上海科技大学
ShanghaiTech University

ARTS1422 Data Visualization

Lecture 13

VIS4AI:
Visualization for Explainable Classifiers

Quan Li

Spring 2024

2024.04.16

Outline

Motivation

A study from Cost-Effective HealthCare (CEHC) (Cooper et al. 1997 [1]) Predicting the **probability of death** (POD) for patients with pneumonia

If HighRisk(x):

admit to hospital

Else:

treat as outpatient

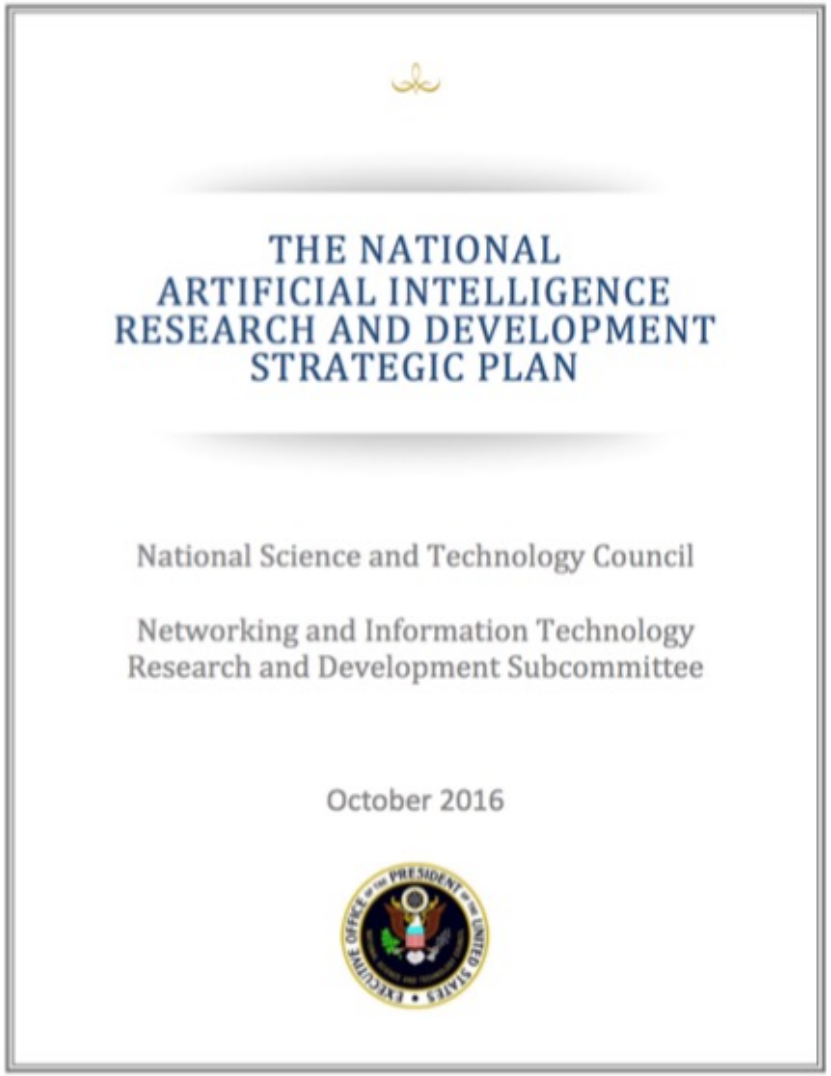
The rule-based model learned:

HasAsthma(x) => LowerRisk(x)

High risk --> aggressive treatment

We want the system to be explainable sometime!

Motivation


THE NATIONAL ARTIFICIAL INTELLIGENCE RESEARCH AND DEVELOPMENT STRATEGIC PLAN

National Science and Technology Council

Networking and Information Technology Research and Development Subcommittee

October 2016

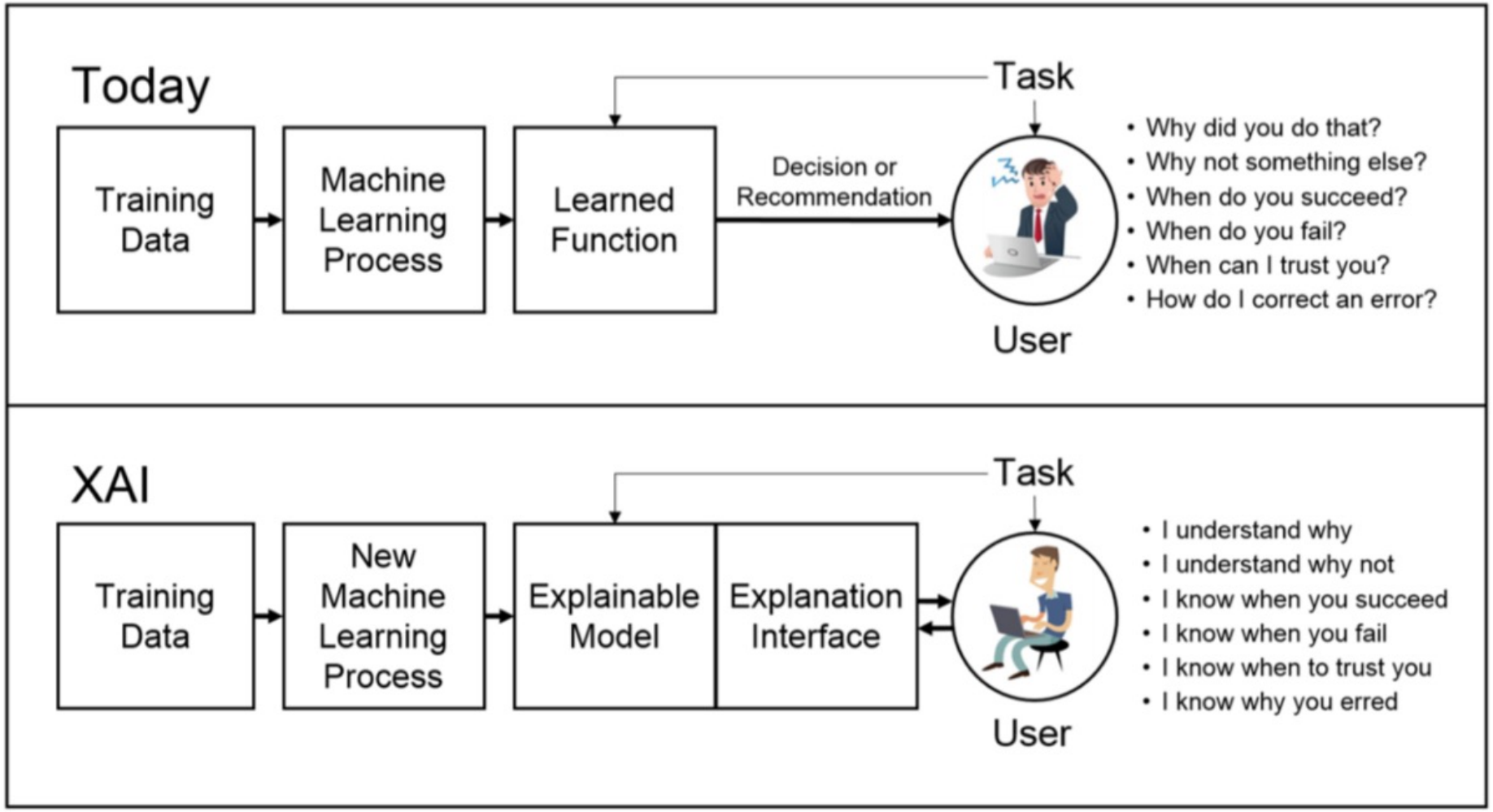
“

Strategy 2: Developing Effective Methods for AI-Human Collaboration

Better visualization and user interfaces are additional areas that need much greater development to **help humans understand large-volume modern datasets** and **information coming from a variety of sources**

”

Motivation



The concept of XAI. DARPA, Explainable AI Project 2017 [2]

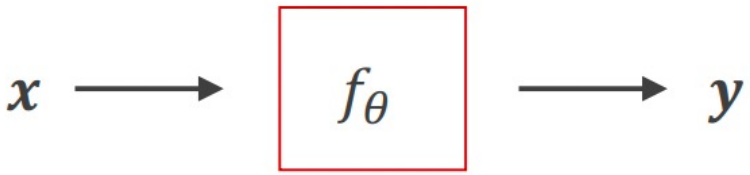
Classification

Classification: Identifying any observation $\boldsymbol{x} \in \mathcal{X}$ as a class $y \in \mathcal{Y}$, $\mathcal{Y} = \{1,2,\dots,K\}$, given a training set $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$

Classification Model (Classifier): An algorithm f, learned from $\mathcal{D}$, specified by parameters θ, output is a vector representing a probability distribution:

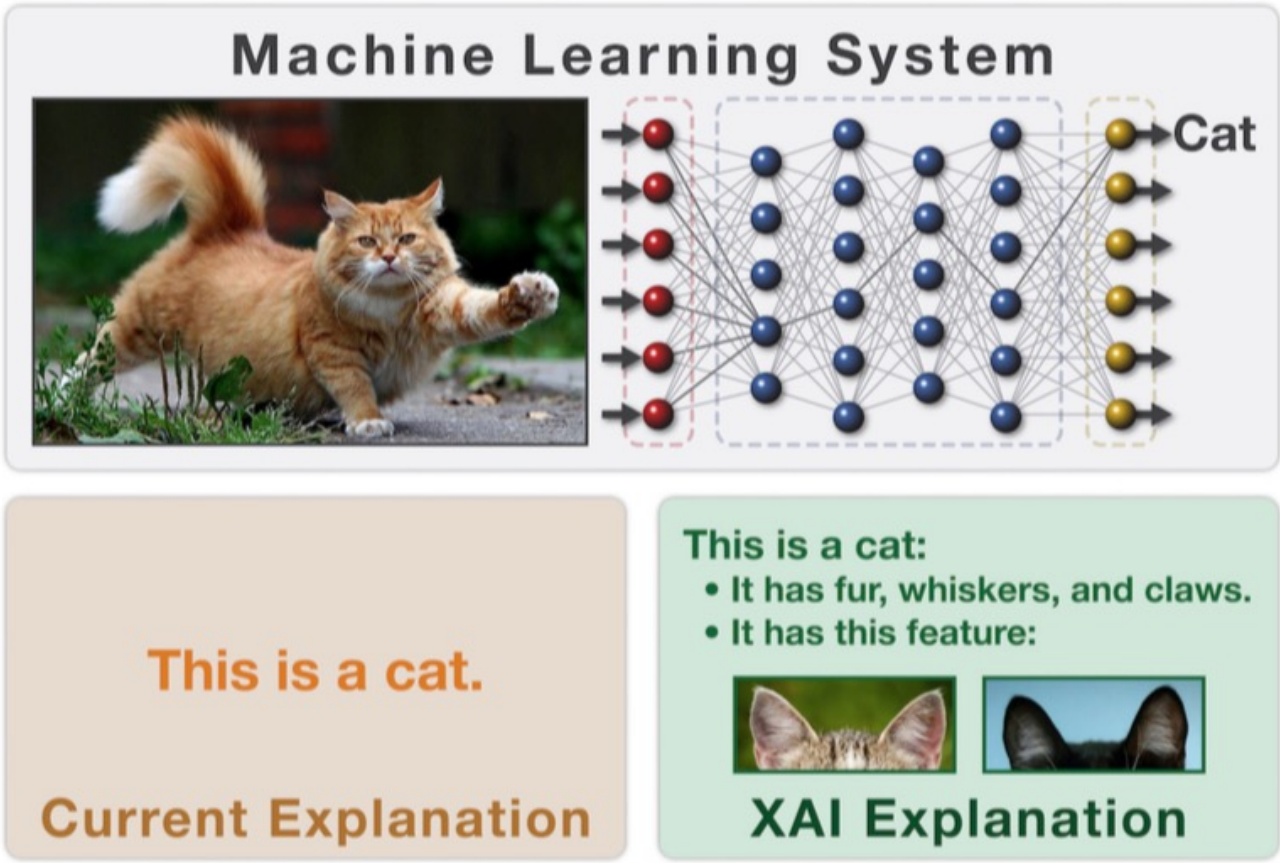
$$\boldsymbol{y} = f_\theta(\boldsymbol{x}),$$

where $\boldsymbol{y} = (y_i) \in \mathbb{R}^K$, $y_i = p(y = i \mid \boldsymbol{x}, \mathcal{D})$.



What is explainability?

The **interpretability** of a model: The ability to explain the reasoning of its predictions so that humans can understand. (Doshi-Velez and Kim 2017 [3])



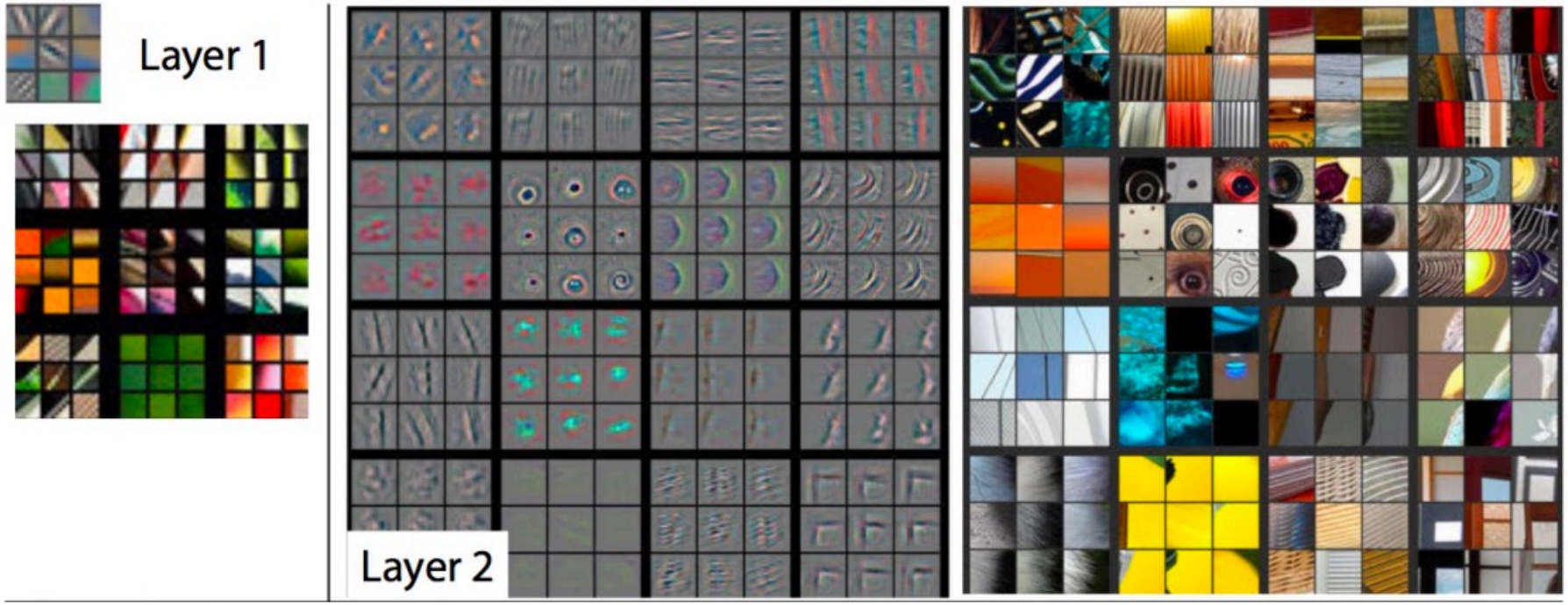
DARPA, Explainable AI Project 2017 [2]

Why explainable?

- The Curiosity of Humans
  - What has the classifier learned from the data?
- Limitations of Machines
  - Human knowledge as a complement
- Moral and Legal Issues
  - The "right to explanation"
  - Fairness (non-discrimination)

Why explainable?

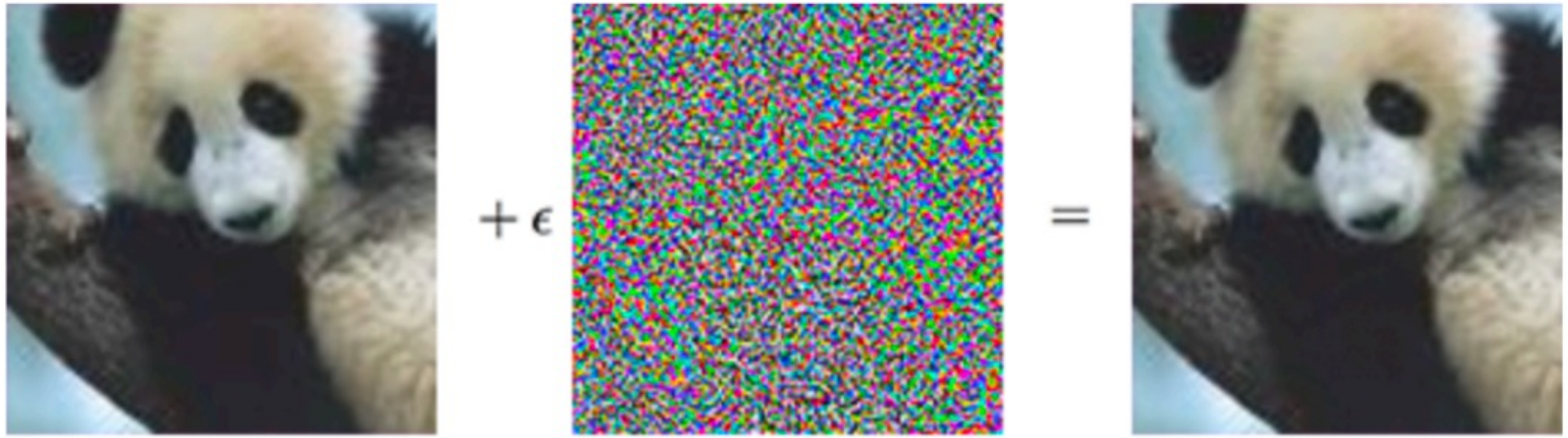
- The Curiosity of Humans
  - What has the classifier learned from the data?



Zeiler and Fergus 2014

Why explainable?

- Limitations of Machines
  - Human knowledge as a complement
  - Robustness of the model



"panda"

57.7% confidence

"gibbon"

99.3% confidence

Adversarial examples attack

(https://blog.openai.com/adversarial-example-research/)

Why explainable?

- Moral and Legal Issues
  - The "right to explanation"

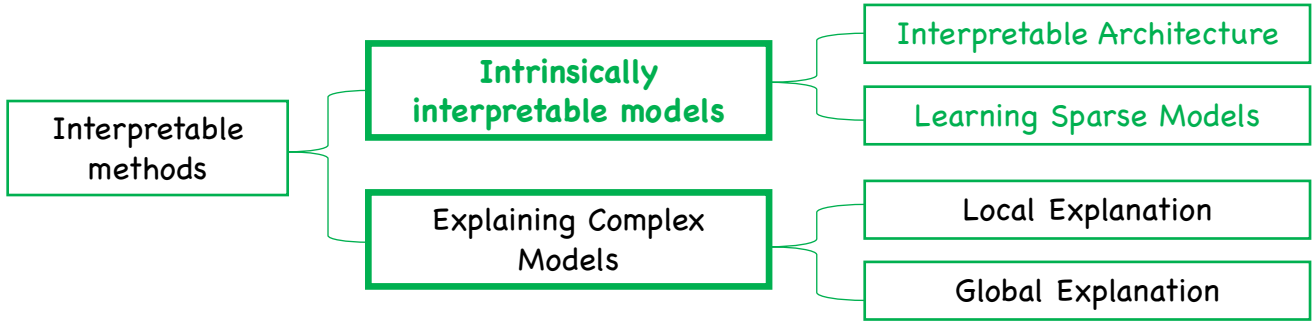
    The EU general data protection regulation (GDPR 2018) Recital 71:

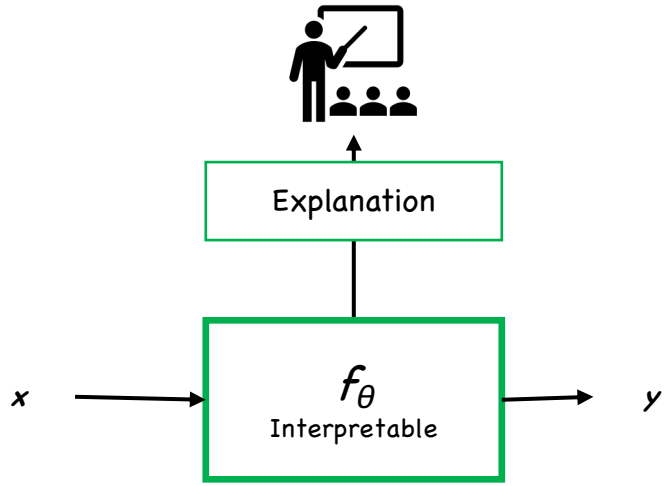
    In any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision.

  - Fairness (non-discrimination)
    - Classification systems for loan approval
    - Resume filter for hiring

Explainable Classifier

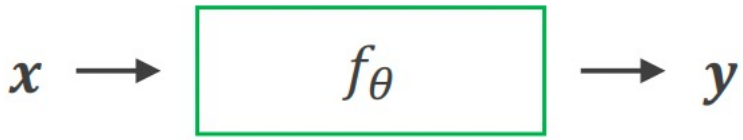
Two strategies to provide interpretability





Interpretable Classifiers

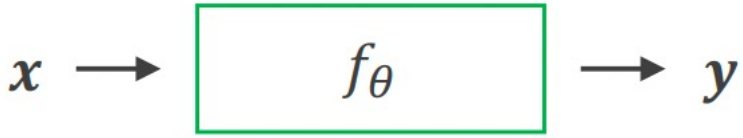
- Classifiers that are commonly recognized as understandable, and hence need little effort to explain them



- Interpretable architecture:
  - f consists of computation blocks that are easy to understand
  - E.g., decision trees
- Learning sparse models:
  - |θ| is smaller so that it is easy to understand
  - E.g., simplication

Interpretable Classifiers

- Classifiers that are commonly recognized as understandable, and hence need little effort to explain them



Categories		Related Papers	Remarks
Interpretable Classifiers	Interpretable Architecture	Decision Trees [7], Rule Lists [27, 59], Rule Sets [60]	rule-based
		Linear Models [6]	linear
		kNNs [12, 22]	instance-based
	Learning Sparse Models	Decision Trees [43], Sparse SVMs [11], Sparse CNNs [29]	simplification
		Sparsity by Bayesian [56], Integer Models [55, 58]	direct-sparsity

Not as explainable as they seemed to be!

Interpretable Classifiers

- Interpretable Architecture – Classic Methods
- kNN (instance-based)
  - t is classified as Y because a, b, and c are similar to t.
  - Limits: lack close instances to t
- Decision Tree (rule-based)
  - Seem to be interpretable

  - Limits: performance vs. explainability

Explaining Complex Classifiers

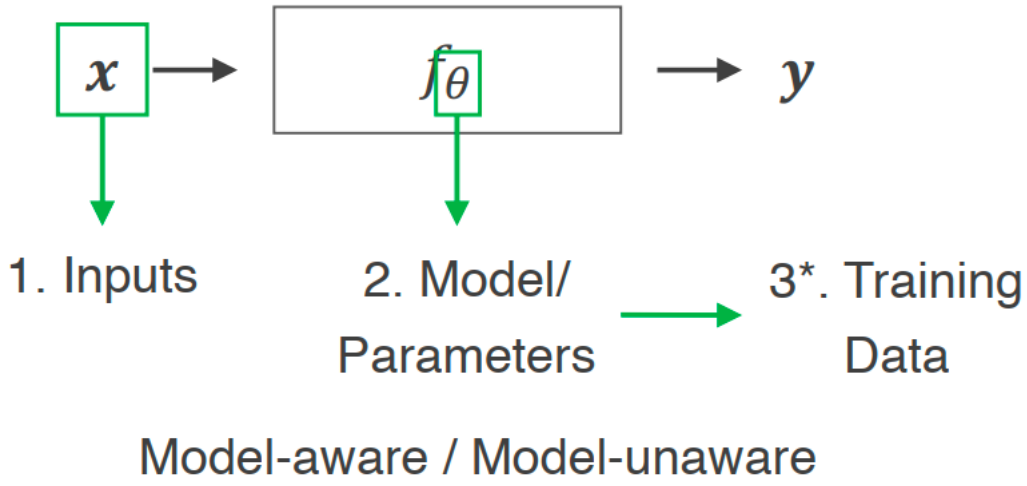
- What are explanations of classifiers?
  - Cognitive Science (Lombrozo 2006):
  - Explanations are characterized as arguments that demonstrate all or a subset of the causes of the explanandum (the subject being explained), usually following deductions from natural laws or empirical conditions)

- What is the explanandum?
  - The prediction of the classifier. (Local explanation)
    - Why is x classified as y?
  - The classifier itself. (Global explanation)
    - What has the classifier learned in general?

A summary of local explanations on $\mathcal{X}$

Explaining Complex Classifiers

- What is explanations?
  - Cognitive Science (Lombrozo 2006):
    - Arguments … of the causes of the explanandum …
- What are the causes of the prediction(s) of a classifier?



Explaining Complex Classifiers

Categories			Related Papers	Remarks
Explanations of Classifiers	Local	Model-unaware	Sensitivity Analysis [50, 28, 51]	gradient-based
			LIME [46]	model induction
			Generate Visual Explanations [19]	extra labels
		Model-aware	De-convolution [65],	CNN
			Layer-wise Propagation [4],	CNN
			Prediction Difference [66],	Image
			Output Decomposition [36],	LSTM
			Direct Mapping [21]	RNN
	Global	Unaware	Greedy-pick [46], Top-k [65]	sampling
		Model-aware	Partition Hidden Space [14, 44],	NN
			Activation maximization [13, 50],	CNN
			Network Dissection [5]	CNN

Gradients (ImageNet 2013)
(Simonyan et al. 2014)

$\frac{\partial y_i}{\partial \boldsymbol{x}}(\boldsymbol{x}_{test})$

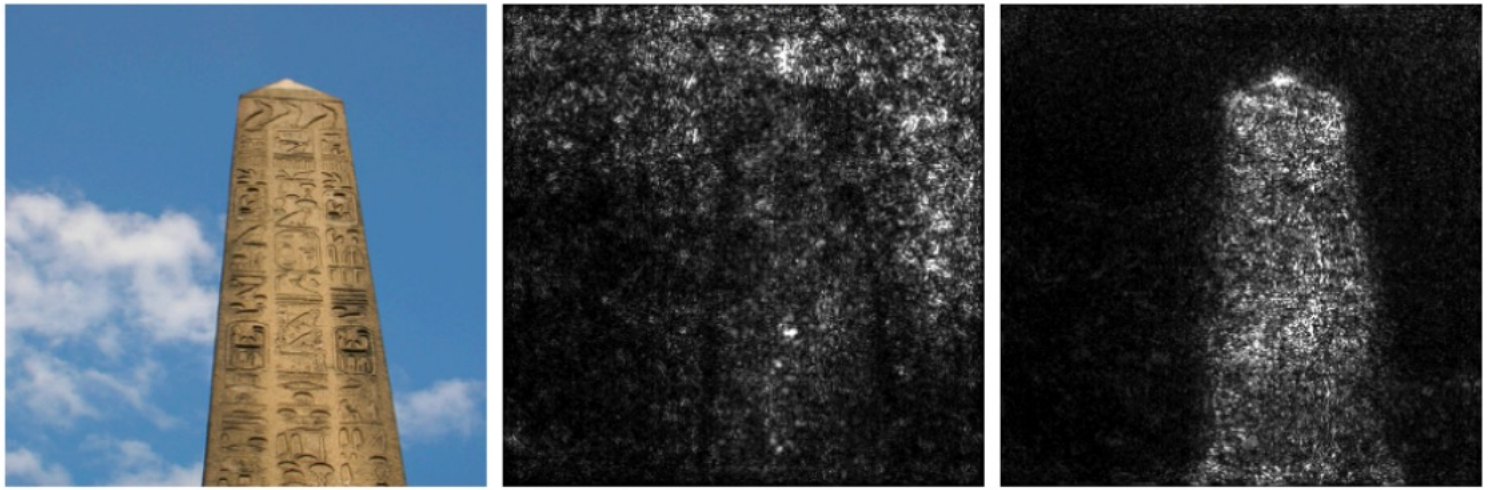
1. Too noisy!
2. High grad => important?

