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

AI + HI

Artificial Intelligence + Human Intelligence

Scalable interactive tools to make sense of complex large-scale datasets and models
Today's Main Topics

Secure AI

Interprettable 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.

Secure AI

The self-driving Uber was traveling north at about 40 m.p.h.

New York Times, 2018

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.
<table>
<thead>
<tr>
<th>Secure AI</th>
<th>Interpretable AI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack &amp; Defense</strong> (DNN)</td>
<td>Understand Industry Models</td>
</tr>
<tr>
<td>ShapeShifter</td>
<td>ActiVis</td>
</tr>
<tr>
<td>SHIELD</td>
<td><strong>Interactive Learning</strong> (Education)</td>
</tr>
<tr>
<td><strong>Do-it-yourself Adversarial ML</strong></td>
<td>GAN Lab</td>
</tr>
<tr>
<td>ADAGIO</td>
<td><strong>Research landscape</strong></td>
</tr>
<tr>
<td>MLsploit</td>
<td>Survey, Gamut</td>
</tr>
</tbody>
</table>
"The toaster has been hacked into thinking it's a blender."
AI Security Problems Are Everywhere

"The toaster has been hacked into thinking it's a blender."

Smart toaster does exist!
AI Security is becoming increasingly important

50 Billion Smart Objects

Source: Cisco
AI Security is becoming increasingly important

- Inflection point:
  - 2010: 6.8 Billion Smart Objects
  - 2015: 7.2 Billion
  - 2020: 7.6 Billion

- World Population:
  - 2010: 7.2 Billion
  - 2015: 7.6 Billion

- # Incidents reported by U.S. federal agencies:
  - 2006: 5,503
  - 2009: 29,999
  - 2012: 48,562
  - 2015: 77,183

Source: Cisco
Source: US Department of Homeland Security
AI Security is becoming increasingly important

- **Inflection point**
  - 2010: 6.8
  - 2015: 7.2
  - 2020: 7.6

**World Population**

- **50 Billion Smart Objects**

**# incidents reported by U.S. federal agencies**

- **Increased > 10-fold**
  - 2006: 5,503
  - 2009: 29,999
  - 2012: 48,562
  - 2015: 77,183

Source: Cisco

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
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

output a single label, e.g., “car”
Object Detection
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
Autonomous car system

Capture → Preprocess → Recognition

Attacker has no access to internal pipeline

Digital Attack
Physically Realizable Adversarial Attack

Capture → Preprocess → Recognition

Autonomous car system

Attacker has no access to internal pipeline

Digital Attack

Manipulate Physical Environment
Physically Realizable Adversarial Attack

Manipulate Physical Environment = More Realistic, Targeted Attack

Attacker has no access to internal pipeline

Digital Attack
Stop Sign → Person

stop sign: 99%

person: 30%, 33%
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
Lu et al. [1] show the current technique cannot fool state-of-the-art object detectors like Faster R-CNN and YOLO.
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

feature map sharing
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: \textit{sum of classification losses}
Solution 1: Fool Multiple Region Proposals

Minimize: \( \text{sum of classification losses} + \text{deviation loss} \)
Solution 1: Fool Multiple Region Proposals

Minimize: \( \text{sum of classification losses} + \text{deviation loss} \)

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

STOP  STOP  STOP

Perturbations more conspicuous in order to survive viewing distances, angles, lighting conditions, camera limitations.
Untargeted Attack
ShapeShifter Motivates New DARPA Program
GARD: Defense for AI

DARPA
State of the art: few physical attacks

Graffiti:
STOP = SPEED LIMIT 45
(Cviklov et al., UC Berkeley, 2017)

Patch:

Banana =
(Brown et al., Google, 2017)

3D Printed Objects:

(Atlase et al., MIT, 2017)

Fooling Deep Neural Networks with Physical Attacks
Security and Privacy Research, Intel Labs
Shang-tse Chen | Cory Cornelius | Jason Martin

- All physical attacks to date are White Box
- No current consideration of resource constraints


Highlights ShapeShifter as the state-of-the-art physical attack
Adversarial Machine Learning Landscape

Attack

Defense
Adversarial Machine Learning Landscape

Attack

Defense

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

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

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 a 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)

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

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]

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

Georgia Tech Intel
Adversarial Attack on Speech-to-text

OPEN INTEL DOT COM

MODEL

OPEN EVIL DOT COM
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 = Attack & Defense for Audio in a Gadget with Interactive Operations
isn't the party also to announce his engagement to joanna
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
A Framework for Interactive Experimentation with Adversarial Machine Learning Research


[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
EASY INTEGRATION OF RESEARCH
**MLsploit**

