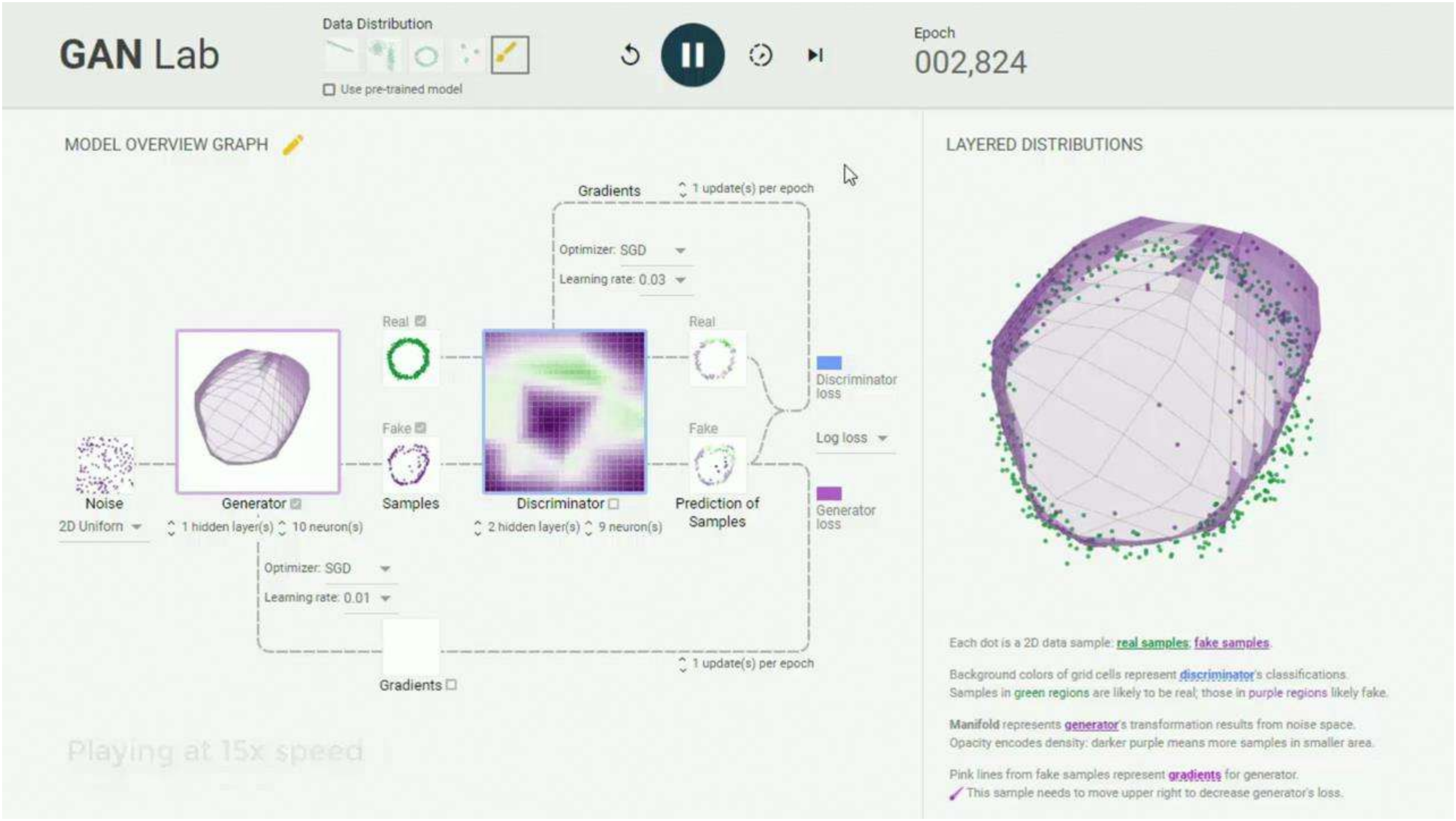

GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,574
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.03
Real
Fake
Noise
2D Uniform
Generator
1 hidden layer(s)
neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Discriminator loss
Log loss
Generator loss
Optimizer: SGD
Learning rate: 0.03
1 update(s) per epoch
Gradients
Each dot is a 2D data sample:
fake samples.
Visualization of generator using manifold
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.


GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,662
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.03
Real
Fake
Discriminator loss
Log loss
Generator loss
Noise
2D Uniform
Generator
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
1 update(s) per epoch
Animation of how it transforms input space into a ring
Visualization of generator using manifold
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.