Local Explanations

Sensitivity Analysis -
Why is x classified as y?

Local Explanations

- Sensitivity Analysis – Why is x classified as y?

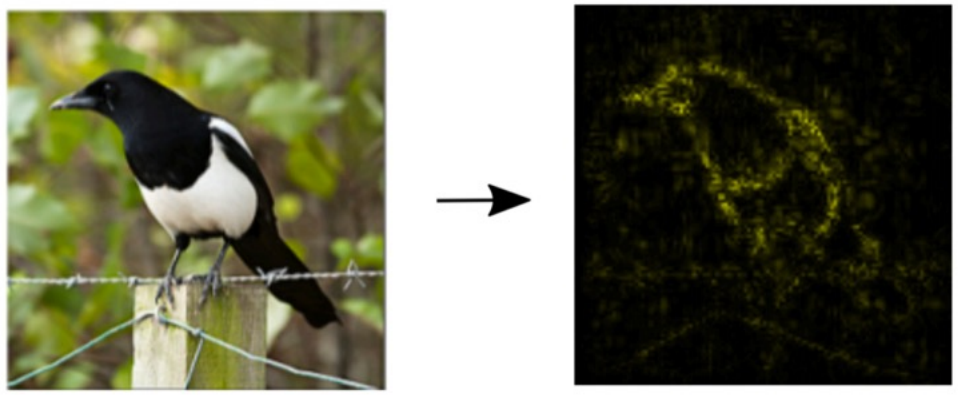


SmoothGrad (Smilkov et al. 2017)
Sampling noisy images and average the gradient map $\frac{1}{n}\sum_{j=1}^{n}\frac{\partial y_i}{\partial \boldsymbol{x}}(\boldsymbol{x}_{test}+\mathcal{N}(0,\sigma^2))$

Limit: Expensive; Non-deterministic

Local Model-Aware Explanations

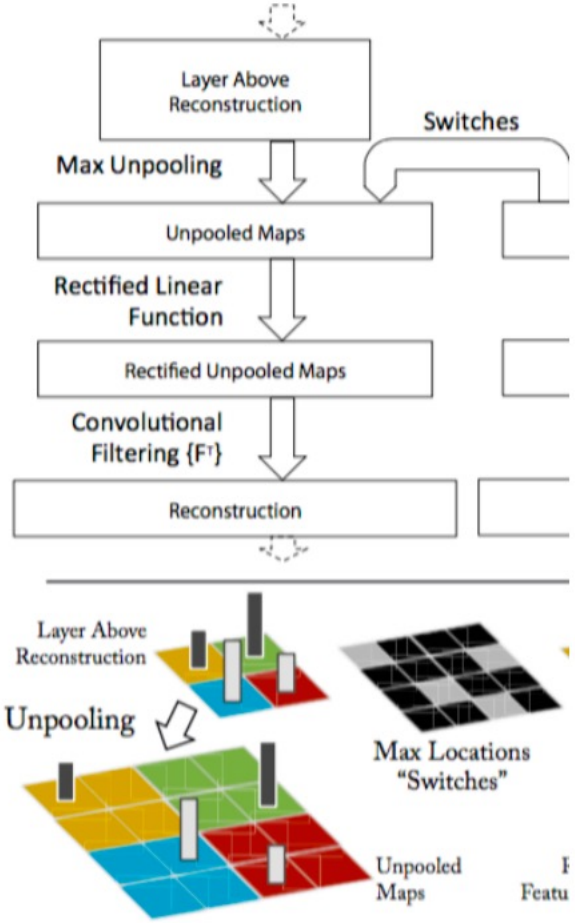
- Utilizing the structure of the model -- CNN



De-convolution (Zeiler and Fergus 2014):
Inverse operations of different layers

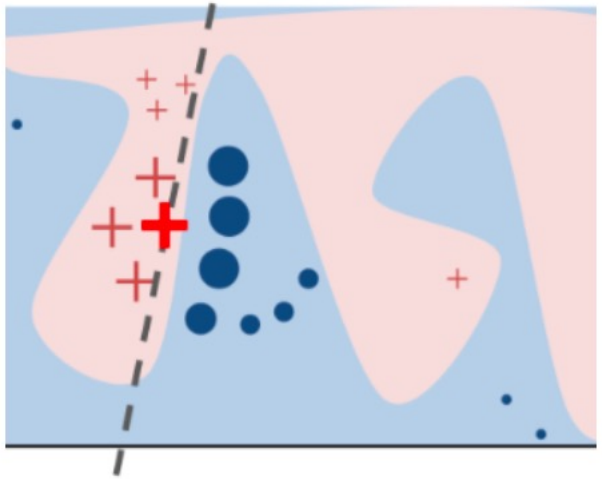
Pros:
- Can apply to neurons
- Better explanations

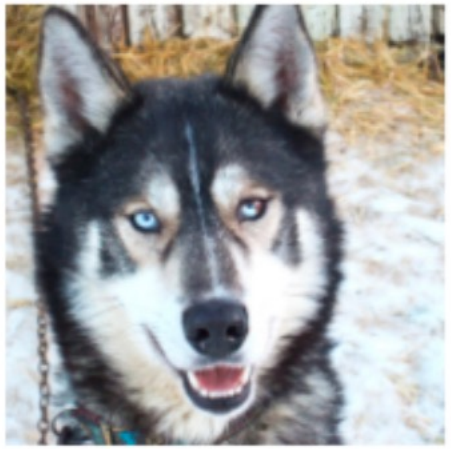
Cons:
- Only for layer-wise, invertible models
- No relations



Local Model-Unaware Explanations

- Model Induction





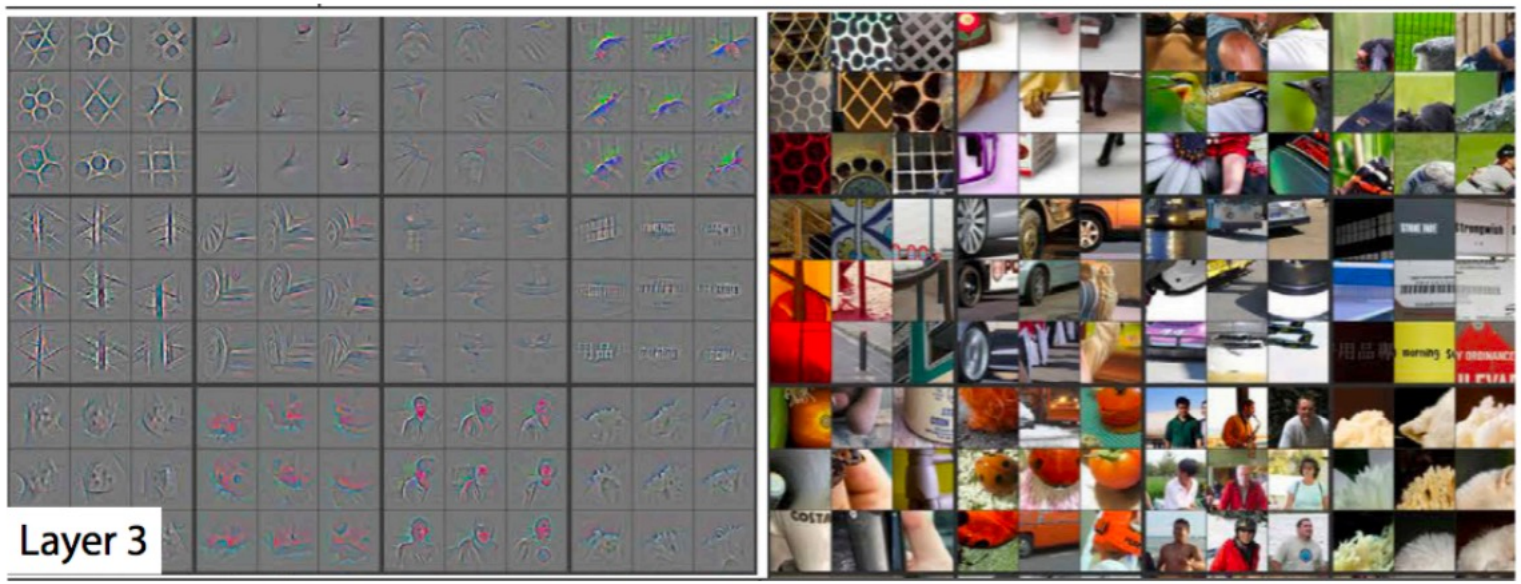
(a) Husky classified as wolf

(b) Explanation

- Locally approximate a complex classifier using a simple one (linear) 0-1 explanation (Ribeiro et al. 2016)
- Limits
  - Induction of a simple one is by random sampling local points
  - Expensive
  - Generating images patch require extract efforts

Global Model-unaware Explanations

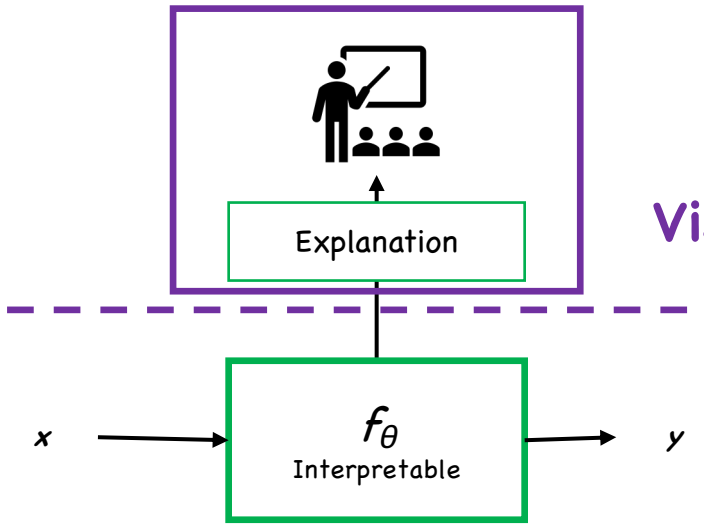
- Sampling local explanations
- Select top-k instances with max activations (Zeiler and Fergus 2014)



- Select local explanations that greedily covers the most important features (Ribeiro et al. 2016)
- Limit to the data; special case; expensive

Explainable Classifiers

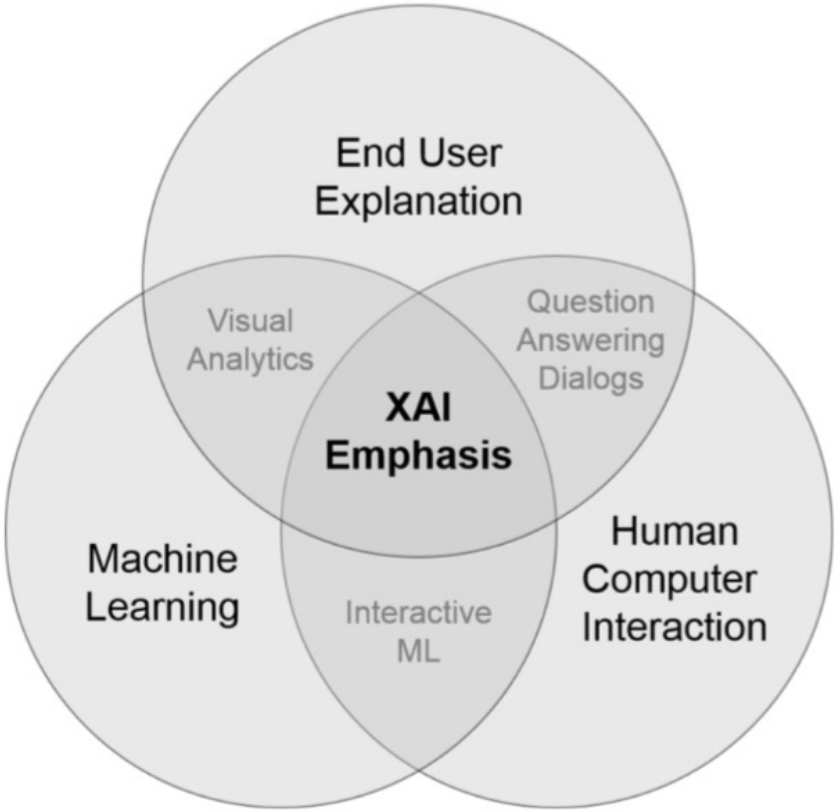
The lack of human in the study!



Visualization for Interpretable Machine Learning Models

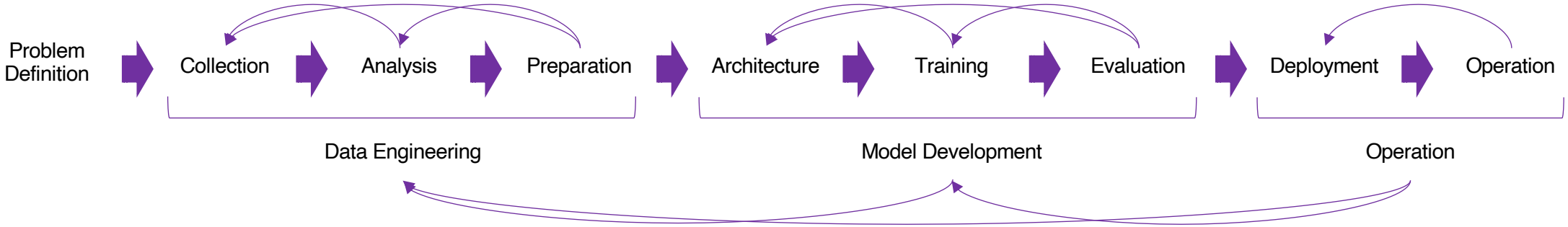
Visualization for Explainable Classifiers

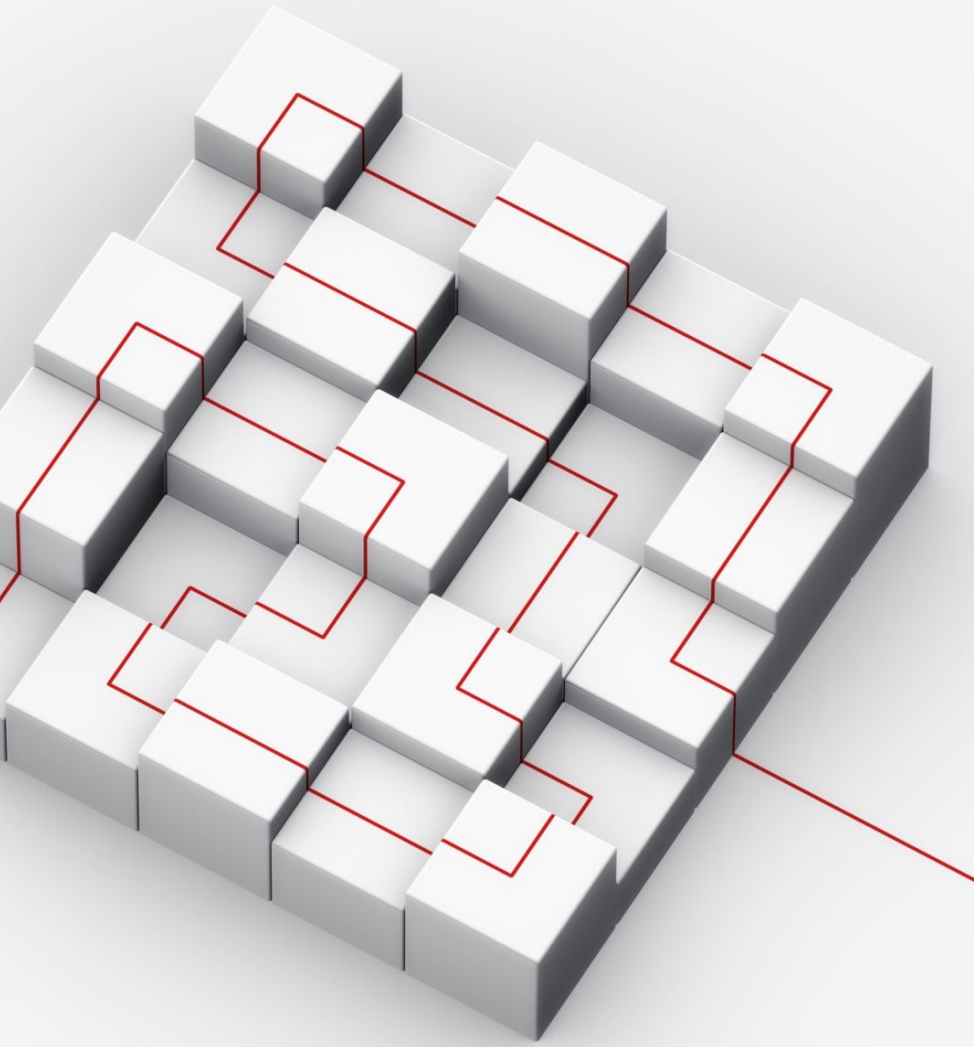
- What roles is visualization playing in explainable classifiers?



DARPA, Explainable AI Project 2017

The Life Cycle of a Classifier



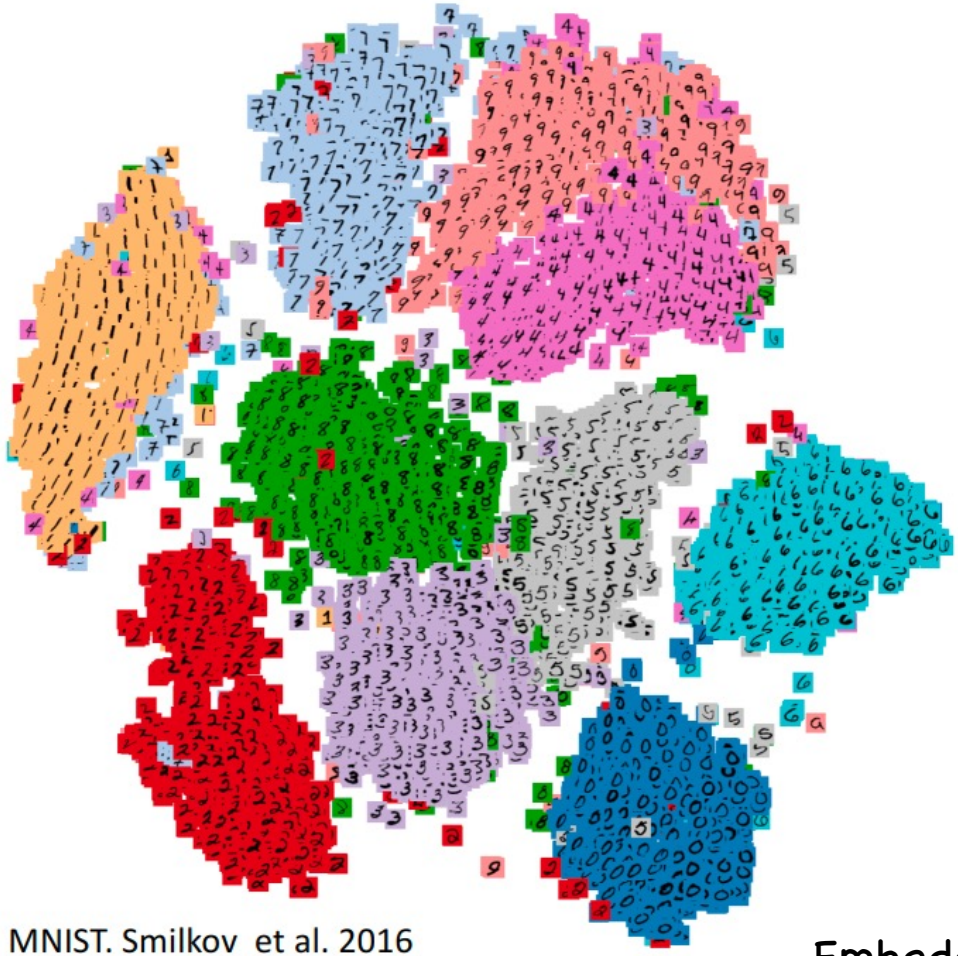


What are the problems?

- Vis for Exploratory Data Analysis
  - What does my dataset look like? Any mislabels?
- Vis for Model Development
  - Architecture: what is the classifier? How to compute?
  - Training: How the model gradually improves? How to diagnose?
  - Evaluation: What has the model learned from the data?
  - Comparison: Which classifier should I choose?
- Vis for Operation
  - Deploy: How to establish users' trust?
  - Operation: How to identify possible failure?

Visualization for Exploratory Data Analysis

- What does my dataset look like?



MNIST. Smilkov et al. 2016

Embedding Projector

It might be difficult to classify between (3,5) and (4,9)!

Methods:

- PCA
- Multidimensional Scaling
- t-SNE

Augmenting:

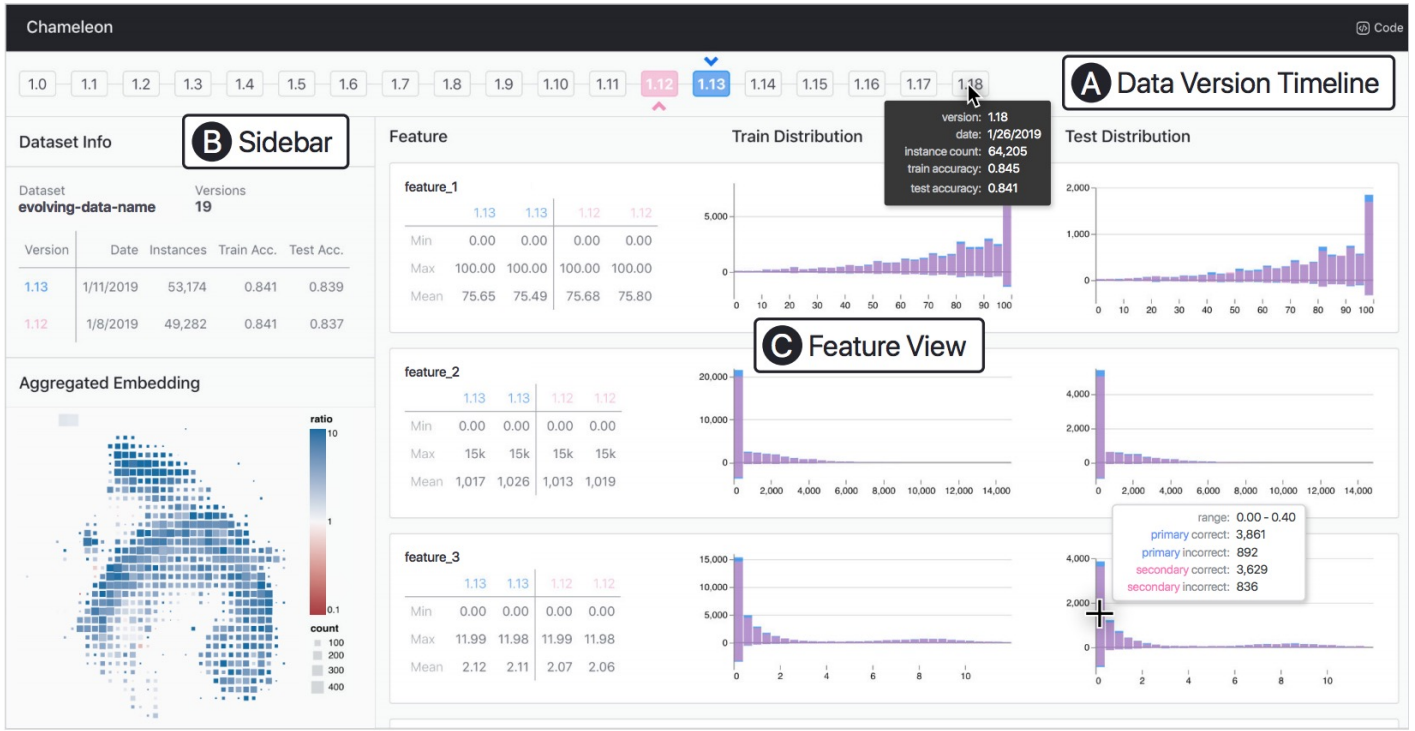
- Glyph (Smilkov et al. 2016)
- Color (Wang and Ma 2013)

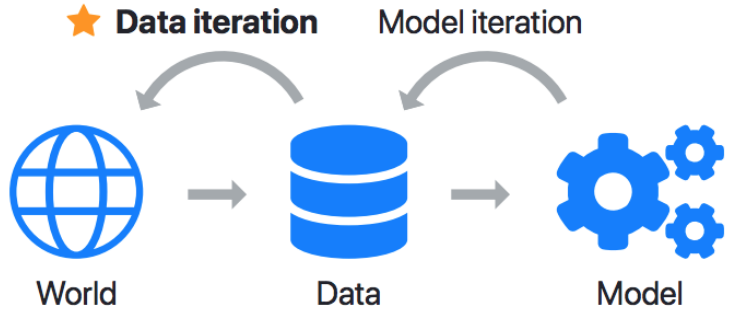
Visualization for Exploratory Data Analysis

上海科技大学 ShanghaiTech University

Understanding and visualizing data iteration in machine learning

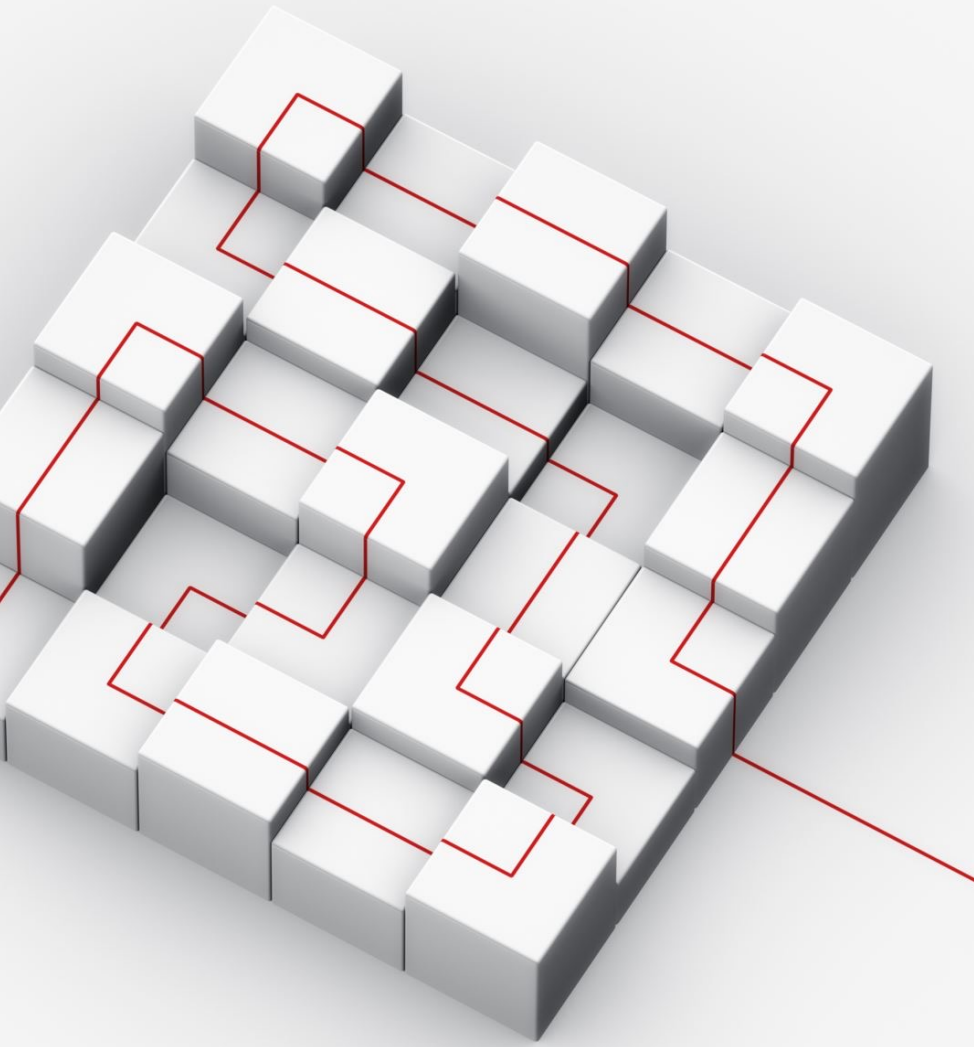
- While prior visualization tools for ML primarily focus on modeling, ML practitioners reveal that they improve model performance frequently by iterating on their data (e.g., collecting new data, adding labels) rather than their models





- The Data Version Timeline (A) lists data versions across the top of the interface and allows users to select a primary and a secondary version to visualize below
- The Sidebar (B) shows version summaries and multiples views that visualize changing instance predictions
- Practitioners can use the sidebar views to filter data in the Feature View (C), which visualizes each feature of a dataset as a histogram with both selected data versions, faceted by performance and the train/testing split

Understanding and Visualizing Data Iteration in Machine Learning Fred Hohman, Kanit Wongsuphasawat, Mary Beth Kery, Kayur Patel ACM Conference on Human Factors in Computing Systems (CHI). Honolulu, HI, USA, 2020.