### Attack Pipeline

- **Run**
- **Edit**
- **Duplicate**
- **Delete**
- **View Sample Files**

- **FINISHED**
- attack-resnet50_v2-fgsm
  - epsilon: 4

- **FINISHED**
- evaluate-resnet50_v2

### 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

### Attack-Defend Pipeline (SLQ)

- **Run**
- **Edit**
- **Duplicate**
- **Delete**
- **View Sample Files**

- **FINISHED**
- attack-resnet50_v2-fgsm
  - epsilon: 4

2019-01-09 18:07:33.090666: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: __mm512_movehl_epi32

2019-01-09 18:07:33.090682: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: __mm512_movehl_epi32

2019-01-09 18:07:33.090698: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: __mm512_movehl_epi32

---

testattack.zip
Accuracy: 0.6800
Intel® AI Courses

Learn AI theory and follow hands-on exercises with our free courses for software developers, data scientists, and students. These lessons cover AI topics and explore tools and optimized libraries that take advantage of Intel® processors in personal computers and server workstations.

MLsplot

github.com/mlsplot

software.intel.com/en-us/ai/courses
<table>
<thead>
<tr>
<th>Secure AI</th>
<th>Interpretable AI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack &amp; Defense (DNN)</strong></td>
<td><strong>Understand Industry Models</strong></td>
</tr>
<tr>
<td>ShapeShifter</td>
<td>ActiVis</td>
</tr>
<tr>
<td>SHIELD</td>
<td></td>
</tr>
<tr>
<td><strong>Do-it-yourself Adversarial ML</strong></td>
<td><strong>Interactive Learning (Education)</strong></td>
</tr>
<tr>
<td>ADAGIO</td>
<td>GAN Lab</td>
</tr>
<tr>
<td>MLsploit</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Research landscape</strong></td>
</tr>
<tr>
<td></td>
<td>Survey, Gamut</td>
</tr>
</tbody>
</table>
Practitioners very interested in **why** and **how** AI works
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.
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, ...

Enjoying nice weather with kiki

tags: #mycat, #cute
date: 10/1/2017
location: 33.7, 88.4
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
Challenge #1

How to visualize many model parameters?

INPUT

Where is Mercedes-Benz Stadium located?

MODEL

OUTPUT

Number: 11%
Person: 8%
Location: 81%
Challenge #1

How to visualize many model parameters?

INPUT

Where is Mercedes-Benz Stadium located?

MODEL

many layers

OUTPUT

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%

many layers
particularly useful
ActiVis Key Ideas #1
Model Overview to Activation Details

ActiVis: Visualization of Deep Neural Networks #15782570

COMPUTATION GRAPH VISUALIZATION

Click a node on the top to see its details

INSTANCE SELECTION
Left column shows correctly classified instances. Right column shows misclassified instances, with border colors indicating predicted classes.

DESC

ENTRY

ABBR

HUM
ActiVis Key Ideas #1
Model Overview to Activation Details

Model Architecture Overview

ActiVis: Visualization of Deep Neural Networks #15782570

Click a node on the top to see its details
ActiVis Key Ideas #1

Model Overview to Activation Details

Model Architecture Overview

Zoom and pan to explore the graph
Challenge #2

How to analyze many data instances?

Observation: Two Analytics Patterns

<table>
<thead>
<tr>
<th>INSTANCE-LEVEL</th>
<th>SUBSET-LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>How model responds to individual instances?</td>
<td>How model behaves at higher-level categorization (e.g., by topic)?</td>
</tr>
<tr>
<td>Useful for debugging</td>
<td>Useful for large datasets</td>
</tr>
</tbody>
</table>
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets

Neuron Activation

- Each row represents a group of instances. Each column is a neuron. Columns sorted by activation strength for neuron idx 2.
- By class:
  - DESC
  - ENTY
  - ABBR
  - HUM
  - NUM
  - LOC
- By user-defined filters:
  - Contain 'Where'
  - Contain 'located'
  - Contain 'how many'
  - Contain 'How'
- By instance ID
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets

Columns: Neurons

Neuron Matrix view

Rows: Instance subsets
Unified Analysis for Instances & Subsets

- **Columns**: Neurons
- **Rows**: Instance subsets
- **Neuron Matrix view**: A set of instances whose true class is DESCRIPTION
**ActiVis Key Ideas (2)**

**Unified Analysis for Instances & Subsets**

- **Columns:** Neurons
- **Rows:** Instance subsets
- **Neuron Matrix view:** A set of instances whose true class is **DESC**ription
- **Neuron's average activation strength**