GAN Lab
Data Distribution
Use pre-trained model
Epoch
000,744
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.03
Real
Fake
Discriminator loss
Log loss
Generator loss
Noise
2D Uniform
Generator
1 hidden layer(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Optimizer: SGD
Learning rate: 0.03
1 update(s) per epoch
Gradients
fake samples
Visualization of generator using manifold
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

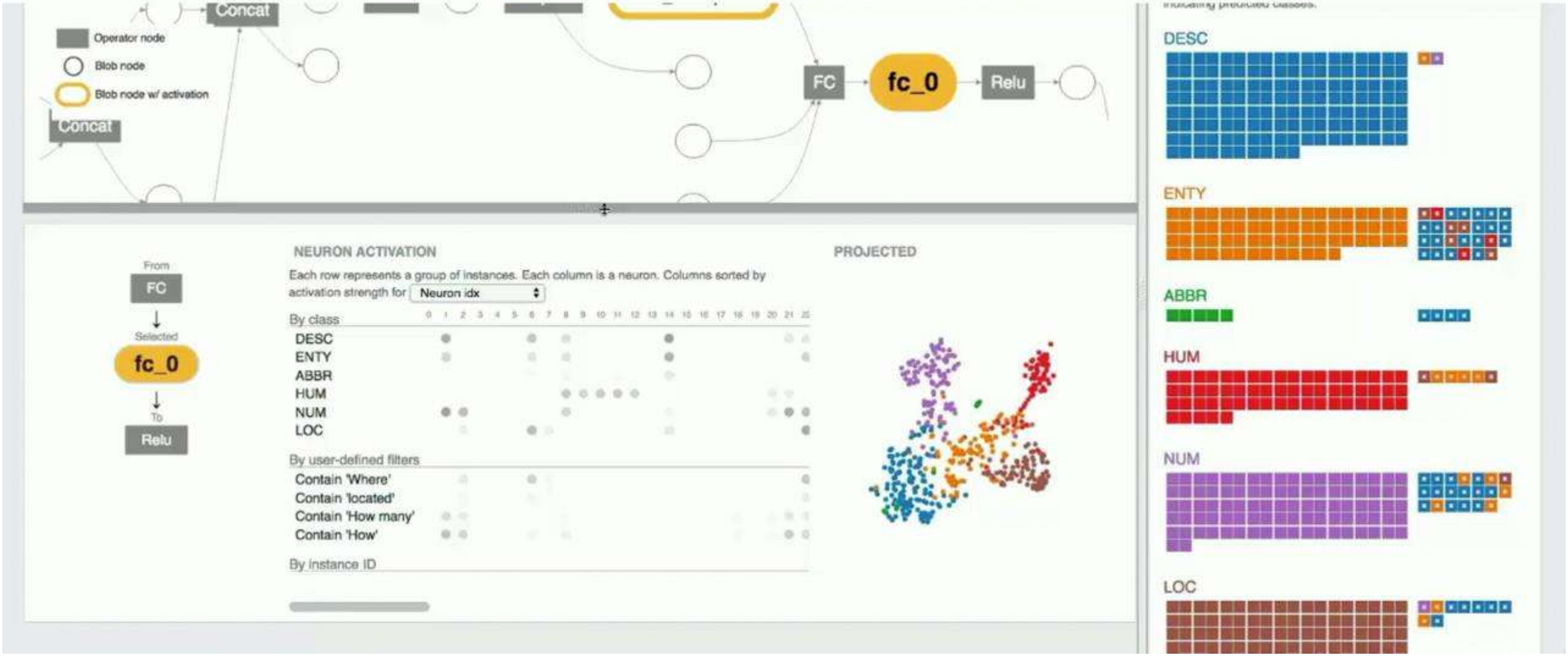


GAN Lab
Data Distribution
Use pre-trained model
Epoch
Users can dynamically adjust hyperparameters
MODEL OVERVIEW GRAPH
Gradients
3 update(s) per epoch
Optimizer: SGD
Learning rate: 0.03
Real
Fake
Real
Fake
Discriminator loss
Log loss
Generator loss
Noise
2D Uniform
Generator
1 hidden layer(s)
10 neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Optimizer: SGD
Learning rate: 0.03
1 update(s) per epoch
Gradients
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.


GAN Lab
Data Distribution
Use pre-trained model
Epoch
002,824
MODEL OVERVIEW GRAPH
LAYERED DISTRIBUTIONS
Gradients
1 update(s) per epoch
Optimizer: SGD
Learning rate: 0.03
Real
Fake
Discriminator loss
Log loss
Generator loss
Noise
2D Uniform
Generator
1 hidden layer(s)
10 neuron(s)
Samples
Discriminator
2 hidden layer(s)
9 neuron(s)
Prediction of Samples
Optimizer: SGD
Learning rate: 0.01
1 update(s) per epoch
Gradients
Playing at 15x speed
Each dot is a 2D data sample: real samples; fake samples.
Background colors of grid cells represent discriminator's classifications.
Samples in green regions are likely to be real; those in purple regions likely fake.
Manifold represents generator's transformation results from noise space.
Opacity encodes density: darker purple means more samples in smaller area.
Pink lines from fake samples represent gradients for generator.
This sample needs to move upper right to decrease generator's loss.