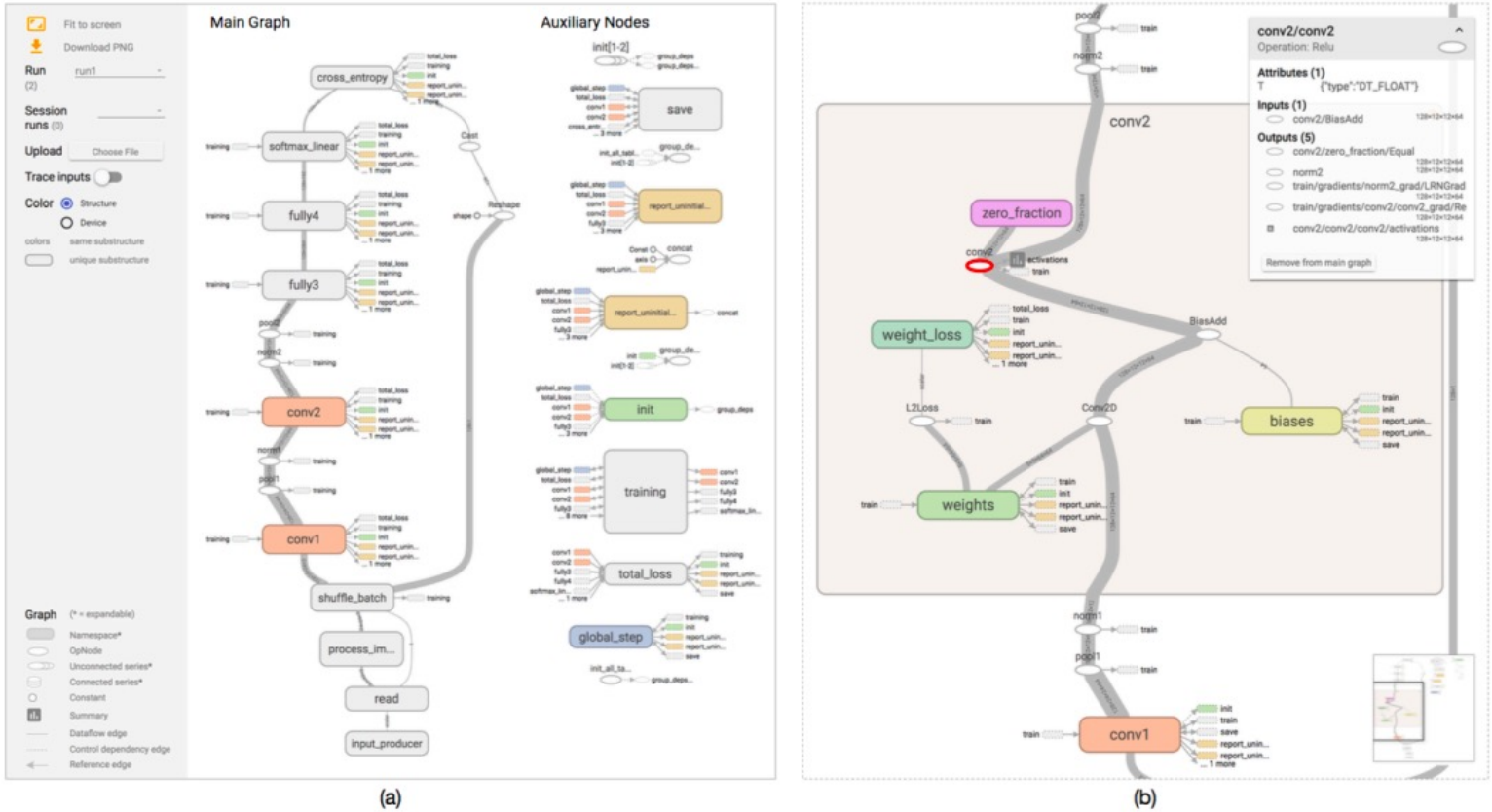


What are the problems?

- Vis for Exploratory Data Analysis
  - ~~What does my dataset look like? Any mislabels?~~
- Vis for Model Development
  - Architecture: what is the classifier? How to compute?
  - Training: How the model gradually improves? How to diagnose?
  - Evaluation: What has the model learned from the data?
  - Comparison: Which classifier should I choose?
- Vis for Operation
  - Deploy: How to establish users' trust?
  - Operation: How to identify possible failure?

Visualization for Model Development

- Architecture: How to explain the computation of a model?



K. Wongsuphasawat et al., "Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow," in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 1-12, Jan. 2018, doi: 10.1109/TVCG.2017.2744878.

#Global

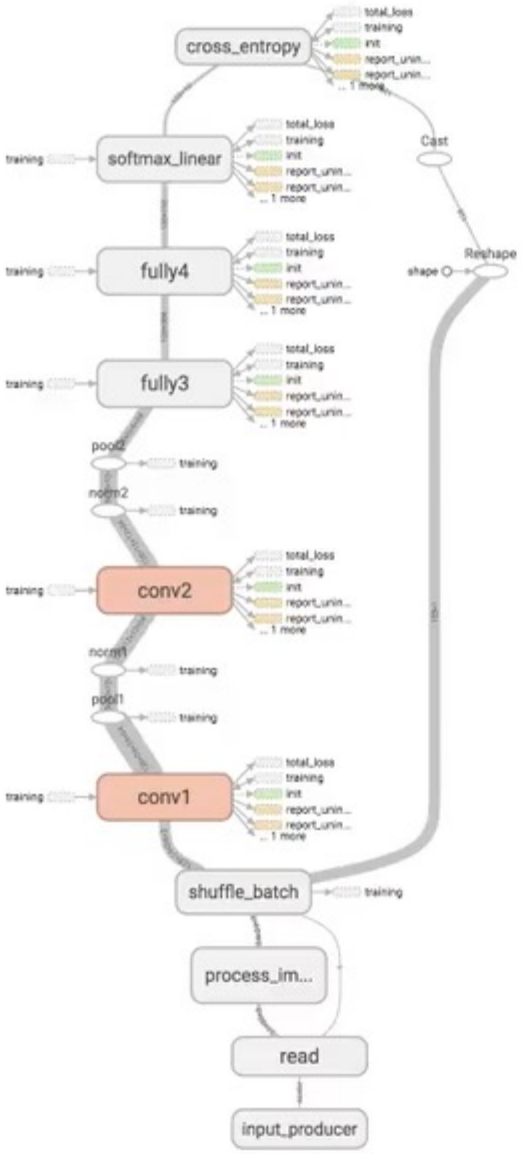
Data Flow Graph (TensorBoard). Wongsuphasawat et al. 2017

TensorBoard SCALARS IMAGES AUDIO GRAPHS DISTRIBUTIONS HISTOGRAMS EMBEDDINGS

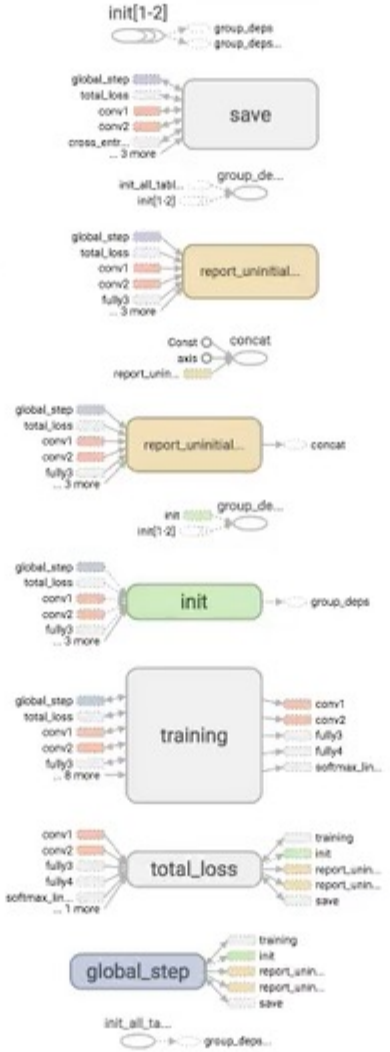
Fit to screen
Download PNG
Run (2) run1
Session runs (0)
Upload Choose File
Trace inputs
Color Structure Device
colors same substructure
unique substructure

Graph (* = expandable)
Namespace*
OpNode
Unconnected series*
Connected series*
Constant
Summary
Dataflow edge
Control dependency edge
Reference edge

Main Graph



Auxiliary Nodes



Visualization for Model Development

- Architecture: How to explain the computation of a model?
- What are the specific tasks?
  - Show an overview of the high-level components and their relationships
  - Recognize similarities and differences between components
  - Examine the nested structure of a high-level component
  - Inspect details of individual operations

- What are the challenges?
  - C1. Mismatch between graph topology and semantics
    - A group of operation → A component?
  - C2. Graph heterogeneity
    - Different importance: inference ? Gradients/optimizations ? Logger/summary
  - C3. Interconnected Nodes
    - Connections between important nodes and less important nodes mess the graph

Data Flow Graph (TensorBoard). Wongsuphasawat et al. 2017

What are the problems?

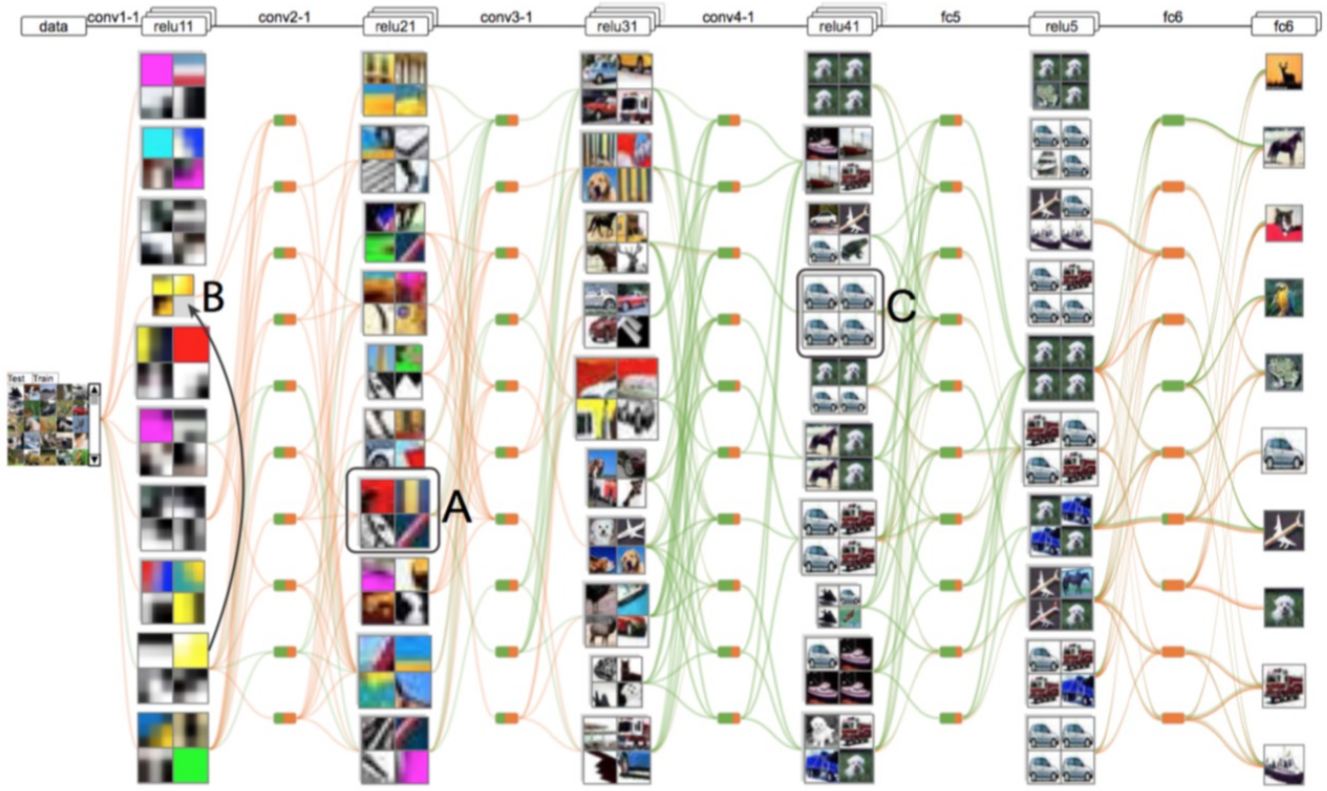
- Vis for Exploratory Data Analysis
  - ~~What does my dataset look like? Any mislabels?~~
- Vis for Model Development
  - ~~Architecture: What is the classifier? How to compute?~~
  - Training: How the model gradually improves? How to diagnose?
  - Evaluation: What has the model learned from the data?
  - Comparison: Which classifier should I choose?
- Vis for Operation
  - Deploy: How to establish users' trust?
  - Operation: How to identify possible failure?

Visualization for Model Development

上海科技大学 ShanghaiTech University

Training: Why the training fails? Analyzing CNN snapshots

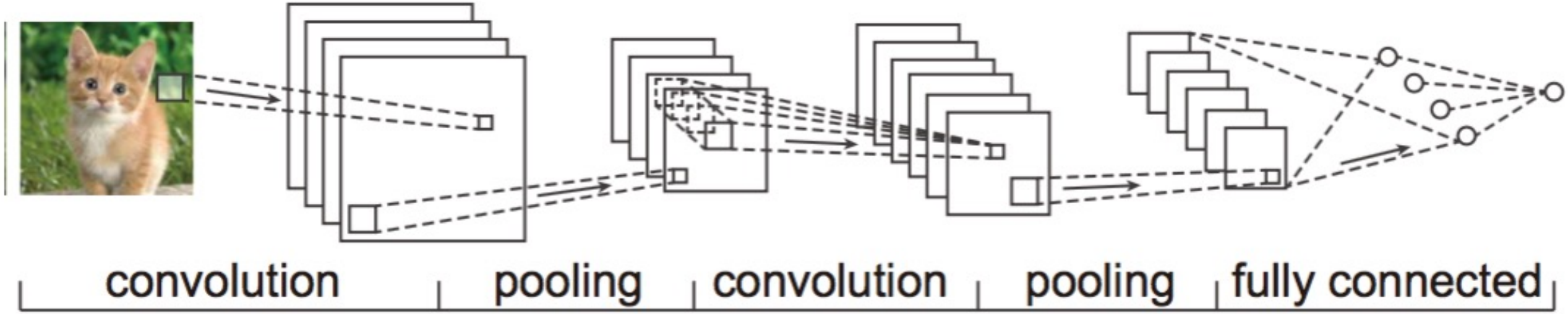
Formulates a CNN as a directed acyclic graph and proposed hybrid visualization to reveal features learned by neurons, which focus on comparing how the depth and width of the network affect the training result

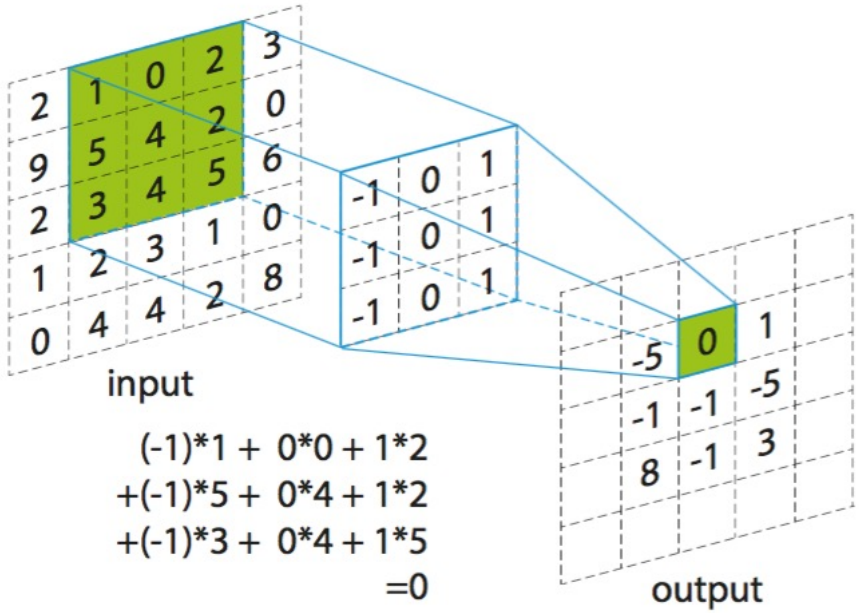


A visualization of a CNN with a large number of layers and neurons using neuron clustering and edge aggregation

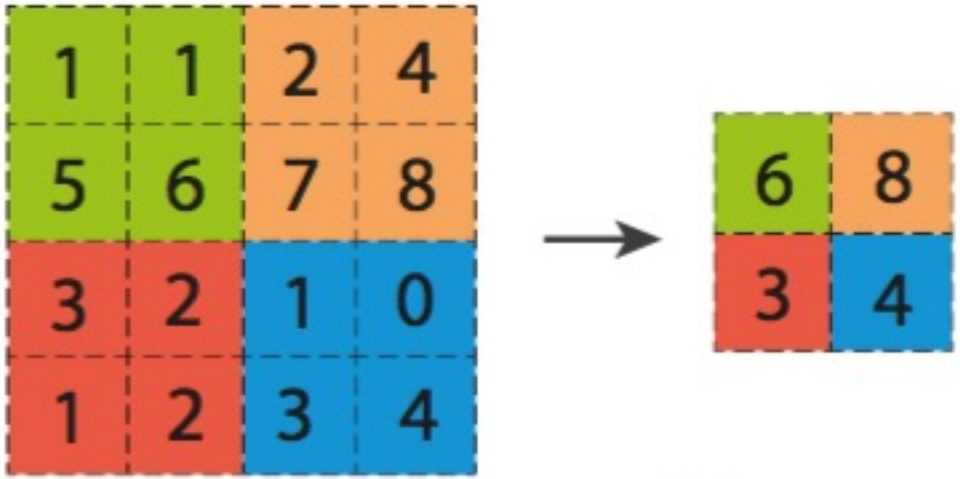
Towards Better Analysis of Deep Convolutional Neural Networks. Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, Shixia Liu. IEEE Transactions on Visualization and Computer Graphics. 23(1): 91-100, 2018.

Visualization for Model Development



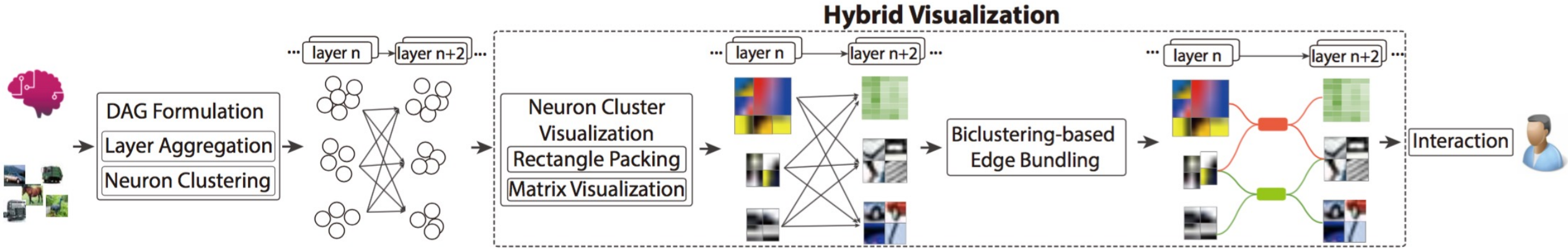


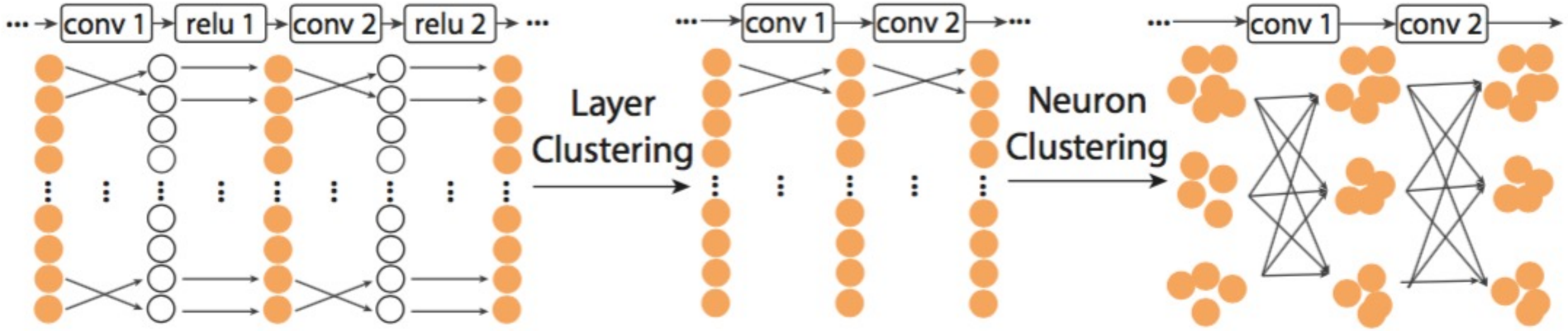
(-1)*1 + 0*0 + 1*2
+(-1)*5 + 0*4 + 1*2
+(-1)*3 + 0*4 + 1*5
=0