Diagram details:
- Average activations for instances where the class is **DESC**
- Neuron: Columns sorted by activation strength for neuron **fc_0**
- Rows: Instance subsets
- By class: DESC, ENTITY, ABBR, HUM, NUM, LOC
ActiVis Key Ideas (2)

Unified Analysis for Instances & Subsets

**Neuron Matrix view**

- **Columns:** Neurons
- **Rows:** Instance subsets
- Another instance subset
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 **FB Learner**

Facebook’s ML platform used by >25% of engineering team
Scalably summarize and interactively visualize neural network feature representations for millions of images.
Scalably summarize and interactively visualize neural network feature representations for millions of images.
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
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

- Neural Networks, Manifolds, and Topology
- Deep Learning, NLP, and Representations
- Calculus on Computational Graphs Backpropagation
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

\[
\min_D \max_G V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))].
\]

**Discriminator**

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

**Generator**

\[
\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D(G(z^{(i)}))\right).
\]
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
- **Discriminator** spots fake

**Counterfeiter** makes 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
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
What type of data to visualize?

Generator (Counterfeiter) -> Discriminator (Police)
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)
How to visually explain the *generator*?
How to visually explain the generator?
How to visually explain the generator?

Generator (Counterfeiter)

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?

Generator (Counterfeiter)

random

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

![2D heatmap diagram](image)

*Data points in this region are likely real.*

*Data points are likely fake.*
VER. 0.5

real data + fake data

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

Each dot is a 2D data sample: **real samples** vs. **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

real data + fake data + generator + discriminator

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.
Each dot is a 2D data sample: **real samples**: Take 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.
Hard to develop mental models for GANs
Hard to develop mental models for GANs
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 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.
Draw a distribution above, then click the apply button.
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 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
Interpretability 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
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
Visual Analytics in Deep Learning

**WHY**
Why would one want to use visualization in deep learning?
- Interpretability & Explainability
- Debugging & Improving Models
- Comparing & Selecting Models
- Teaching Deep Learning Concepts

**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

**WHEN**
When in the deep learning process is visualization used?
- During Training
- After Training

**WHO**
Who would use and benefit from visualizing deep learning?
- Model Developers & Builders
- Model Users
- Non-experts

**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

**WHERE**
Where has deep learning visualization been used?
- Application Domains & Models
- A Vibrant Research Community
Key Takeaways

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

bit.ly/va-dl-survey
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** Local instance explanations
- **C2** Instance explanation comparisons
- **C3** Counterfactuals
- **C4** Nearest neighbors
- **C5** Regions of error
- **C6** Feature importance

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

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
<table>
<thead>
<tr>
<th>Secure AI</th>
<th>Interpretable AI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack &amp; Defense (DNN)</strong></td>
<td><strong>Understand Industry Models</strong></td>
</tr>
<tr>
<td>ShapeShifter</td>
<td>ActiVis</td>
</tr>
<tr>
<td>SHIELD</td>
<td></td>
</tr>
<tr>
<td><strong>Do-it-yourself Adversarial ML</strong></td>
<td><strong>ML Education</strong></td>
</tr>
<tr>
<td>ADAGIO</td>
<td>GAN Lab</td>
</tr>
<tr>
<td>MLsploit</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Research landscape</strong></td>
</tr>
<tr>
<td></td>
<td>Survey, Gamut</td>
</tr>
</tbody>
</table>
TOWARDS SECURE & INTERPRETABLE AI
Scalable Methods, Interactive Visualizations, Practical Tools

Polo Chau
Associate Professor
Associate Director, MS Analytics
Georgia Tech
poloclub.github.io
Draw a distribution above, then click the apply button.
User can draw a distribution for GAN to model.

Draw a distribution above, then click the apply button.
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.
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.
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.
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.
Animation of how it transforms input space into a ring

Visualization of generator using manifold
Users can dynamically adjust 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.
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 **FB Learner**
Facebook's ML platform
used by >25% of engineering team
<table>
<thead>
<tr>
<th>Secure AI</th>
<th>Interpretable AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack &amp; Defense (DNN)</td>
<td>Understand Industry Models</td>
</tr>
<tr>
<td>ShapeShifter</td>
<td>ActiVis</td>
</tr>
<tr>
<td>SHIELD</td>
<td>ML Education</td>
</tr>
<tr>
<td>Do-it-yourself Adversarial ML</td>
<td>GAN Lab</td>
</tr>
<tr>
<td>ADAGIO</td>
<td>Research landscape</td>
</tr>
<tr>
<td>MLsploit</td>
<td>Survey, Gamut</td>
</tr>
</tbody>
</table>