ActiVis Key Ideas (2)

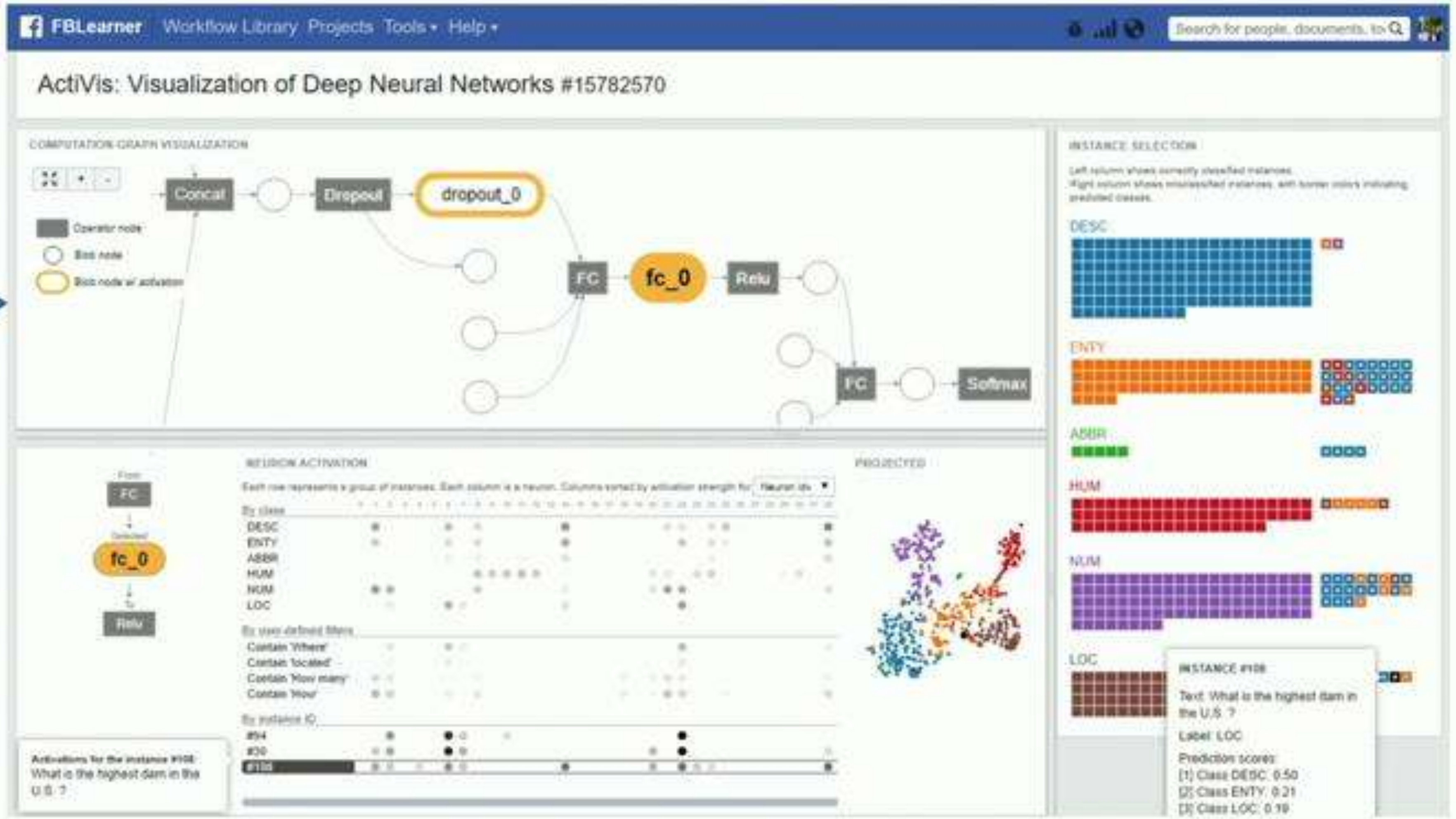
Unified Analysis for Instances & Subsets



Deployed on *FBLearner*

Facebook's ML platform

used by >25% of engineering team



Secure AI

Attack & Defense (DNN)

ShapeShifter

SHIELD

Do-it-yourself Adversarial ML

ADAGIO

MLsploit

Interpretable AI

Understand Industry Models

ActiVis

ML Education

GAN Lab

Research landscape

Survey, Gamut