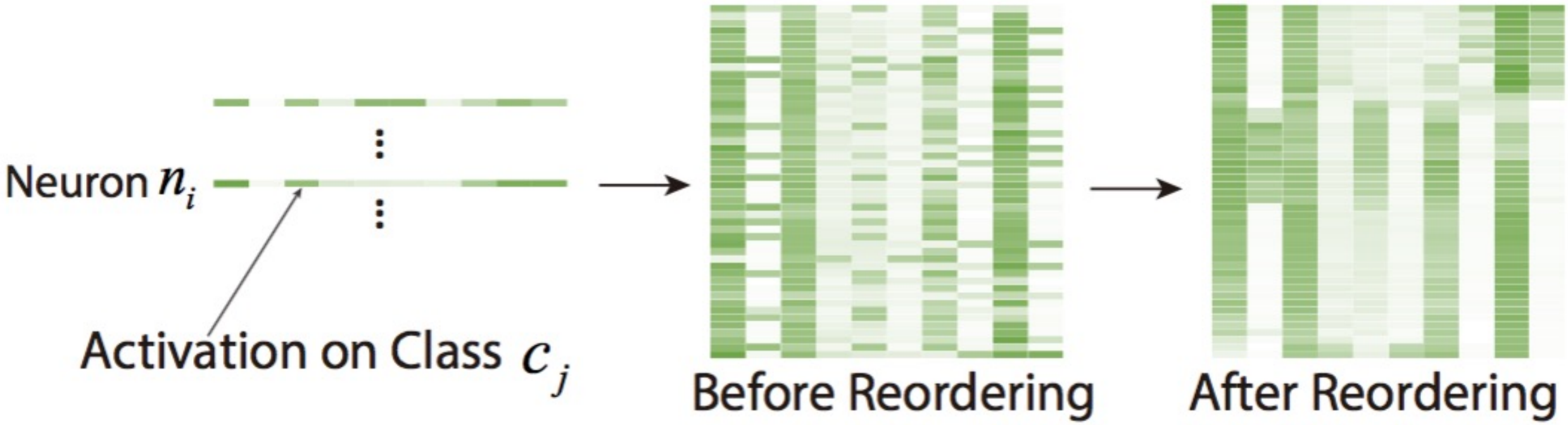
max-pooling

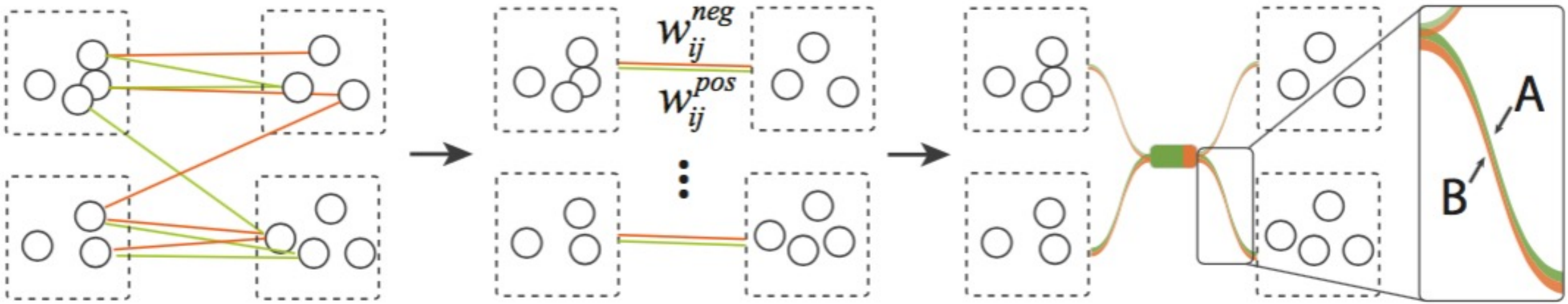
Visualization for Model Development



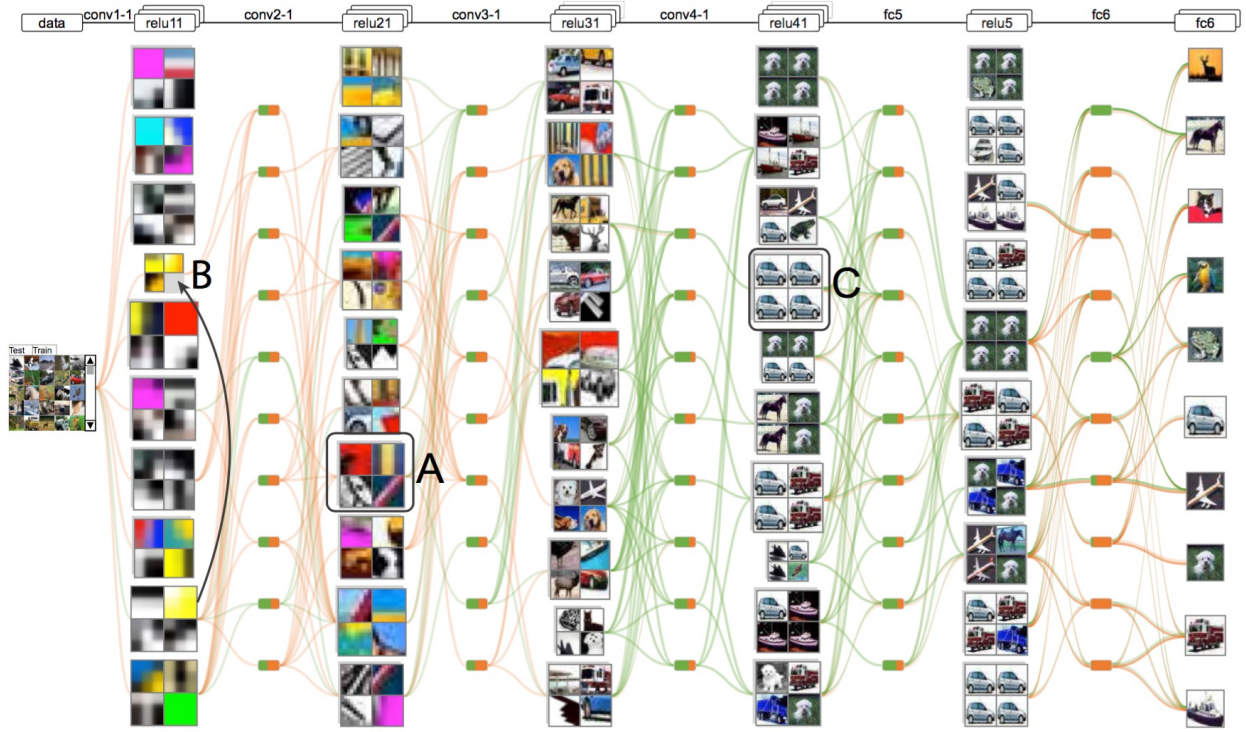


Visualization for Model Development



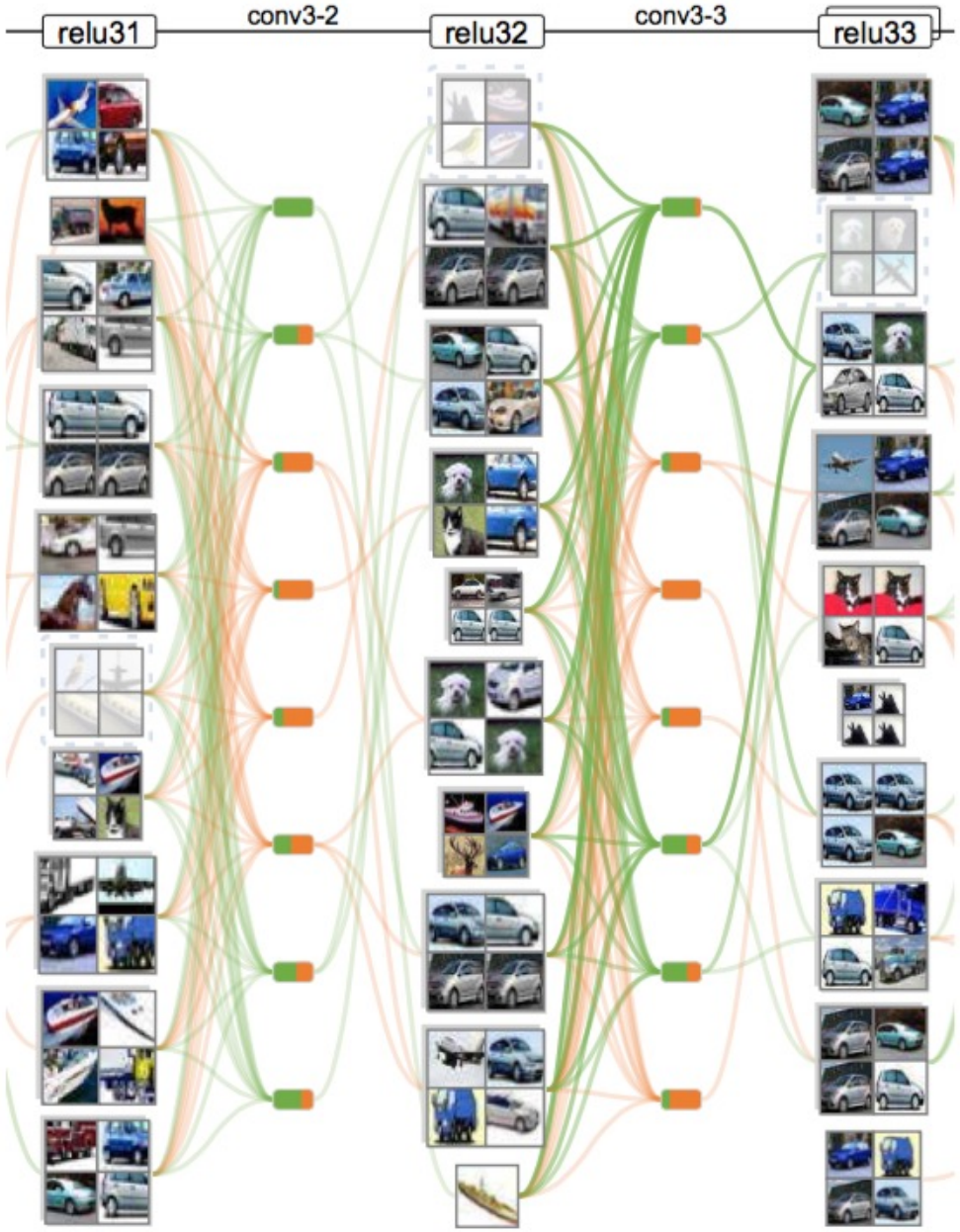


Visualization for Model Development

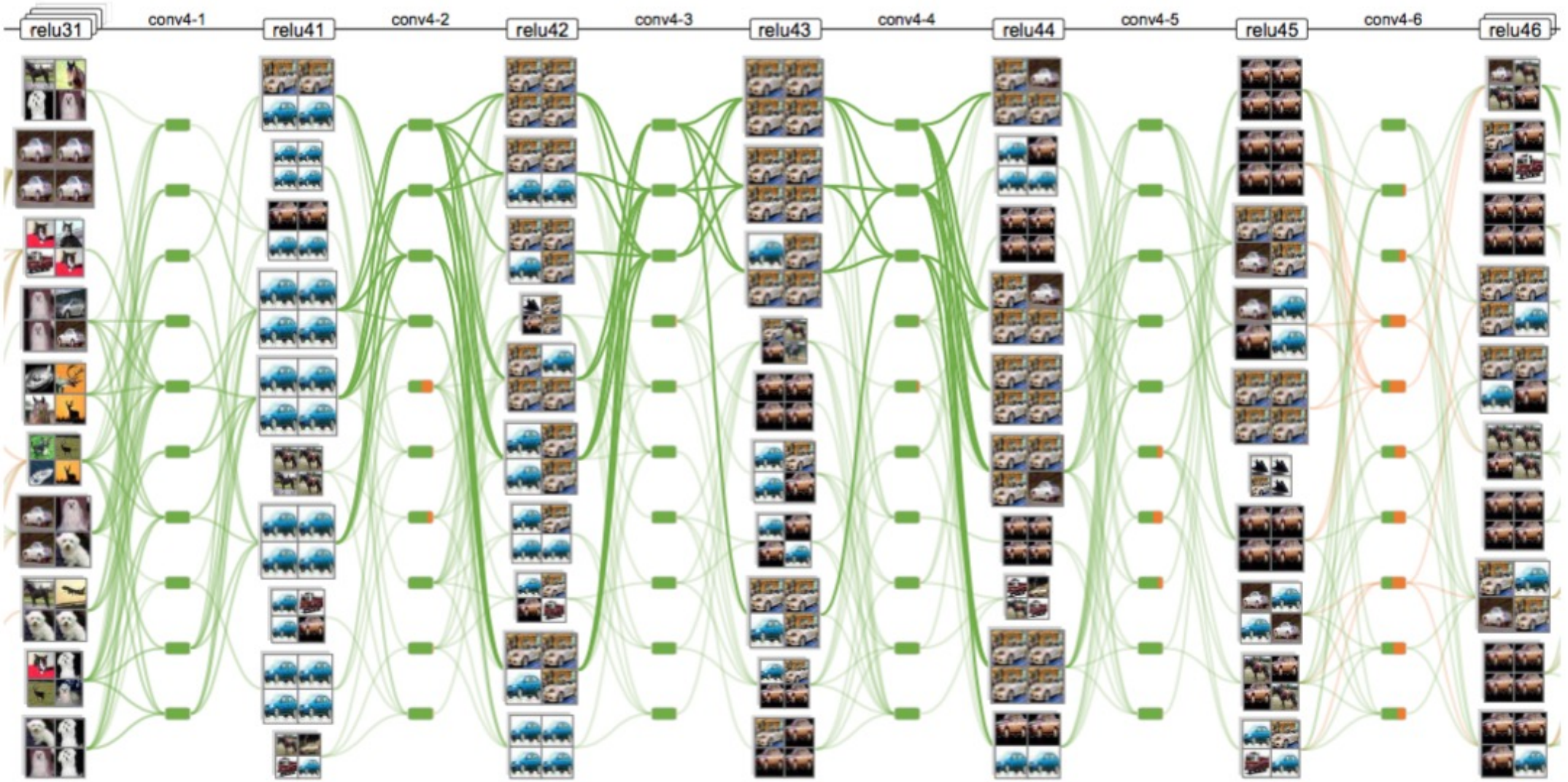


	Error	#ConvLayers
ShallowCNN	11.94%	7
BaseCNN	11.33%	10
DeepCNN	14.77%	20

Visualization for Model Development

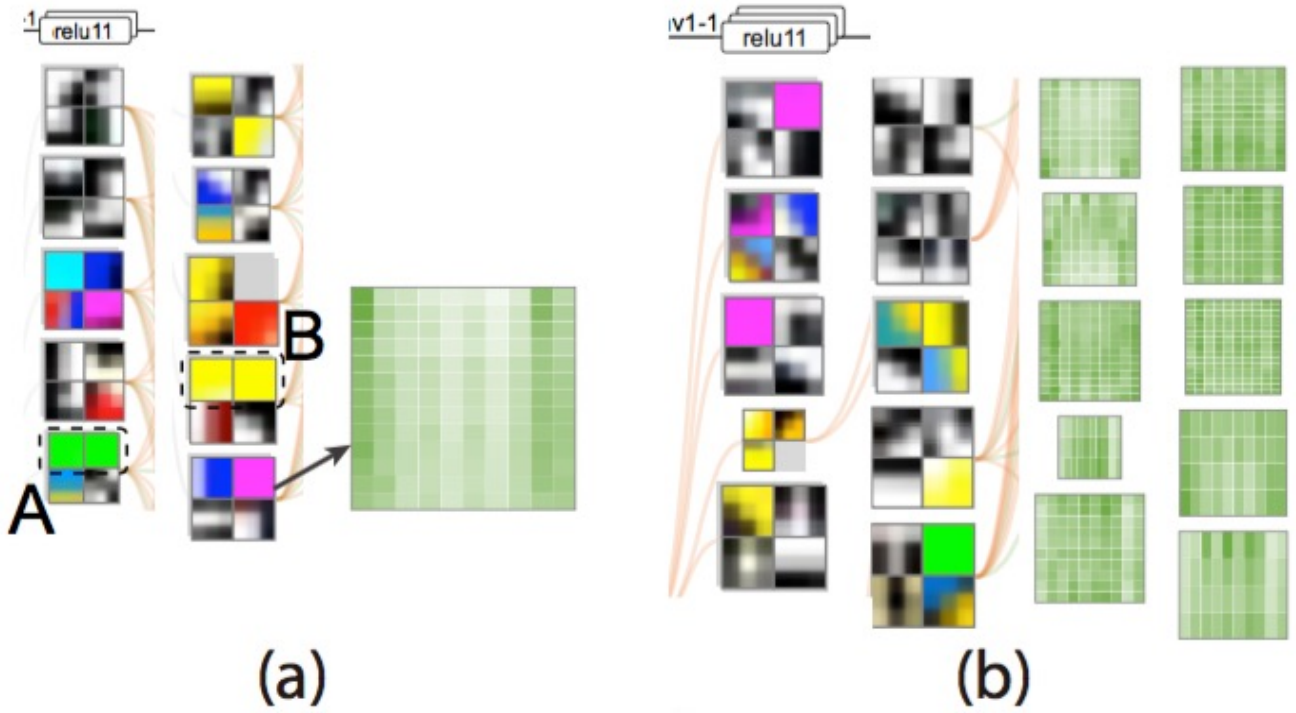


Visualization for Model Development



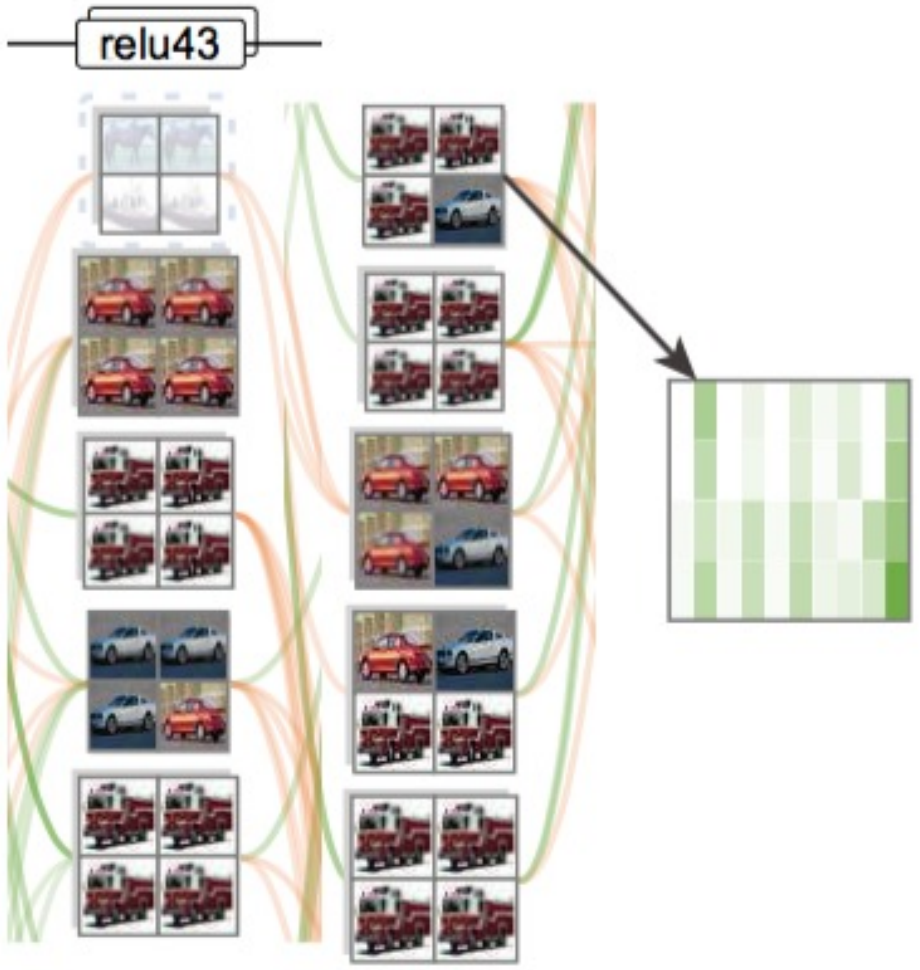
Visualization for Model Development

	Error	#params	#Training loss	Testing loss
BaseCNN * 4	12.33%	4.22M	0.04	0.51
BaseCNN * 2	11.47%	2.11M	0.07	0.43
BaseCNN	11.33%	1.05M	0.16	0.40
BaseCNN * 0.5	12.61%	0.53M	0.34	0.40
BaseCNN * 0.25	17.39%	0.26M	0.65	0.53



Visualization for Model Development

	Error	#params	#Training loss	Testing loss
BaseCNN * 4	12.33%	4.22M	0.04	0.51
BaseCNN * 2	11.47%	2.11M	0.07	0.43
BaseCNN	11.33%	1.05M	0.16	0.40
BaseCNN * 0.5	12.61%	0.53M	0.34	0.40
BaseCNN * 0.25	17.39%	0.26M	0.65	0.53



清華大學

Towards Better Analysis of Deep Convolutional Neural Networks

Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li,

Jun Zhu, Shixia Liu

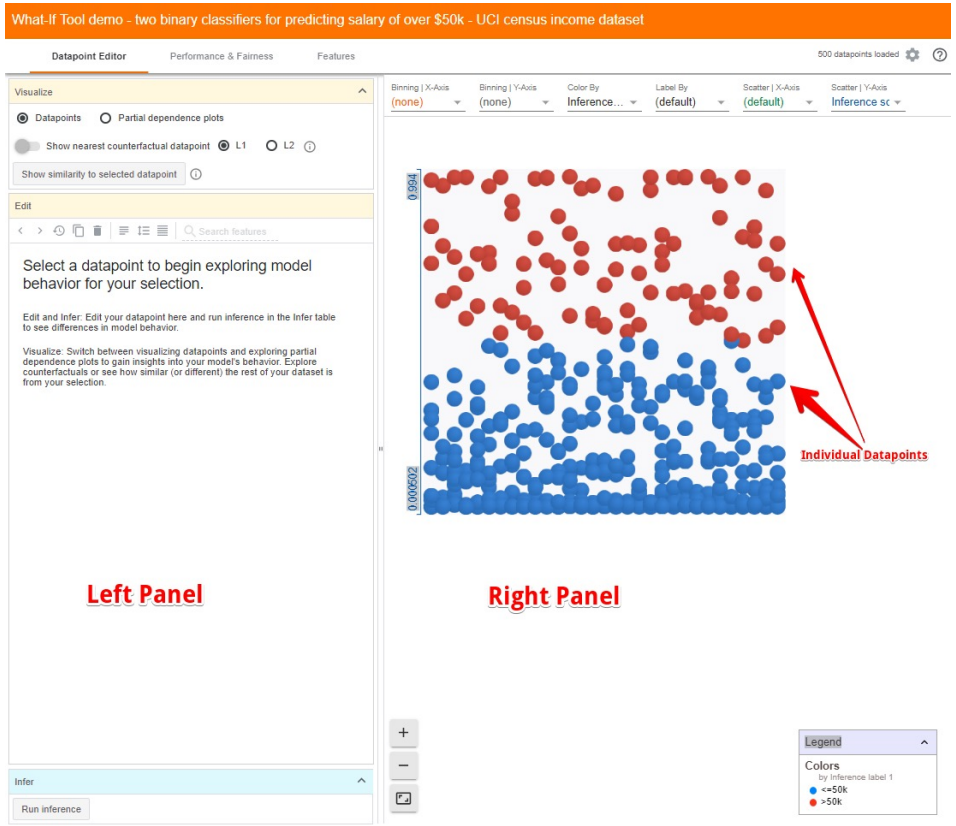
Tsinghua University

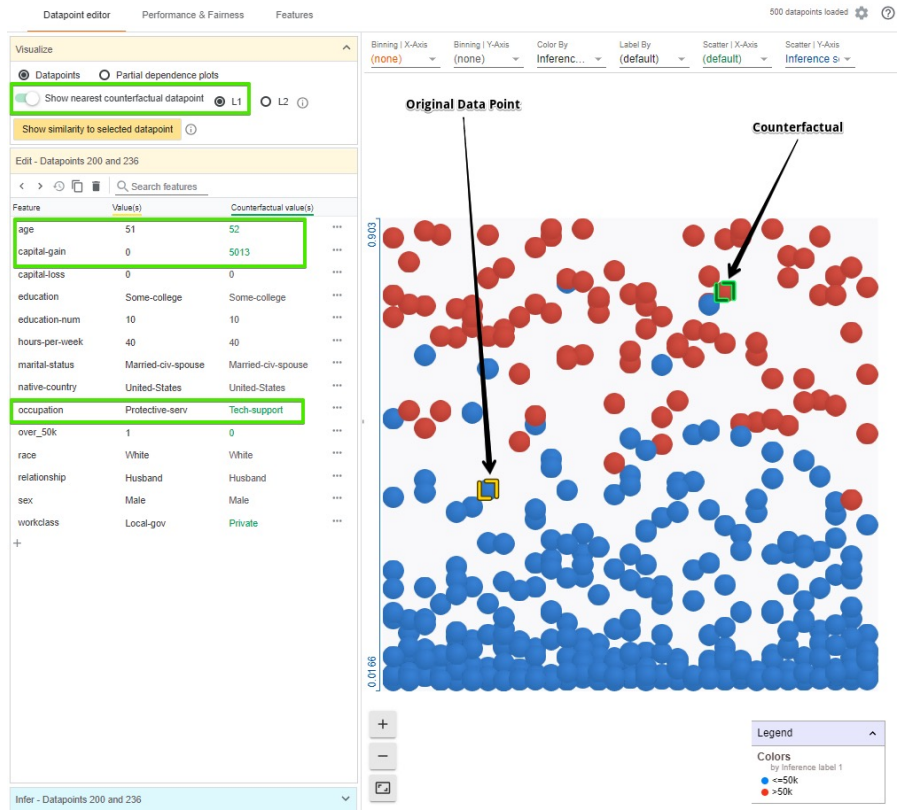
Visualization for Model Development

上海科技大学 ShanghaiTech University

Evaluation: The What-If Tool: Interactive Probing of Machine Learning Models

- Test performance in hypothetical situations, analyze the importance of different data features, and visualize model behavior across multiple models and subsets of input data
- Lets practitioners measure systems according to multiple ML fairness metrics.





Users can edit values of the data point and re-run inference to examine results
Shows the closest counterfactual value to the selected data point, highlighting the delta between the two points

Wexler, James, et al. "The What-If Tool: Interactive Probing of Machine Learning Models." *IEEE transactions on visualization and computer graphics* (2019).

People + AI Research

pair-code.github.io/what-if-tool

The What-If Tool

Interactive Probing of Machine Learning Models

James Wexler

October 2019

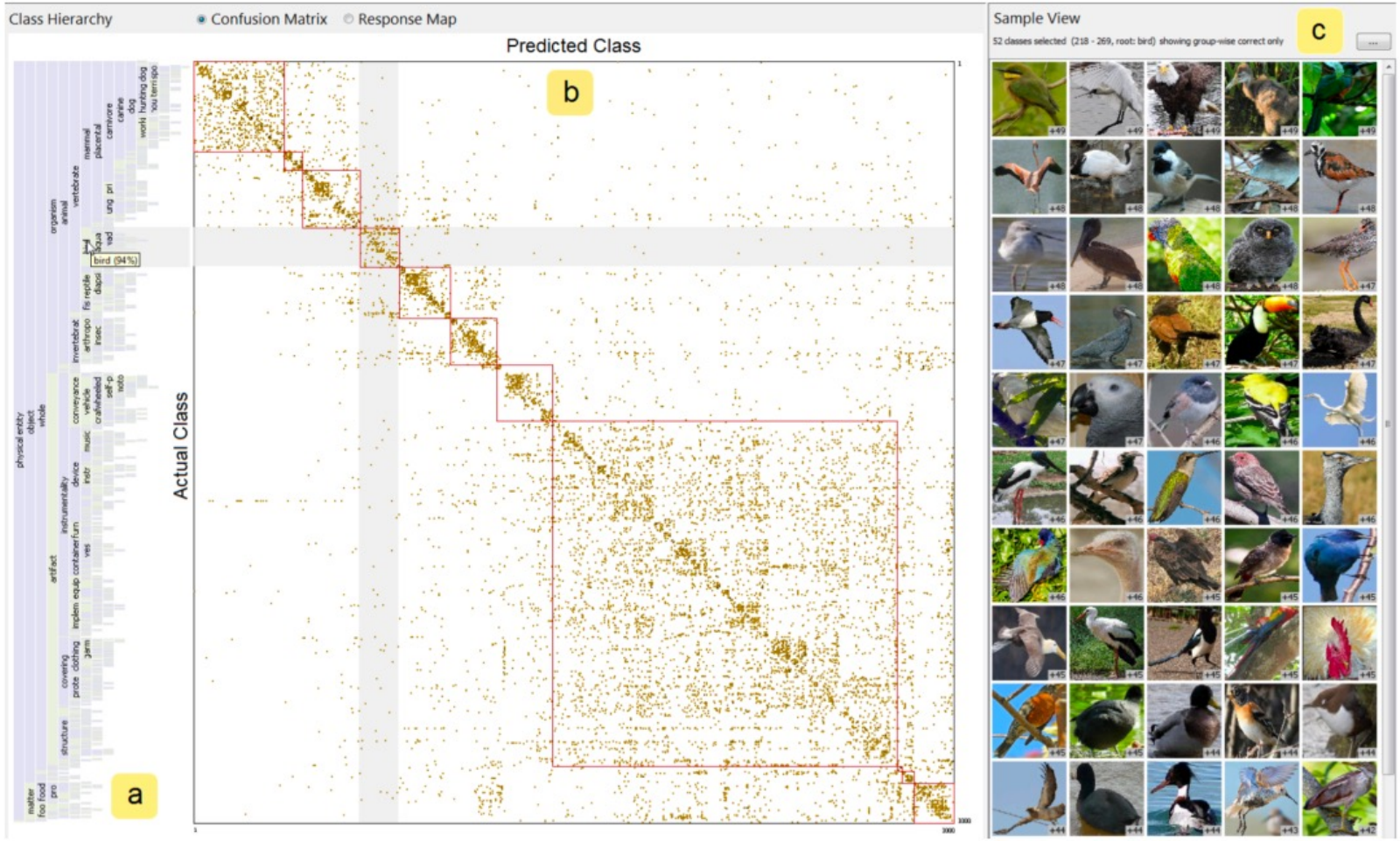
IEEE VIS VAST

What are the problems?

- Vis for Exploratory Data Analysis
  - ~~What does my dataset look like? Any mislabels?~~
- Vis for Model Development
  - ~~Architecture: What is the classifier? How to compute?~~
  - ~~Training: How the model gradually improves? How to diagnose?~~
  - Evaluation: What has the model learned from the data?
  - Comparison: Which classifier should I choose?
- Vis for Operation
  - Deploy: How to establish users' trust?
  - Operation: How to identify possible failure?

Visualization for Model Development

- Evaluation: Do CNN learn class hierarchy?



Confusion matrix of the classification results of the ImageNet using GoogleNet

#Global, #Model-unaware

Blocks. Alsallakh et al. 2017

VAST PAPER

Do Convolutional Neural Networks learn Class Hierarchy?

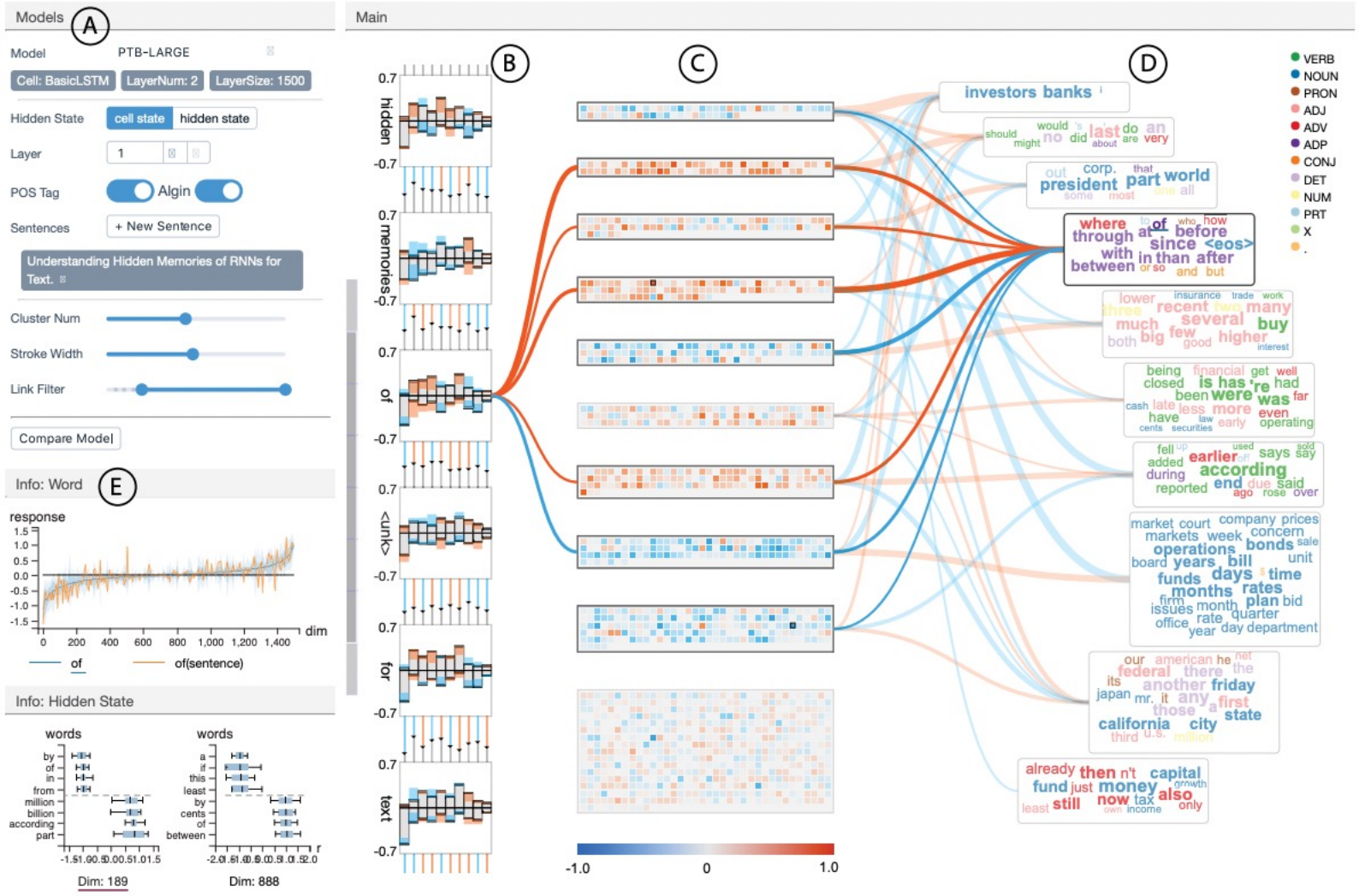
Bilal Alsallakh, Amin Jourabloo, Mao Ye, Xiaoming Liu, Liu Ren

VIS 2017
1-6 October 2017
Phoenix, Arizona, USA
ieeevis.org

Visualization for Model Development

上海科技大学 ShanghaiTech University

- Evaluation: What has an RNN learned from the data?



立志成才 报国裕民

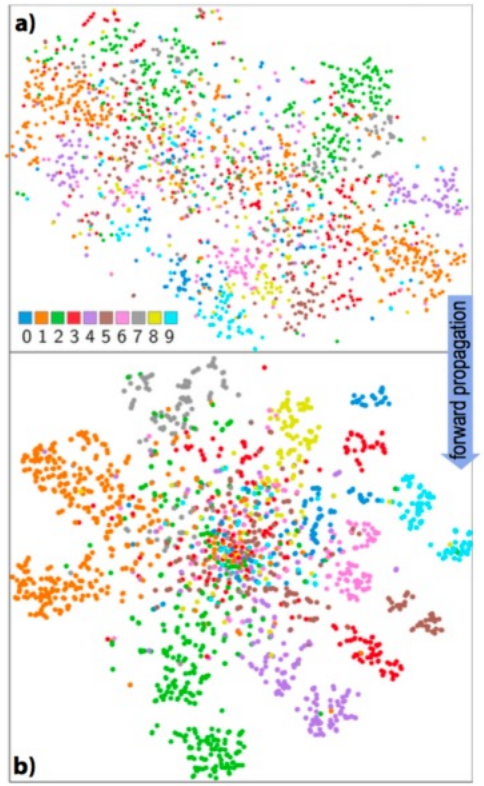
Understanding Hidden Memories of Recurrent Neural Networks

Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, Huamin Qu.

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Visualization for Model Development

- Understanding – Others (Embedding Projection)



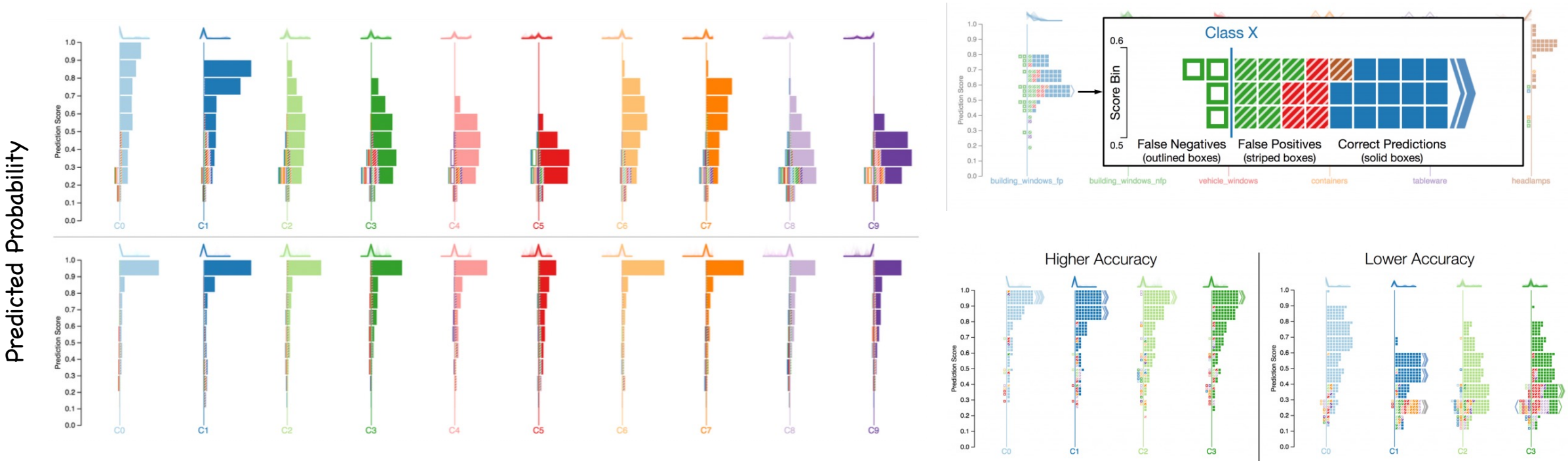
Embedding projection
SVHN test set.
Rauber et al. 2017

Multilingual translation model
t-SNE projection
Each node is a word
Johnson et al. 2016

Visualization for Model Development

Assessment & Comparison

Supporting Interactive Performance Analysis for Multiclass Classifiers



Histograms of predicted probability of instances of each class. Top: RF. Bottom: SVM. Acc: 0.87 (solid: TP, dashed-left: FP, dashed-right: FN)

Ren D, Amershi S, Lee B, et al. Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 61-70.

Squares

Supporting Interactive Performance Analysis for Multiclass Classifiers

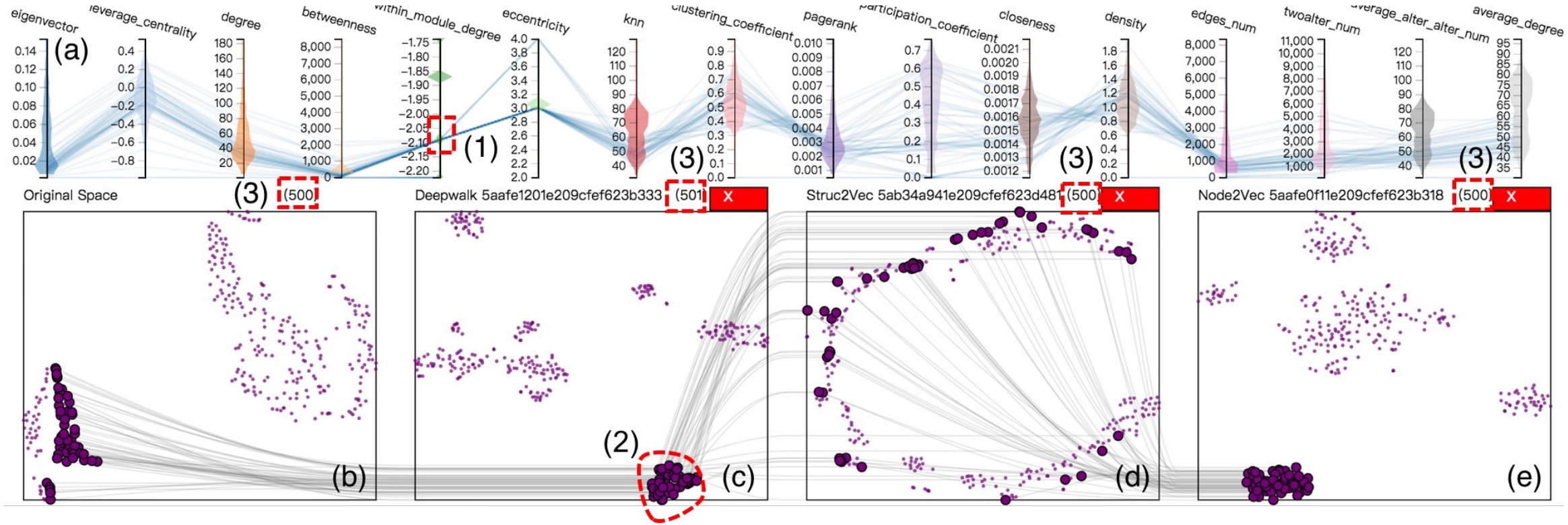
Donghao Ren, Saleema Amershi, Bongshin Lee,
Jina Suh, Jason D. Williams

Visualization for Model Development

上海科技大学 ShanghaiTech University

Assessment & Comparison

Analyze the distribution of a collection of nodes in different graph embedding spaces

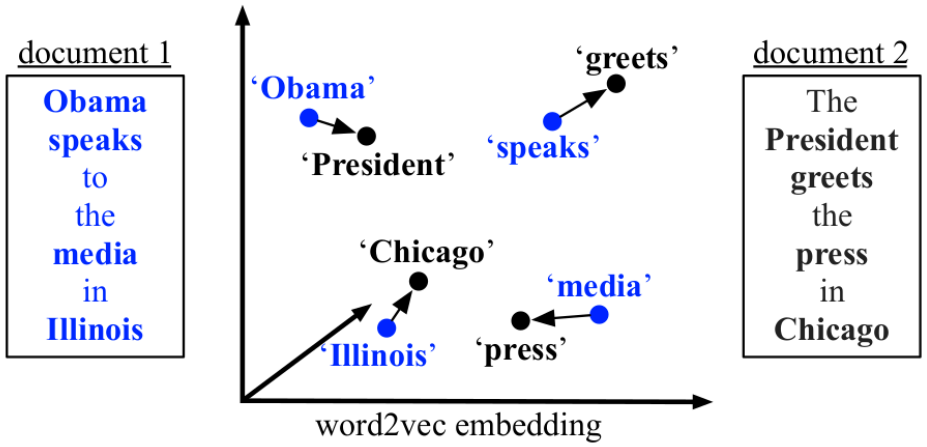


- Several clusters were formed in (c) **DeepWalk** and (e) **node2vec** embedding space, while **struc2vec** had a long and continuous "tail"
- Select one cluster in (c) **DeepWalk** space and all the correspondences were highlighted and connected
- **DeepWalk** and **node2vec** preserved the neighborhood information, e.g., nodes in a cluster well, while **struc2vec** dispersed these nodes and preserved structural information in the graph

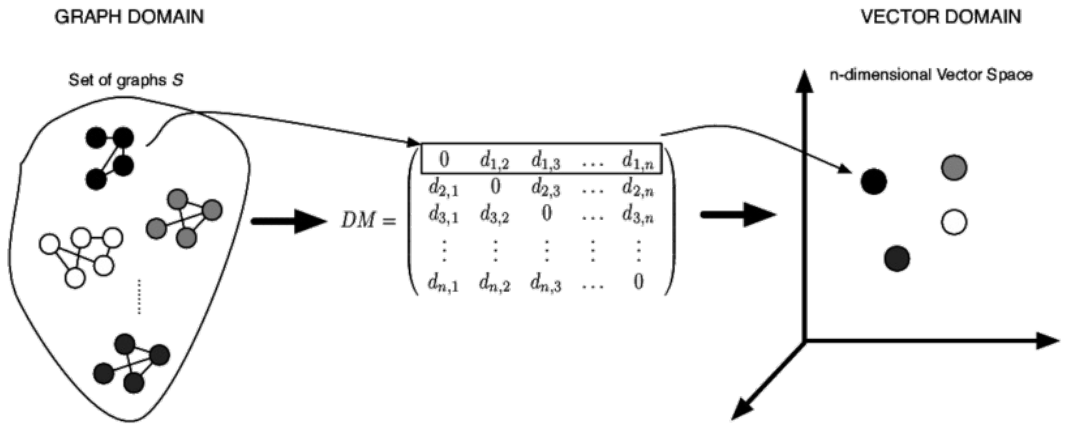
EmbeddingVis: A Visual Analytics Approach to Comparative Network Embedding Inspection.
Quan Li, Kristanto Sean N, Hammad Haleem, Qiaoan Chen, Chris Yi, and Xiaojuan Ma. Proceedings of IEEE VIS 2018 (VAST 2018), Berlin, Germany, Oct 21-26, 2018.

Network/Graph Embedding

- Map each node in a graph/network into a low-dimensional vector space
  - Similar to word embedding (word2vec) in NLP
  - Encode graph information and produce vector representation for each node
  - Vector similarity between nodes indicates node similarity

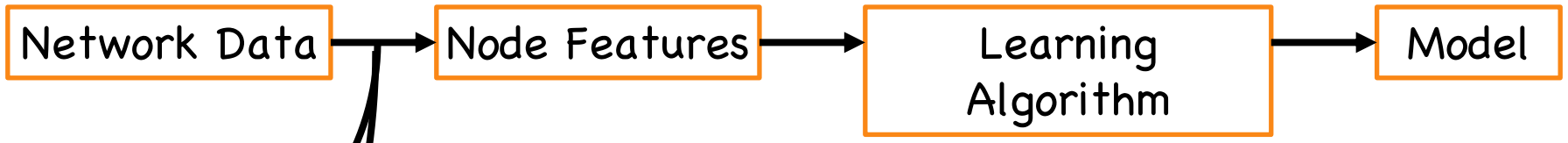


word2vec embedding



network embedding

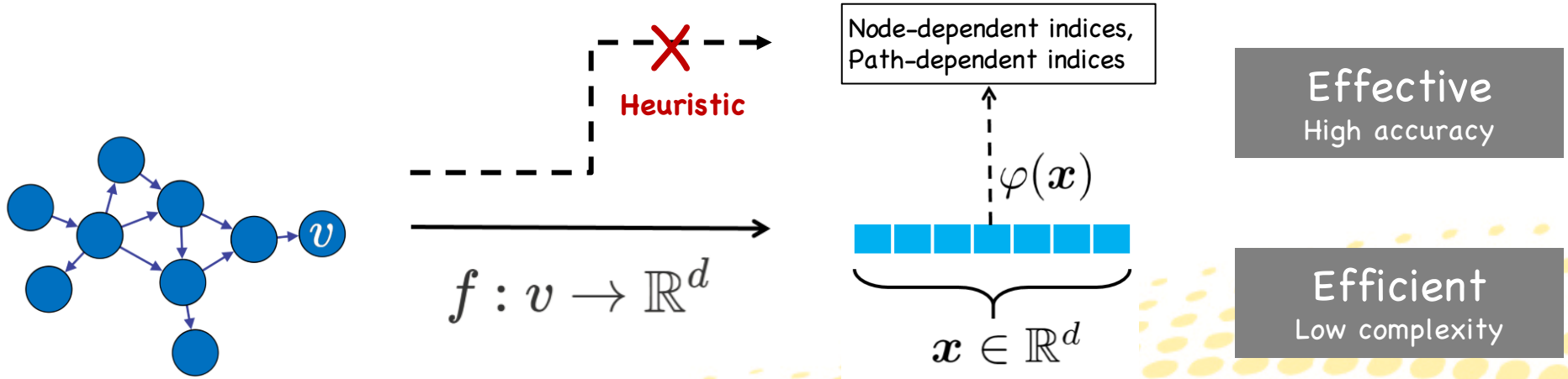
Why Network Embedding?



Feature Engineering ⇨ A task-dependent process

Computational Time ⇨ Grows rapidly as the network size increases

From Graph-based to Network Embedding-based Approach for downstream ML tasks



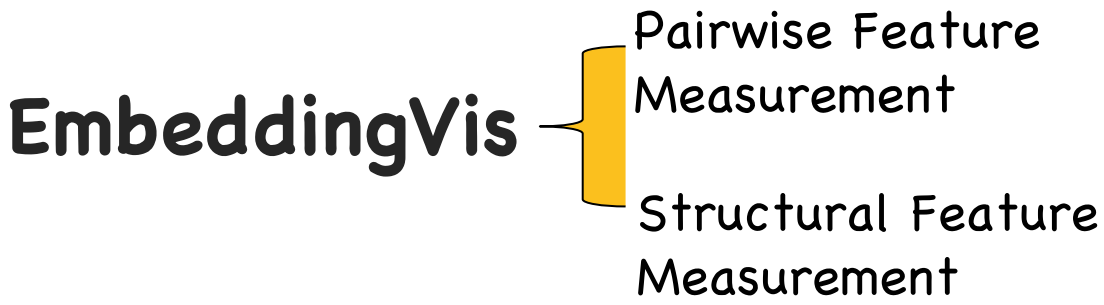
Challenges when Engaging Network Embedding

- **Understanding Obstacles**
  - No explicit meaning of basis vectors
  - Difficult to interpret and not intuitive
- **Inconvenient Comparison**
  - Stochastic nature of the construction procedure
  - Non-transparent hyper-parameters involved
- **Limited Analysis**
  - Projection to a 2D/3D space (coarse-grained analysis)
  - No universal measurement to define graph node similarity

Key Requirements from Participatory Design with WeChat

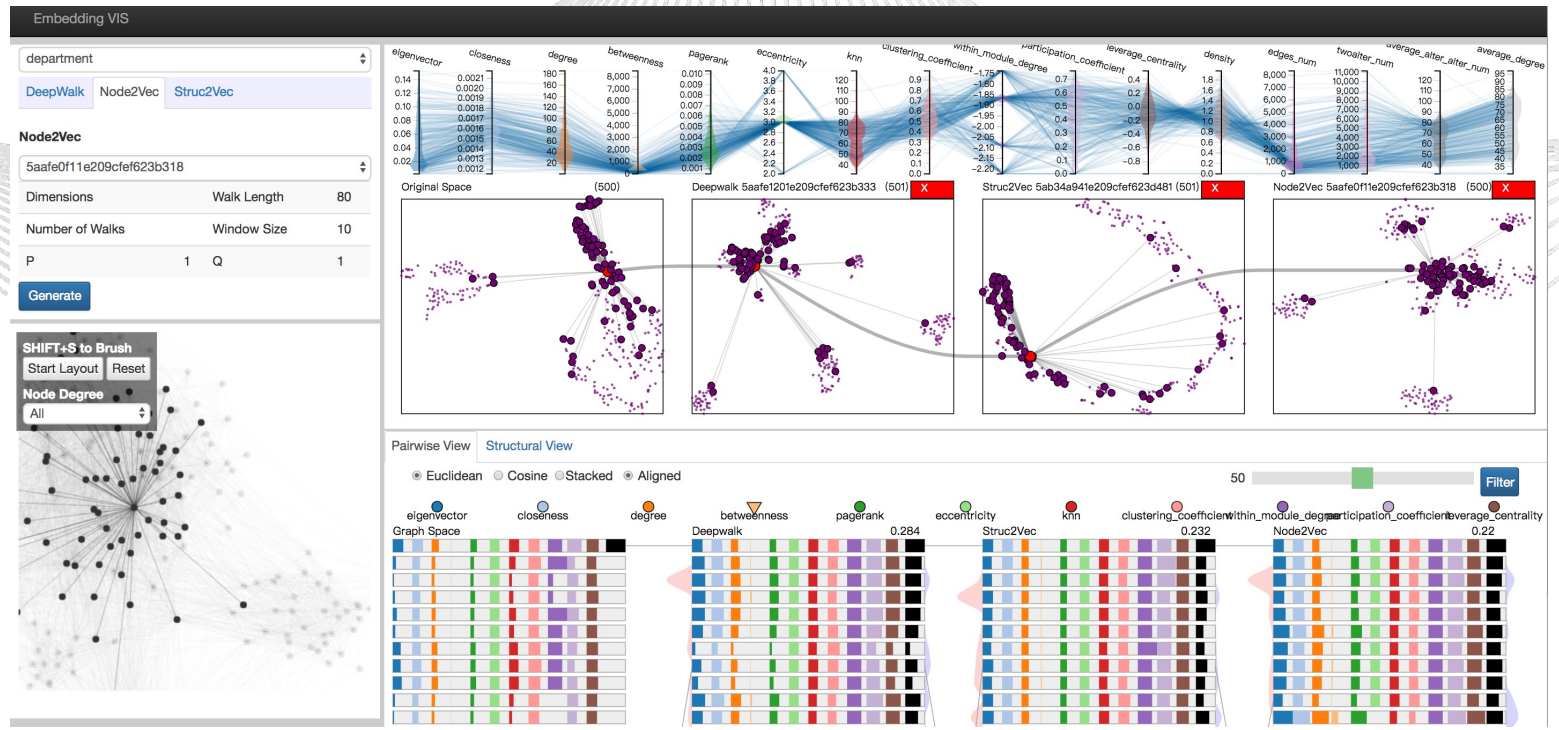
- **Pairwise Feature Measurement**
  - ➢ **Motivation**: Interpret the meaning of "neighbor" or "community" in the embedding space
  - ➢ **Method**: Identify preserved node metrics by different embeddings

- **Structural Feature Measurement**
  - ➢ **Motivation**: Embedding only makes sense when we conduct a pairwise node comparison
  - ➢ **Method**: Analyzing capability of embedding vectors to retain structural characteristics



EmbeddingVis

Pairwise Feature Measurement



Pairwise Feature Measurement

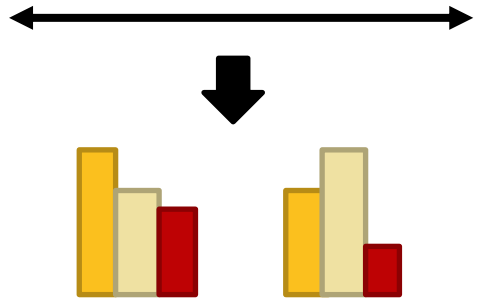
Idea

Original graph space

Similarity by pairwise **feature vector** (node metrics)

Degree
Closeness
Betweenness
…

Regression Approximation



Which and how node metrics are preserved

Embedding space

Similarity by pairwise **embedding vectors** (DeepWalk, node2vec, struc2vec, etc.)

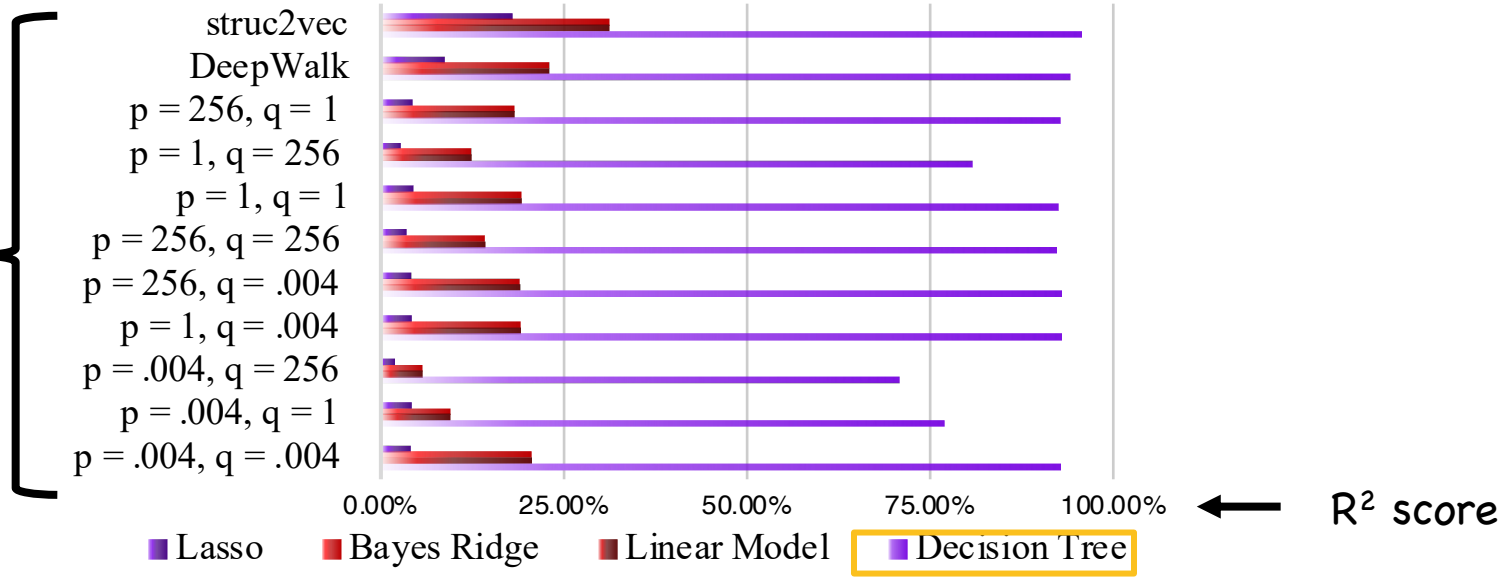
DeepWalk
Node2vec
Struc2vec
…

Target

Pairwise Feature Measurement

- Leverage **Decision Tree Regression** and extract the feature importance for each node metric

different embeddings



nine versions of node2vec

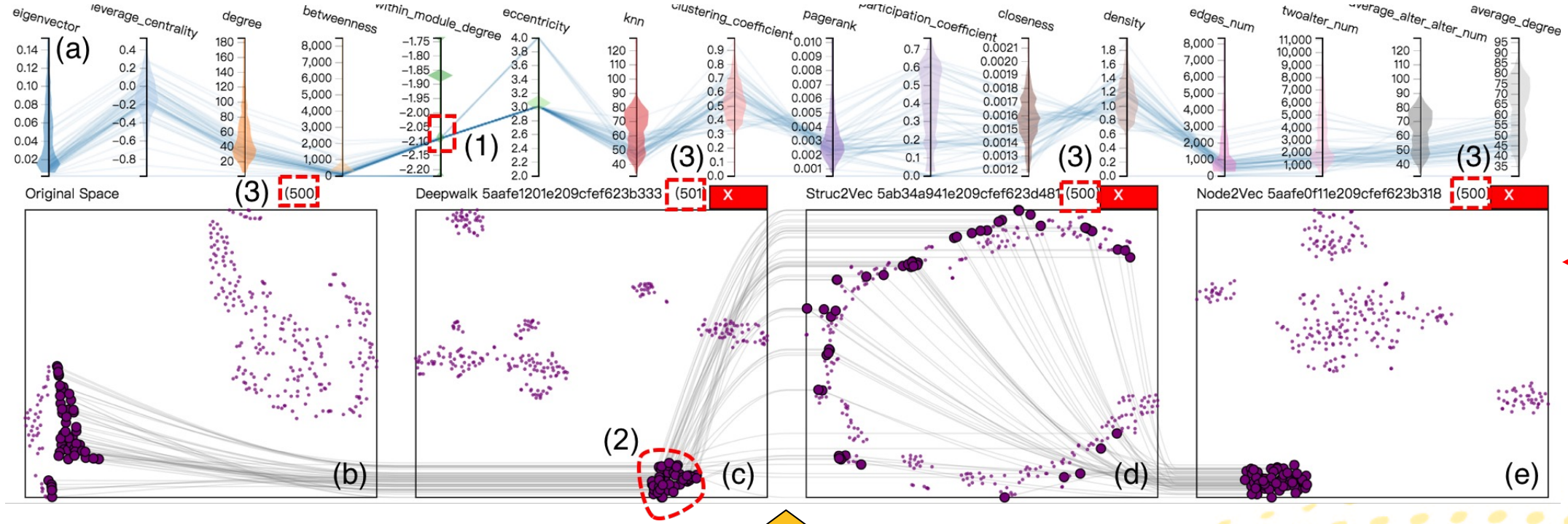
	DeepWalk	p=0.004, q=0.004	p=1, q=0.004	p=256, q=0.004	p=256, q=256	p=1, q=1	p=256, q=1	struc2vec
Degree	0.33%	0.23%	0.24%	0.30%	0.22%	0.21%	0.27%	70.98%
Betweenness	4.30%	4.10%	4.07%	3.89%	4.62%	4.03%	4.01%	5.18%
Leverage_Centrality	6.88%	6.43%	5.14%	4.84%	6.09%	6.55%	5.22%	3.00%
KNN	8.82%	7.51%	5.79%	5.53%	8.09%	7.86%	5.89%	1.69%
Closeness	9.82%	8.35%	8.00%	7.69%	8.20%	7.80%	8.16%	1.77%
PageRank	14.50%	12.89%	10.53%	10.60%	13.65%	13.15%	9.50%	11.00%
Within_Module_Degree	41.93%	50.02%	57.73%	58.82%	49.29%	49.62%	58.79%	2.03%

Rank the node metrics based on the feature importance

Cluster Transition View

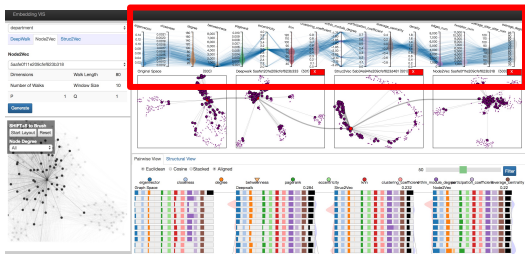
- Compare preserved node metrics across different embedding spaces

Parallel coordinate plot

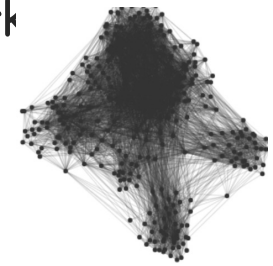


T-SNE embedded transition diagram

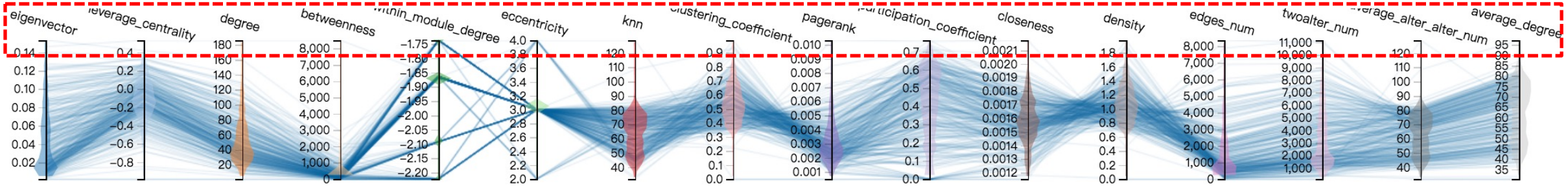
Parallel Coordinate Plot



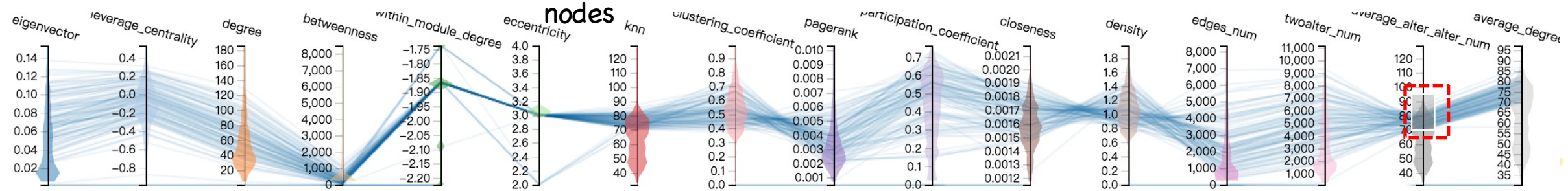
- PCP displays the overview of metric distribution of the network nodes



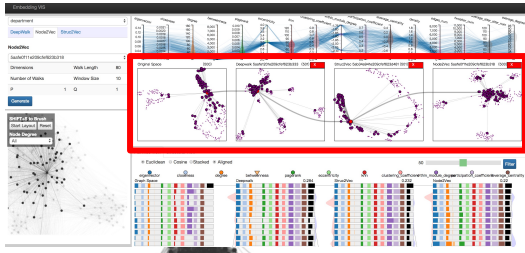
Node metrics

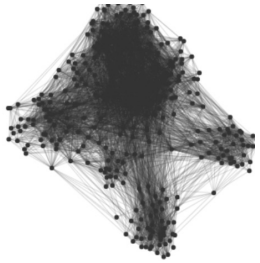


Users can brush on axes to filter nodes

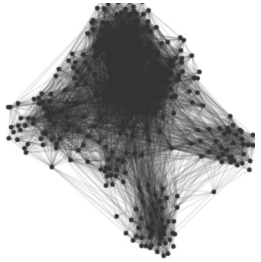


T-SNE Embedded Transition Diagram

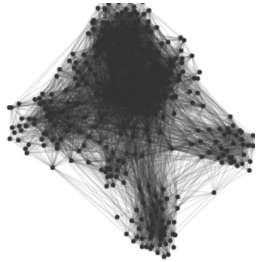




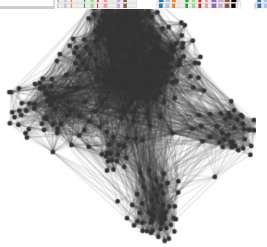
feature vector (node metrics)



DeepWalk embedding vector

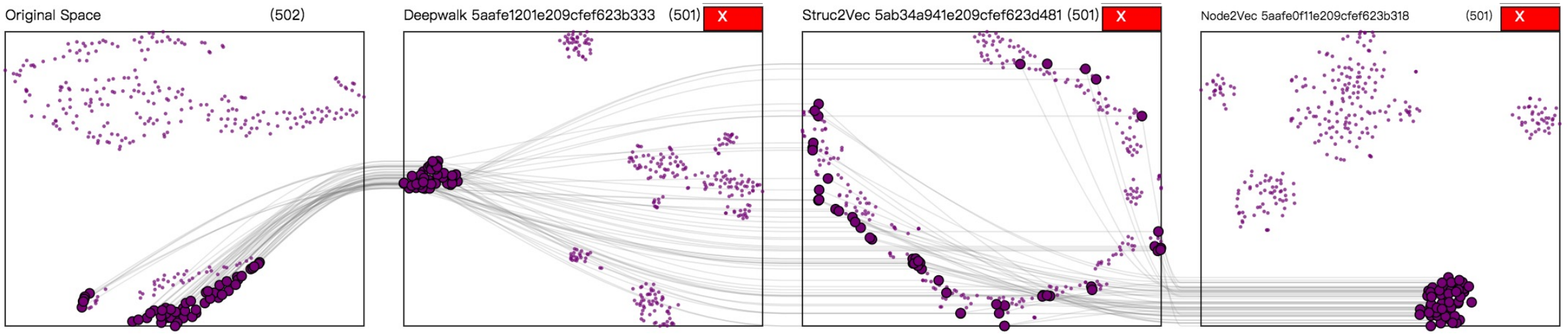


struc2vec embedding vector



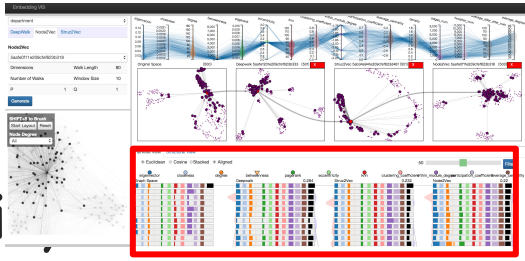
node2vec embedding vector

T-sne Projection

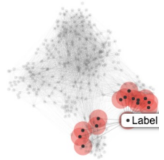


graph space | DeepWalk space | struc2vec space | node2vec space

Pairwise Ranking View



- Explore the neighboring information help assess the output quali
  embedding



Graph space

Deepwalk space

struc2vec space

node2vec space

NDCG: 0.854

NDCG: 0.278

NDCG: 0.908

Select a node

eigenvector | leverage_centrality | degree | betweenness | within_module_degree | eccentricity | knn | clustering_coefficient | pagerank | participation_coefficient | closeness

Graph Space | Deepwalk | 0.854 | Struc2Vec | 0.278 | Node2Vec | 0.908

Neighbors of the node

Neighboring list

Sorted by Euclidean distance to the selected node

Ranking by **feature vector**

Ranking by **DeepWalk embedding vector**

Ranking by **struc2vec embedding vector**

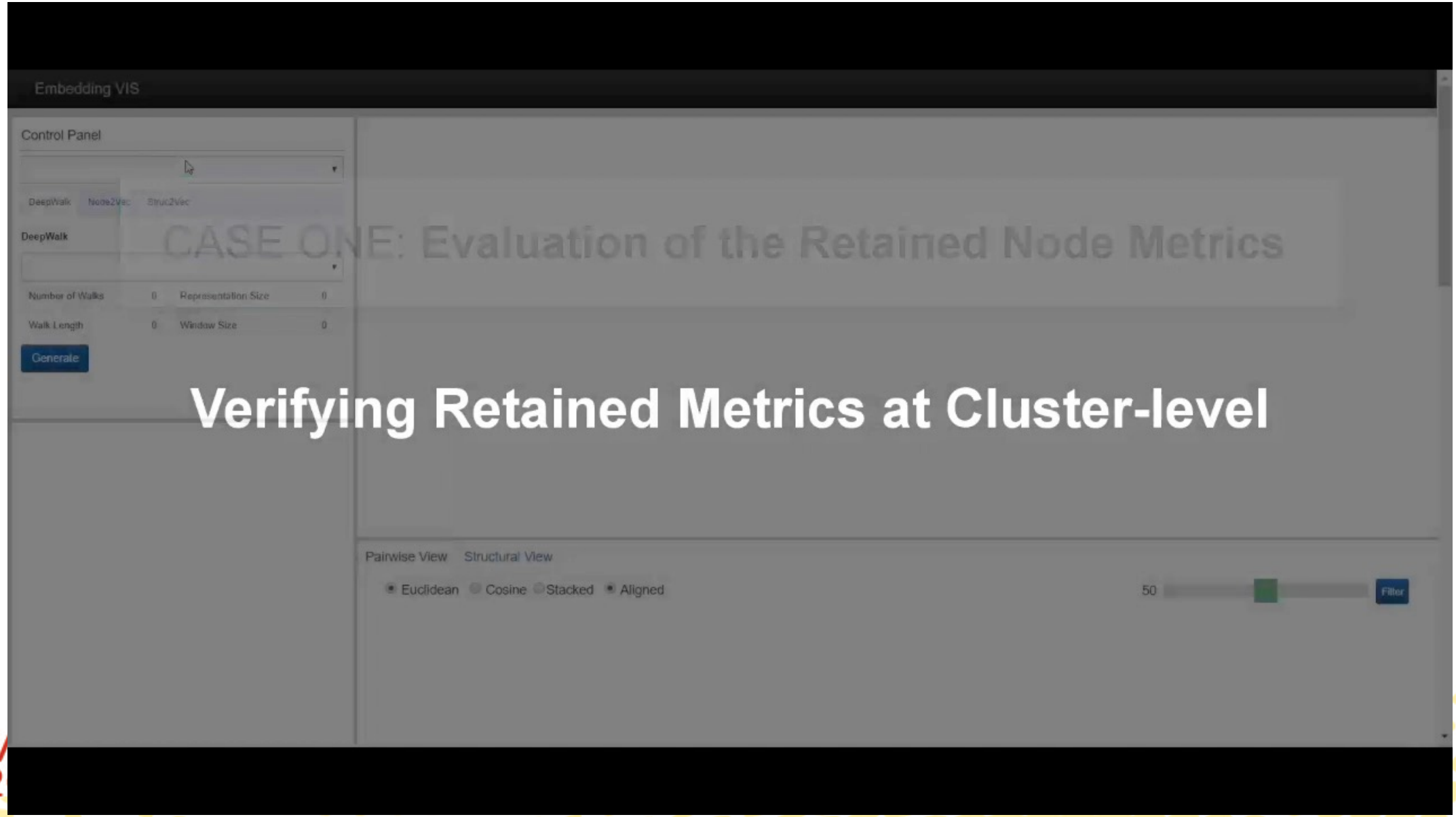
Ranking by **node2vec embedding vector**

How many embedding spaces contains the corresponding node

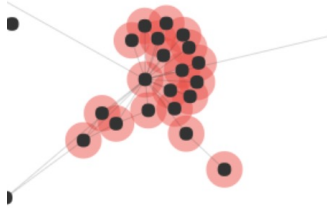
Number of common friends between the neighbor and the selected node

NDCG measures ranking quality of different embeddings comparing with ranking in graph space

Case 1: Verifying Preserved Node Metrics at Cluster-level

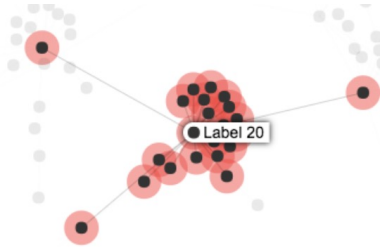


Case 1: Verifying Preserved Node Metrics at Instance-level



Step 1:
Lasso a cluster of nodes

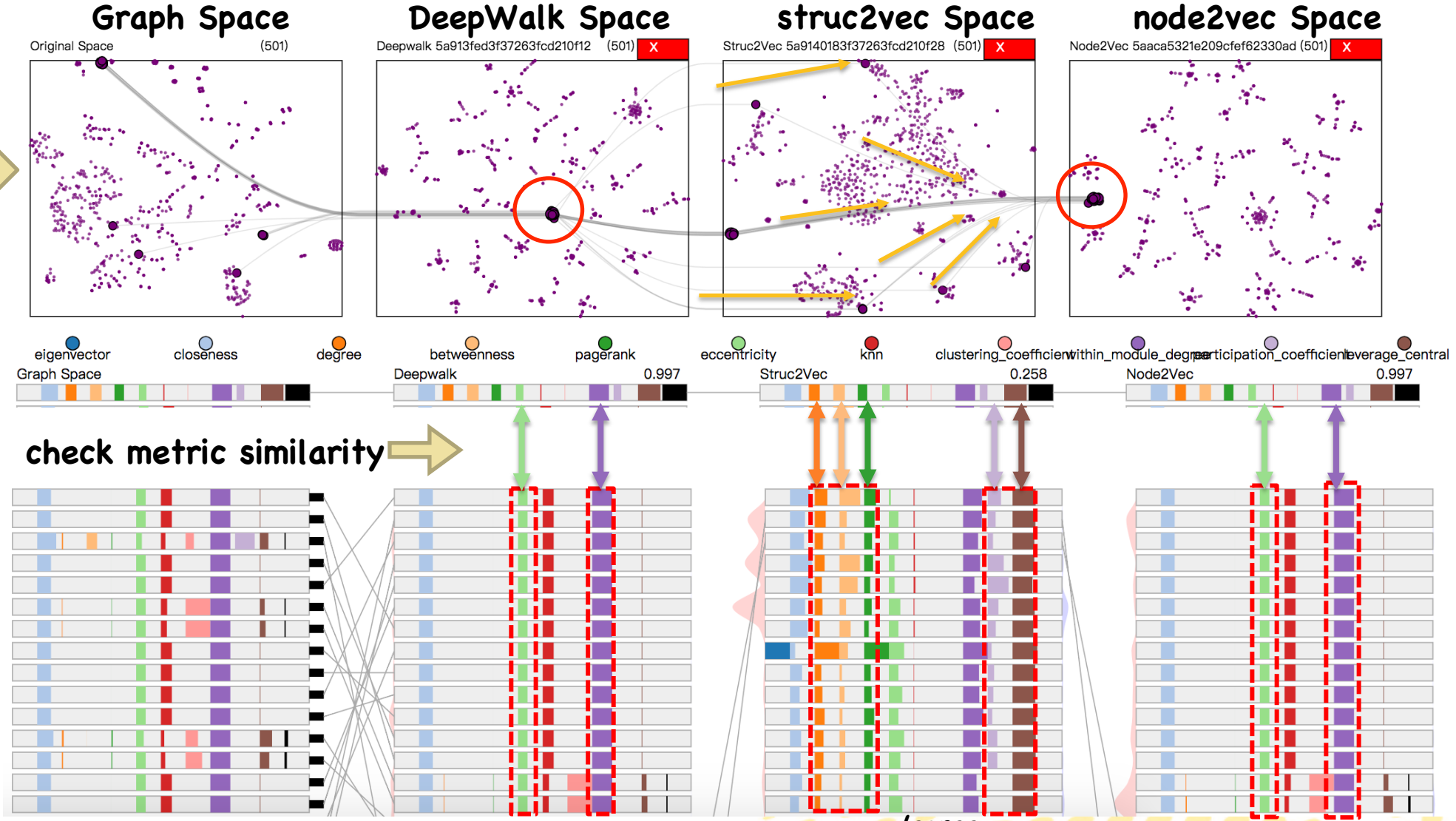
the selected node



Step 2:
Click a hub node with label 20

neighbors

check metric similarity

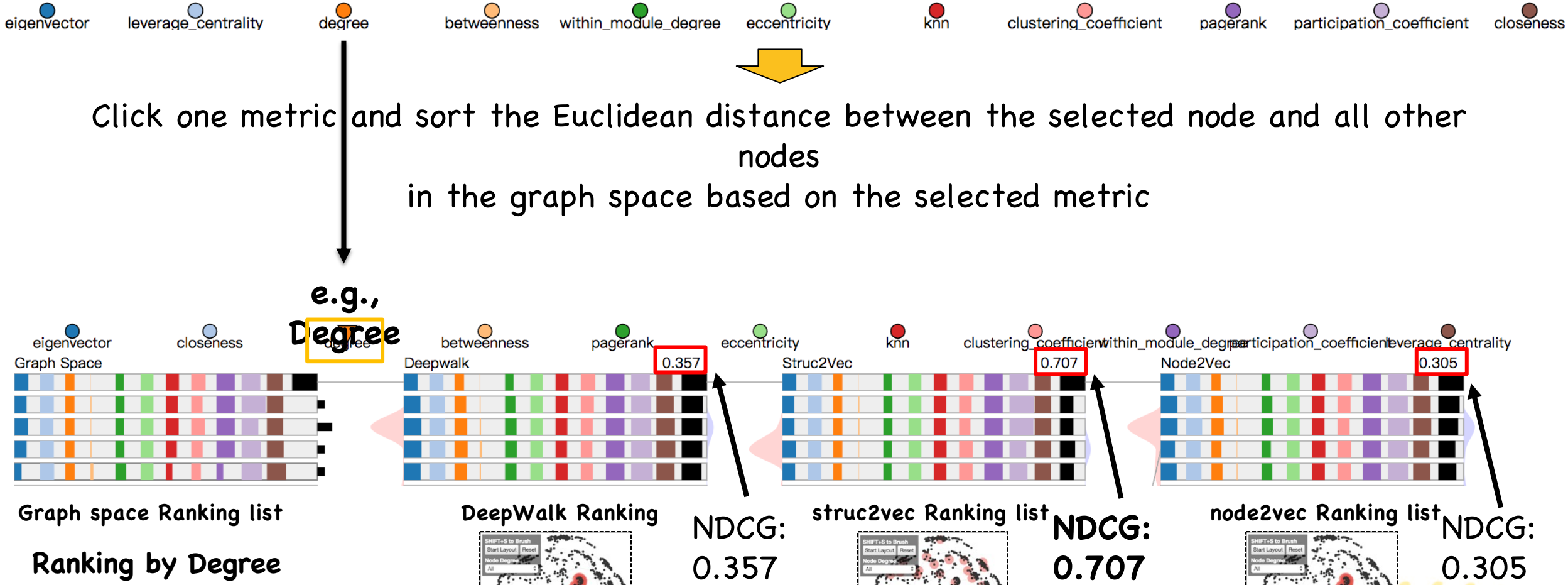


eccentricity
within module degree

degree
leverage centrality
pageRank…

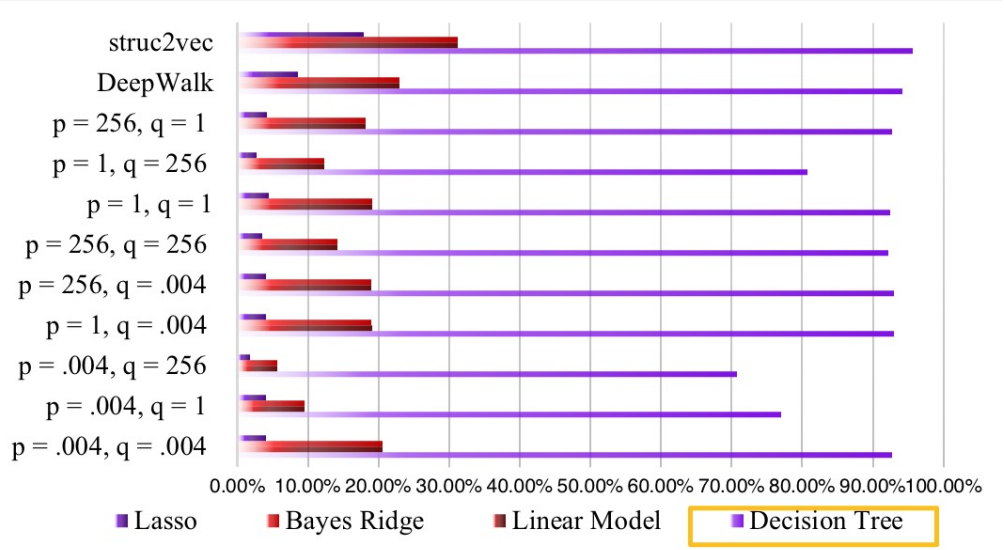
eccentricity
within module degree

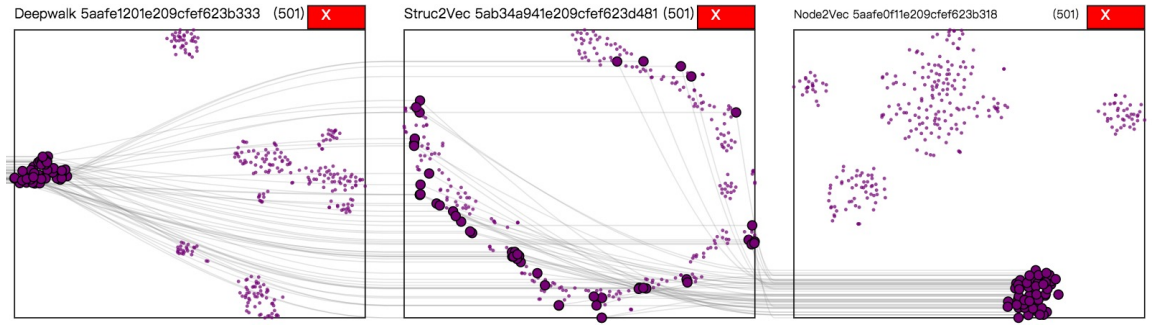
Case 1: Verifying Preserved Node Metrics by NDCG



Takeaway Messages - Pairwise Feature Measurement

- Embedding preserves node metrics in a non-linear way
- Different embeddings preserve different node metrics





DeepWalk: "bound to local links"

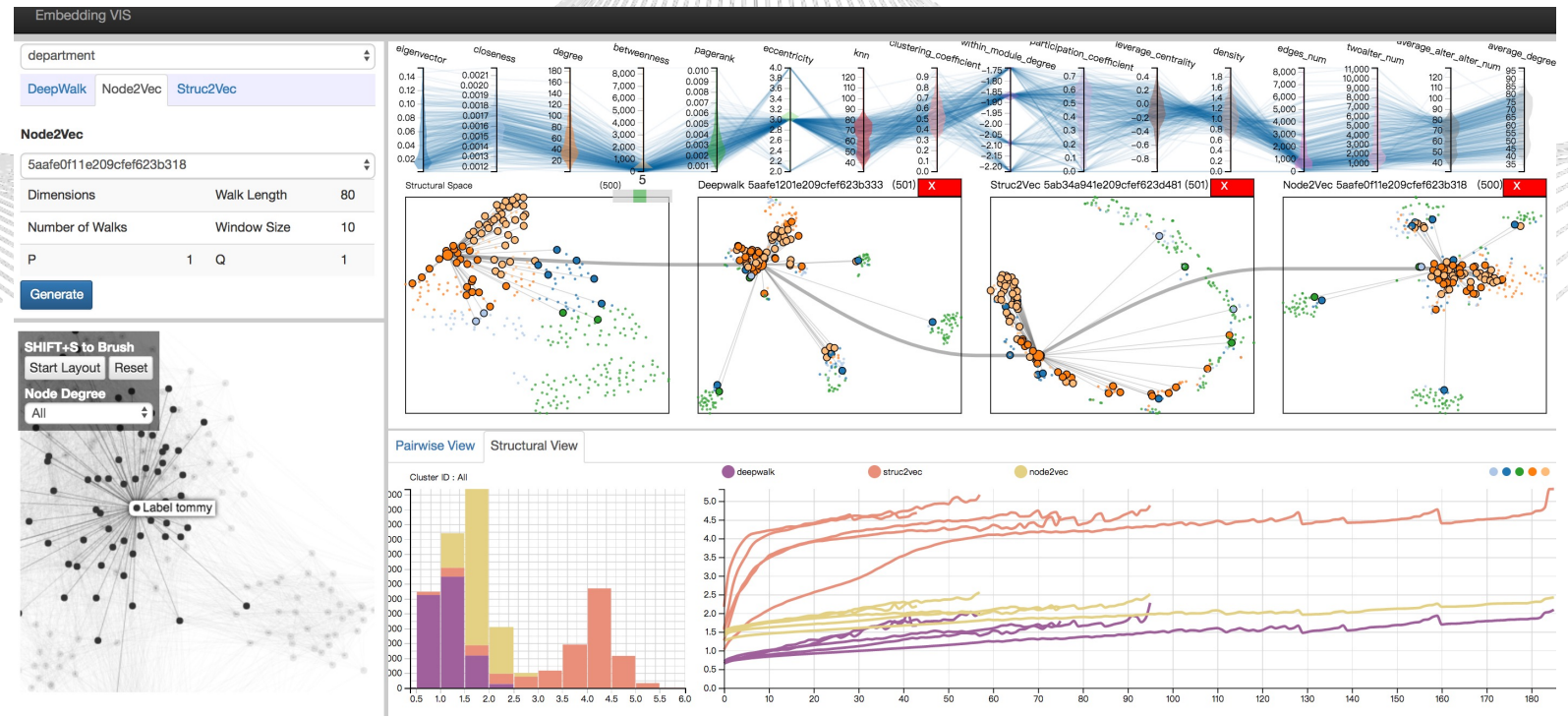
struc2vec: "hub" nodes

node2vec: "bound to local links"

(e.g., with similar degree)

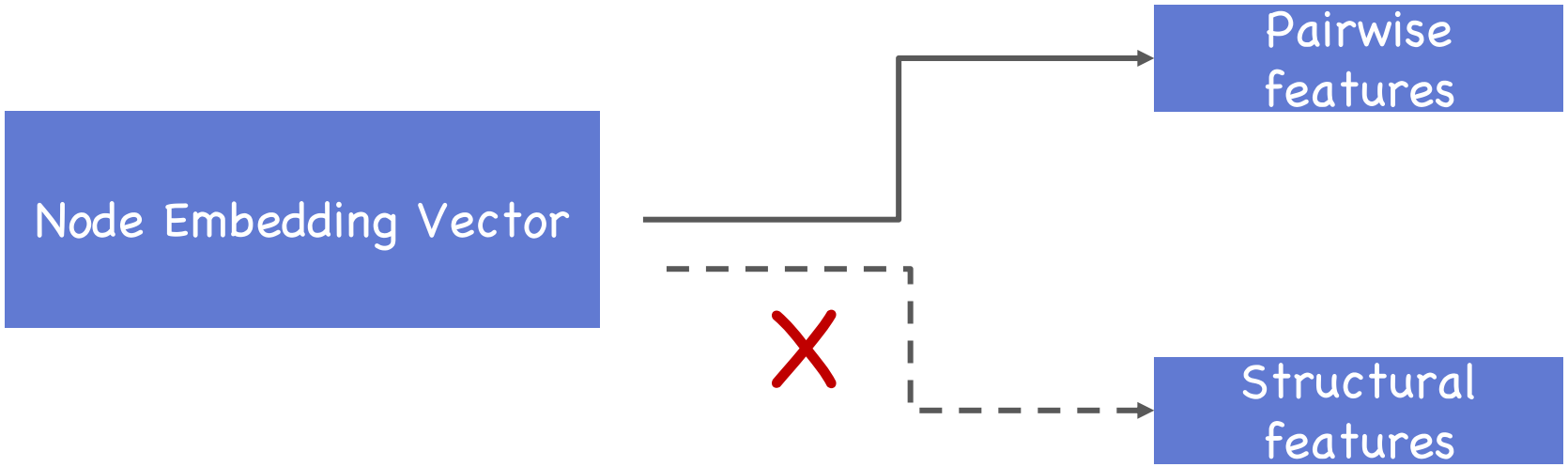
EmbeddingVis

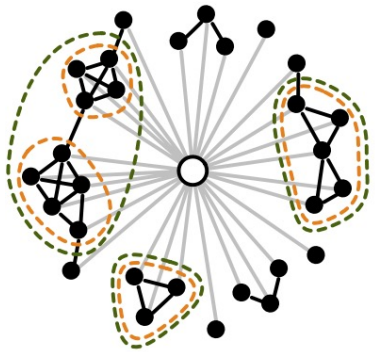
Structural Feature Measurement



Structural Feature Measurement

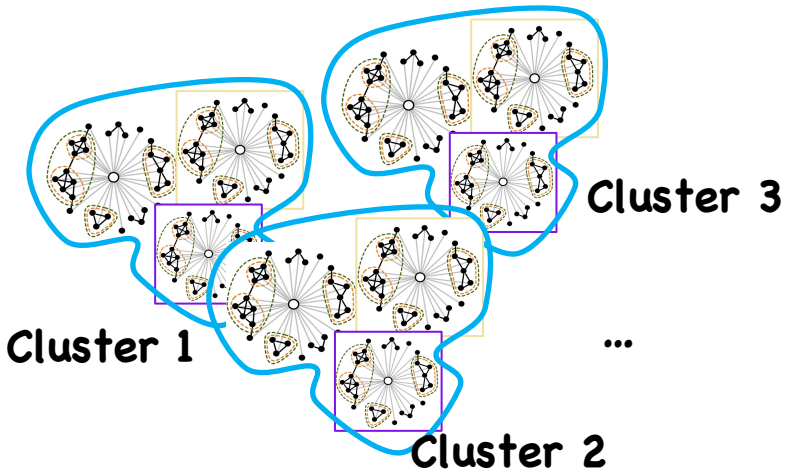
- Embedding only makes sense when comparing one embedded vector with another
- A single vector alone does not provide any structural information of the graph





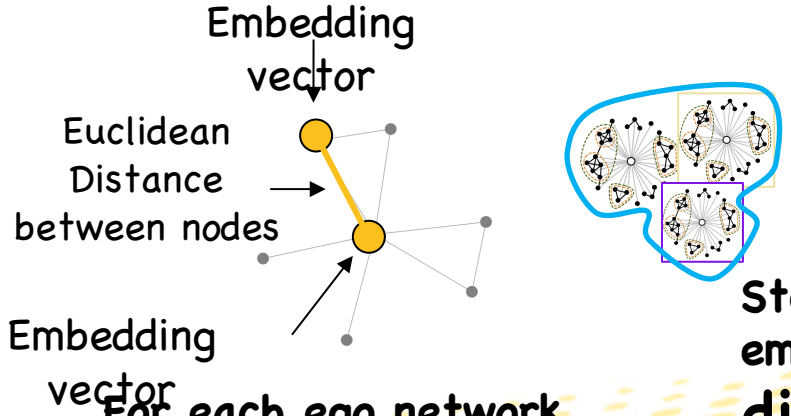
Using the *ego-network* structure

Structural Feature Measurement

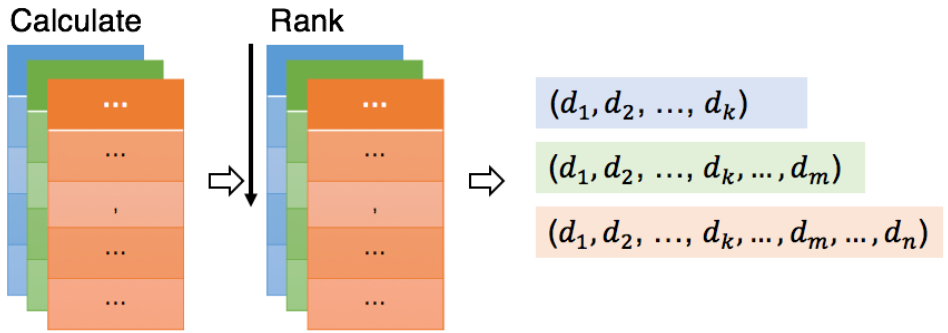


Step 1: Cluster the ego network structures based on selected metrics

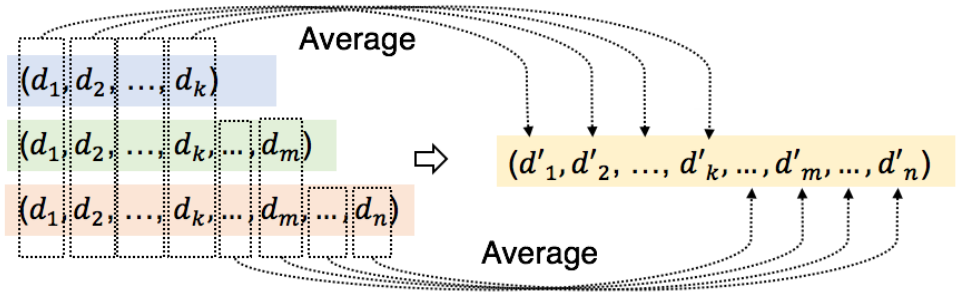
Degree
Density
Edge number
Number of alters
Average degree
Clustering coefficient
...



For each ego network



Step 2: For each cluster, rank distance based on the embedding vectors of the link nodes to create distance vectors

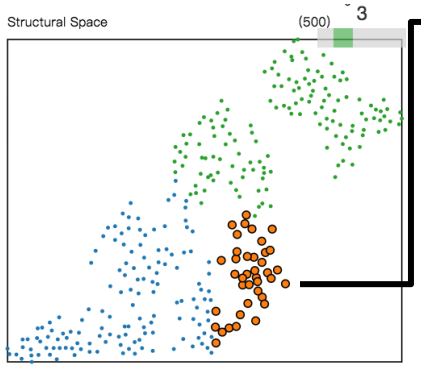


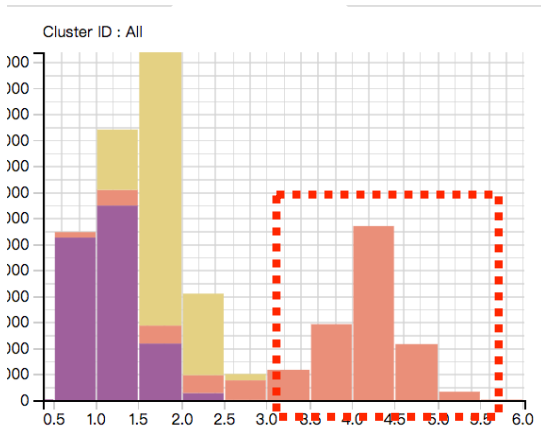
Step 3: Obtain "average distance vector" to represent the structure feature for this cluster

Structural View

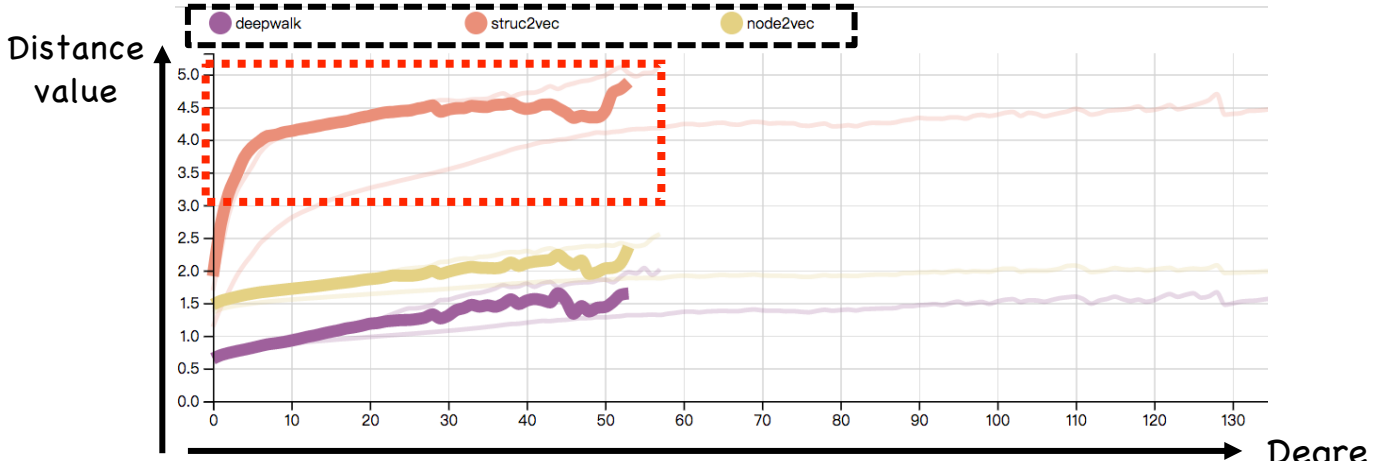
- Compare **distance distribution** of ego network structure clusters across different embeddings

Step 1: Cluster the nodes based on its "ego-network" structures





Step 2: Run three embedding models to get three average distance vectors for one cluster

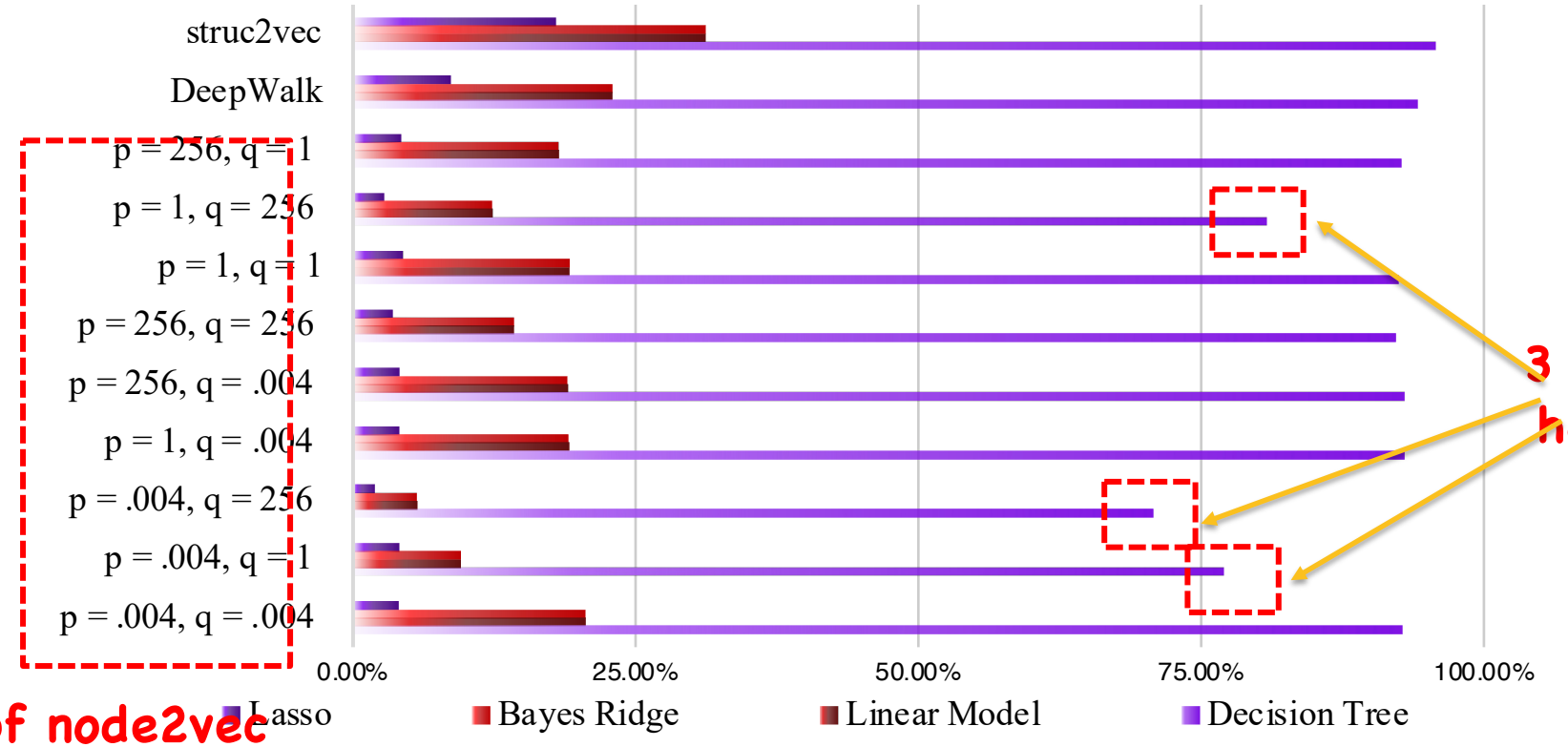


Distance distribution based on embedding vectors from each embedding model

X axis: dimension index of "average distance vector"
Y axis: the value at the corresponding dimension

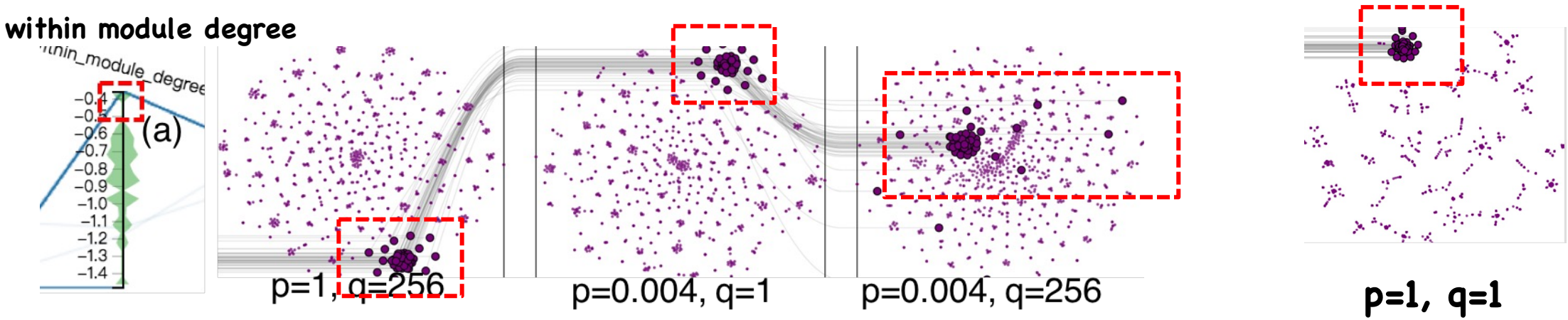
Case 2: Understanding Hyper-Parameters (p and q in node2vec)

- As claimed in **node2vec**, **p** and **q** control the searching strategy of random walk



Case 2: Understanding Hyper-Parameters - p

- When p is small, the local neighboring nodes could not be preserved well



node2vec version 1

node2vec version 2

node2vec version 3

node2vec baseline

Performance: baseline > version 1 and 2 > version 3

Case 2: Understanding Hyper-Parameters - q

We find

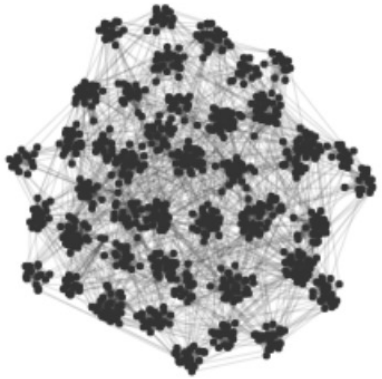
no significant differences across

(p = 256, q = 0.004), (p = 256, q = 1), and **(p = 256, q = 256)**

WHY

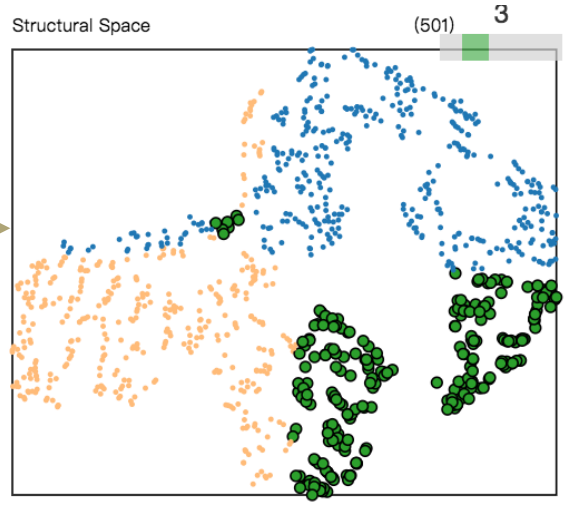
Use a simpler synthetic graph to study q

Case 2: Use a Simple Graph to Understand Hyper-Parameters - q



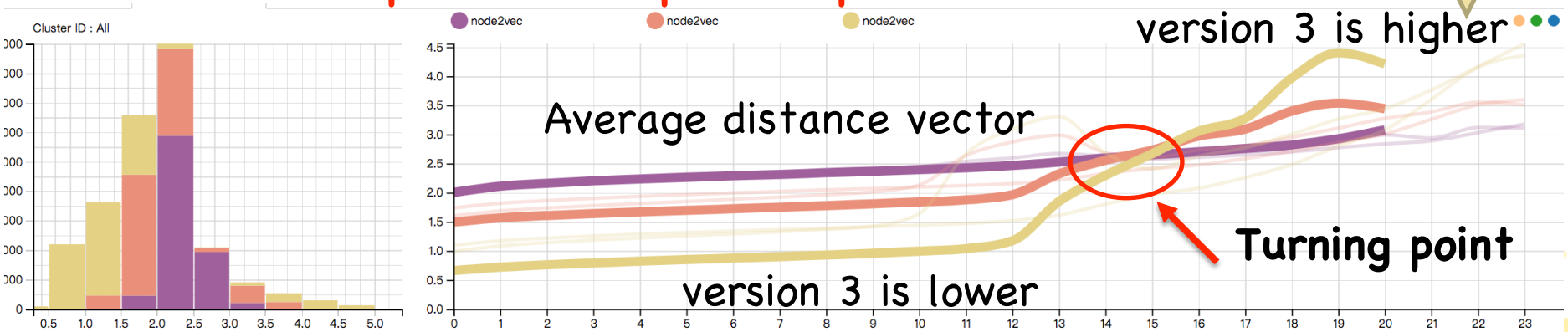
a simple synthetic network

3 clusters
based on its **ego-network structures**

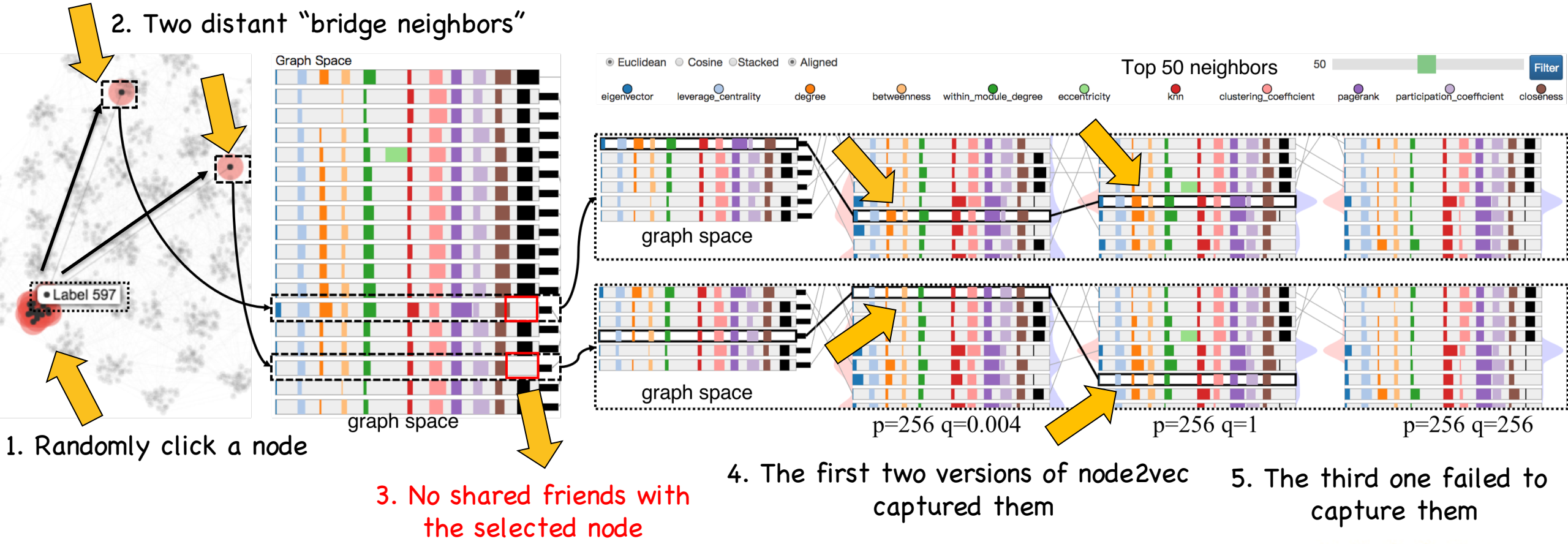


cluster 3

node2vec version 1	node2vec version 2	node2vec version 3
p = 256,	**p = 256**	**p = 256**
q = 0.004	**q = 1**	**q = 256**



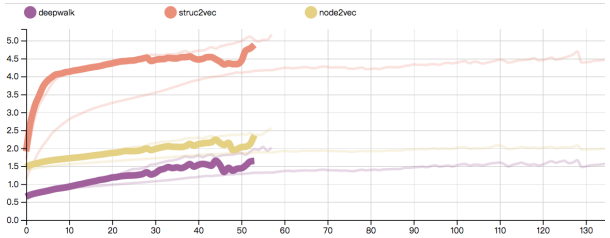
Case 2: Understanding Hyper-Parameters - q



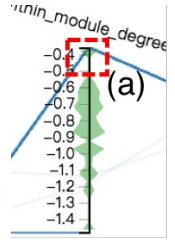
Since the two 'bridge nodes' have no links to any other nodes in the cluster, it has a great chance to miss them when **q is high**

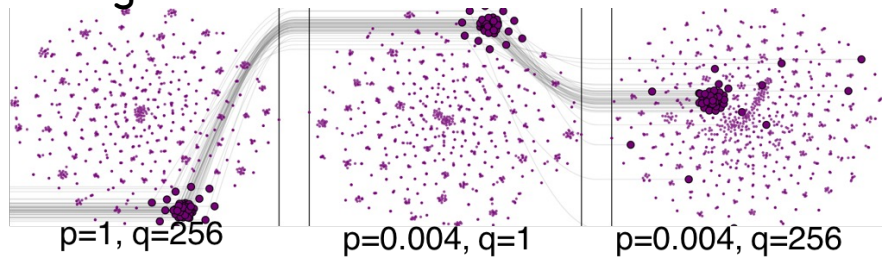
Takeaway Messages - Structural Feature Measurement

- Practical to use embedding vectors to describe structural information



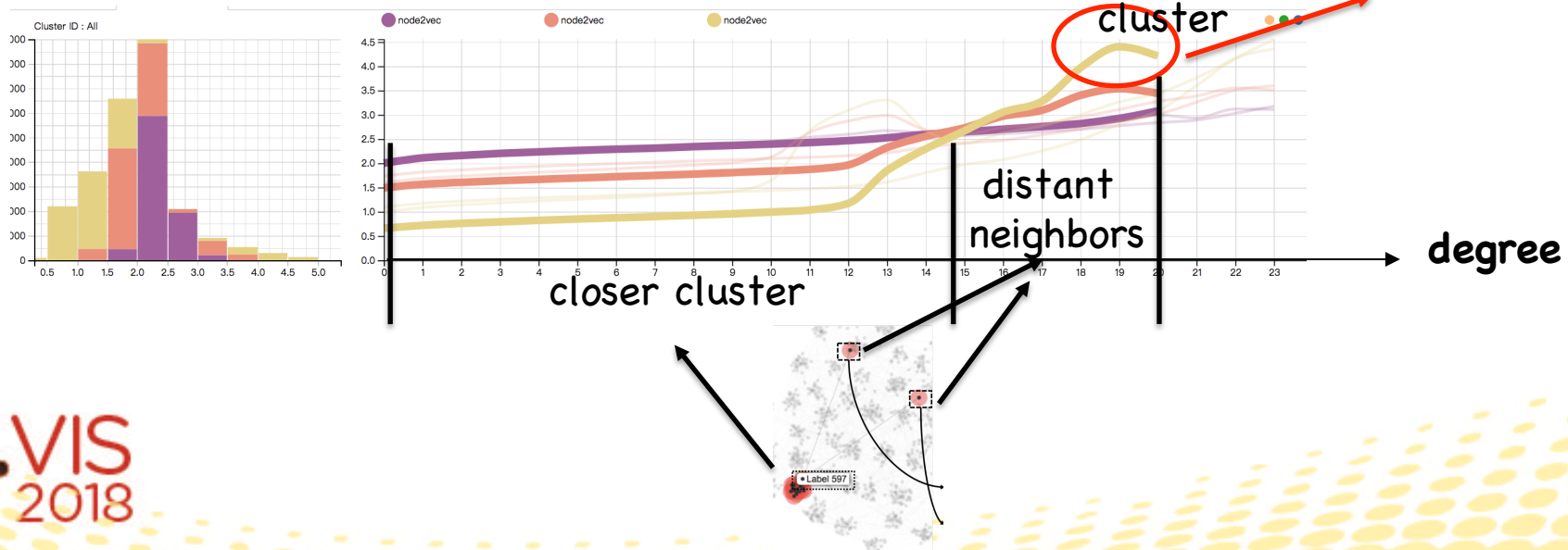
Trend is similar across different embeddings

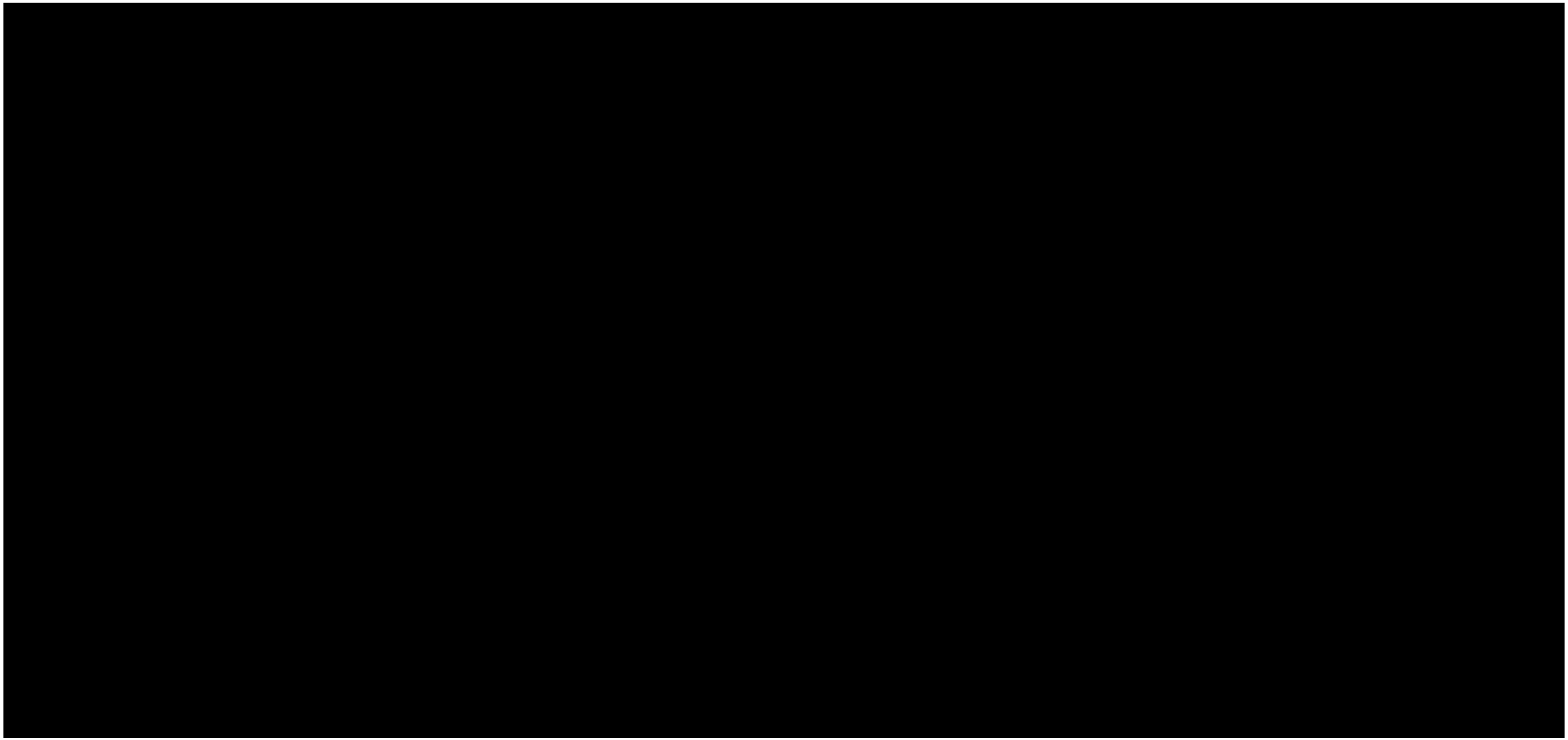




- A small p: cannot preserve local neighboring nodes well
- A high q keeps the ego network structure much close and may ignore those neighbors with no other links to the community

p = 256, q = 0.004 | p = 256 q = 1 | p = 256 q = 256





Visualization for Model Development

- Comments
  - Scalability
    - Most only tested for small datasets like MNIST
  - How to evaluate understanding?
    - Most use expert reviews
  - Is it possible to qualitatively evaluate fairness (non-discrimination) and robustness of classifiers?

What are the problems?

- Vis for Exploratory Data Analysis
  - ~~What does my dataset look like? Any mislabels?~~
- Vis for Model Development
  - ~~Architecture: What is the classifier? How to compute?~~
  - ~~Training: How the model gradually improves? How to diagnose?~~
  - ~~Evaluation: What has the model learned from the data?~~
  - Comparison: Which classifier should I choose?
- Vis for Operation
  - Deploy: How to establish users' trust?
  - Operation: How to identify possible failure?

Visualization for Operation

- Deploy: How to establish users' trust?
  - If users don't trust the model, they will not use it! (Lieberman 1998)
  - Trust is based on experience
  - Interaction boost trust (Stumpf 2007)

- Operation: How to cope with possible failure?
  - Human taking over in case of failure
  - Identify failure for safety-critical applications
  - Better user experience

Few studies in this part

VIS 2021

上海科技大学 ShanghaiTech University

Inspecting the Running Process of Horizontal Federated Learning via Visual Analytics

IEEE Transactions on Visualization and Computer Graphics (TVCG 2021)

Quan Li
Assistant Professor
ShanghaiTech University

Xiguang Wei
Senior Researcher
Semacare

Huanbin Lin
Researcher
WeBank AI

Yang Liu
Principle Researcher
WeBank AI

Tianjian Chen
Deputy General Manager
WeBank AI

Xiaojuan Ma
Associate Professor
HKUST

上海科技大学 ShanghaiTech University

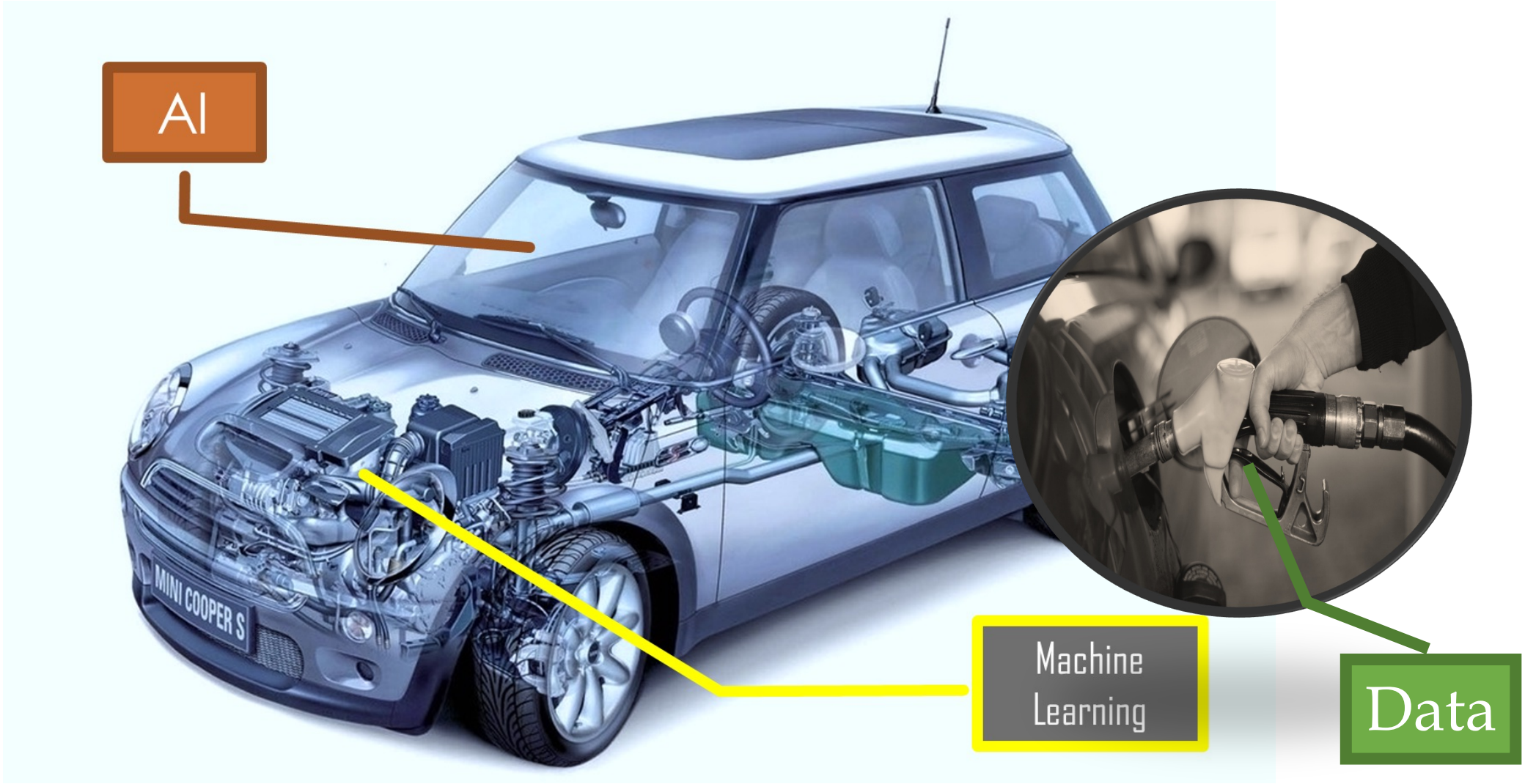
SEMACARE

WeBank 微众银行

香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

立志 成才 报国 裕民

Data, Machine Learning and AI ← Ideal Picture



Limits of Traditional Ways

Traditional Ways

Buying Data - ILLEGAL

Directly buying data from 3rd party companies is getting banned around the world and violates privacy.

Using Desensitization Data - INEFFECTIVE

Getting and using desensitization data between corporations cannot provide any guarantee of the outcome and performance of modeling.

Combining Results - RISKY

Using results from models individually from different data sources: Companies take their own risks to the results.

Current Challenges

Ways Blocked Between Collaborators

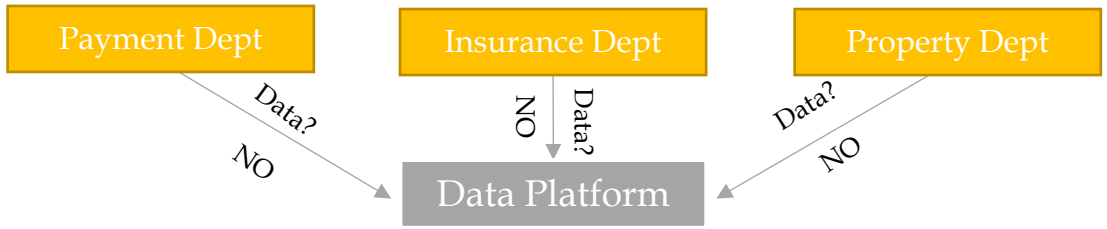
Companies cannot buy data directly under more restricted laws.
Further audit and privacy concerns make companies unwilling to collaborate.

Financial Data include credit reports, transaction history and fraud detection, etc.

User Data include user portrait, activity history, interest labels, and consumption habits, etc.

Unwillingness of Data Sharing within Departments/Subsidiaries

Parent company finds it hard to build a universal data platform.



Challenges to AI: Data Privacy and Confidentiality

French regulator fines Google $57 million for GDPR violations

Share on Facebook Share on Twitter +

Google hasn't transparently implemented GDPR rules, French regulator claims.

1 . France's National Data Protection Commission (CNIL) found that Google provided information to users in a non-transparent way.

"The relevant information is accessible after several steps only, implying sometimes up to 5 or 6 actions"
- CNIL said.

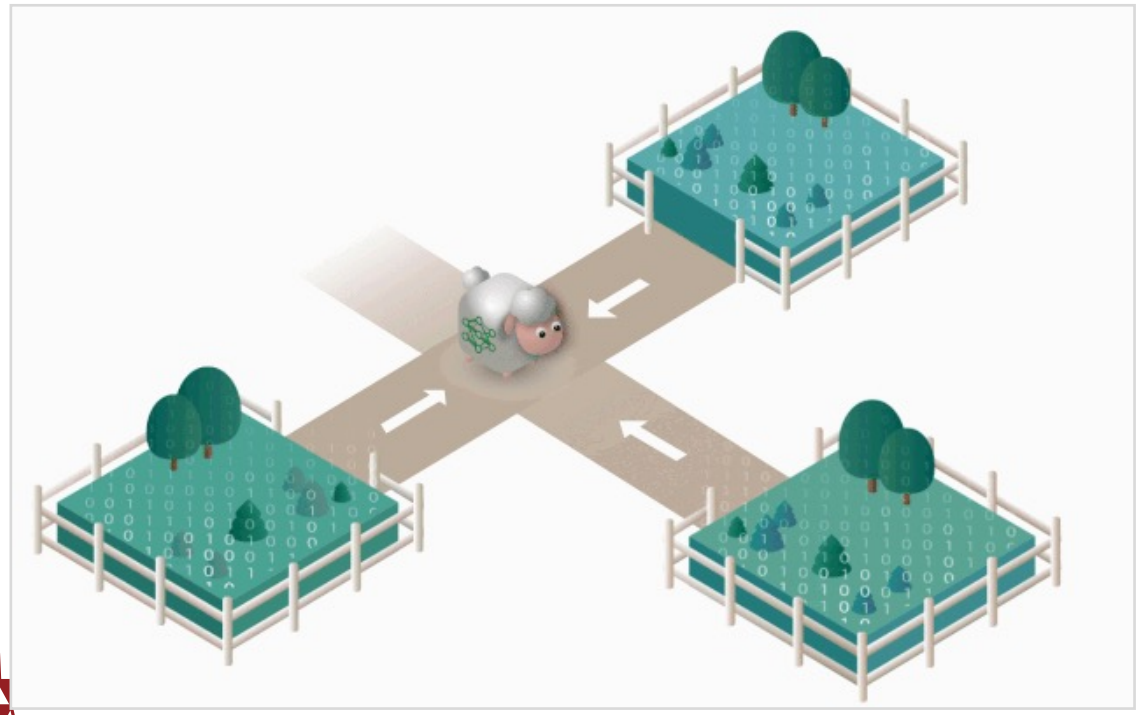
2. The users' consent, CNIL claims, "is not sufficiently informed," and it's "neither 'specific' nor 'unambiguous'."

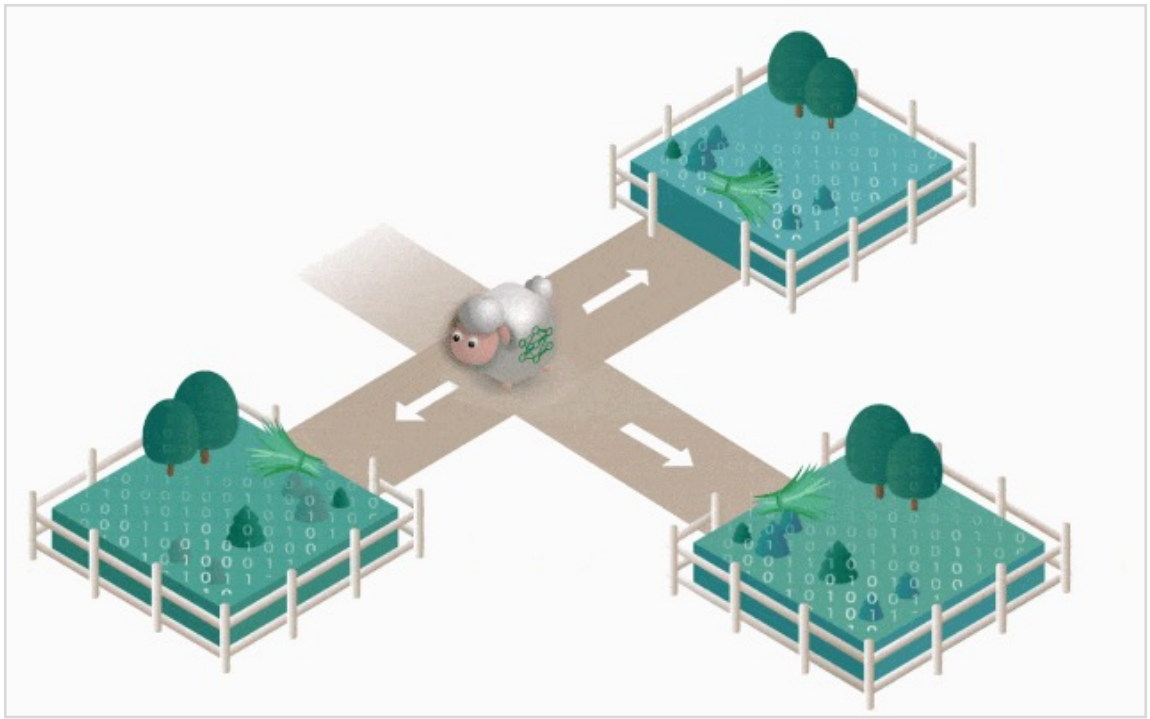
To date, this is the largest fine issued against a company since GDPR came into effect last year.

Federated Learning

Suffocation of Data Collaboration limits the Effectiveness of ML

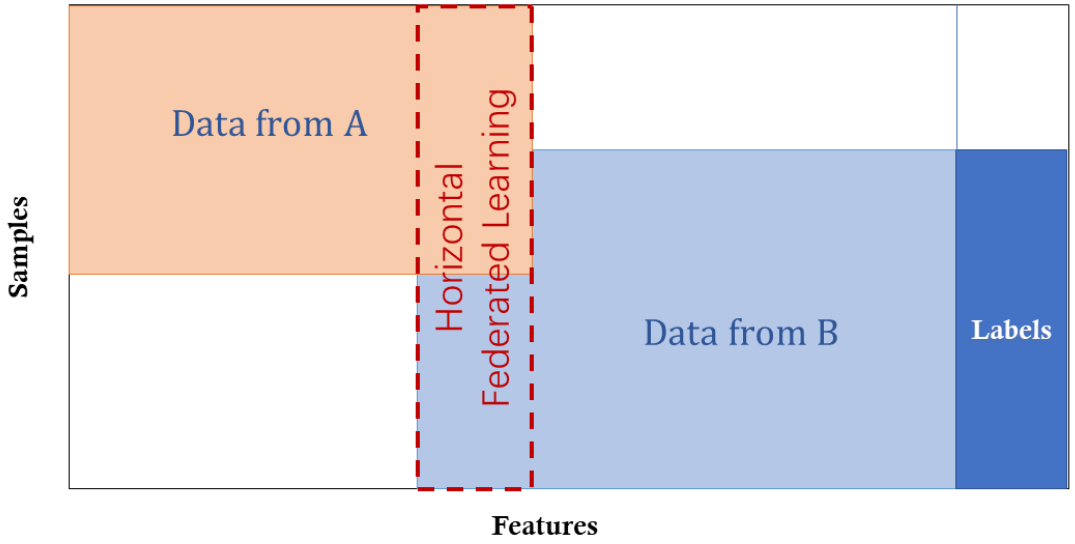
Utilize sufficient data from different sources without sharing sensitive personal data





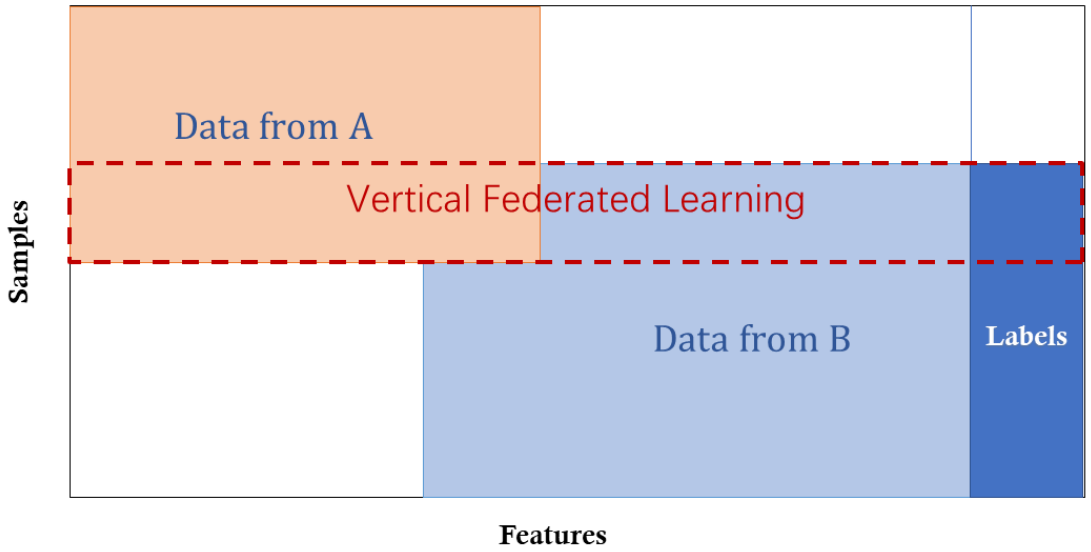
Categorization of Federated Learning

Horizontal FL



- Large overlap of features of the two data sets

Vertical FL



- Large overlap of sample IDs (users) of the two data sets

Federated Learning

Challenges when Promoting HFL in Applications

Limited Information for Inspection

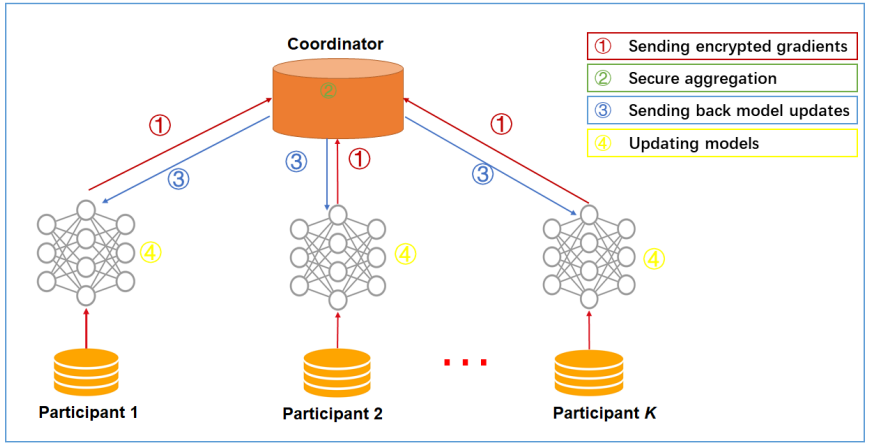
HFL server has neither access to the clients' data nor full control of the clients' behaviors due to its data privacy mechanism design

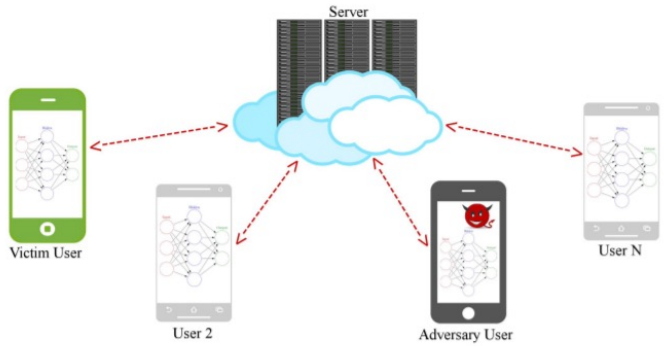
Shallow-level Analysis

Existing tools do not support fine-grained analysis such as potential anomaly detection and contribution assessment

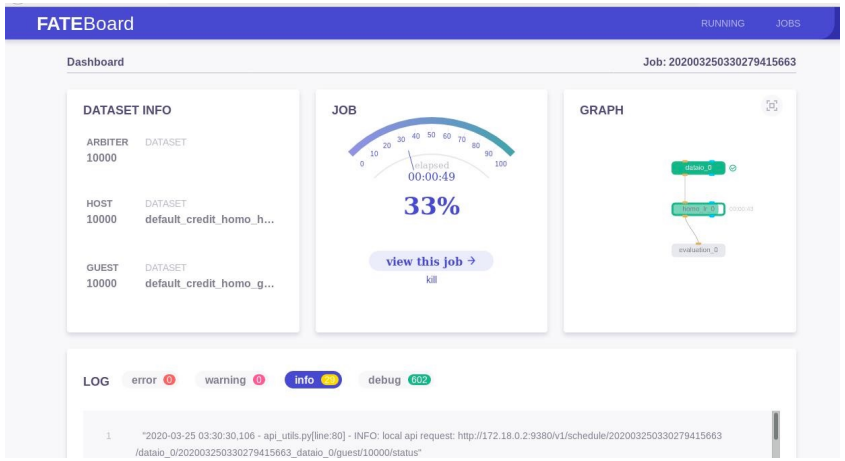
Inefficient Intervention

HFL maintainers have trouble making swift and informed decisions based only on simple logs and metrics

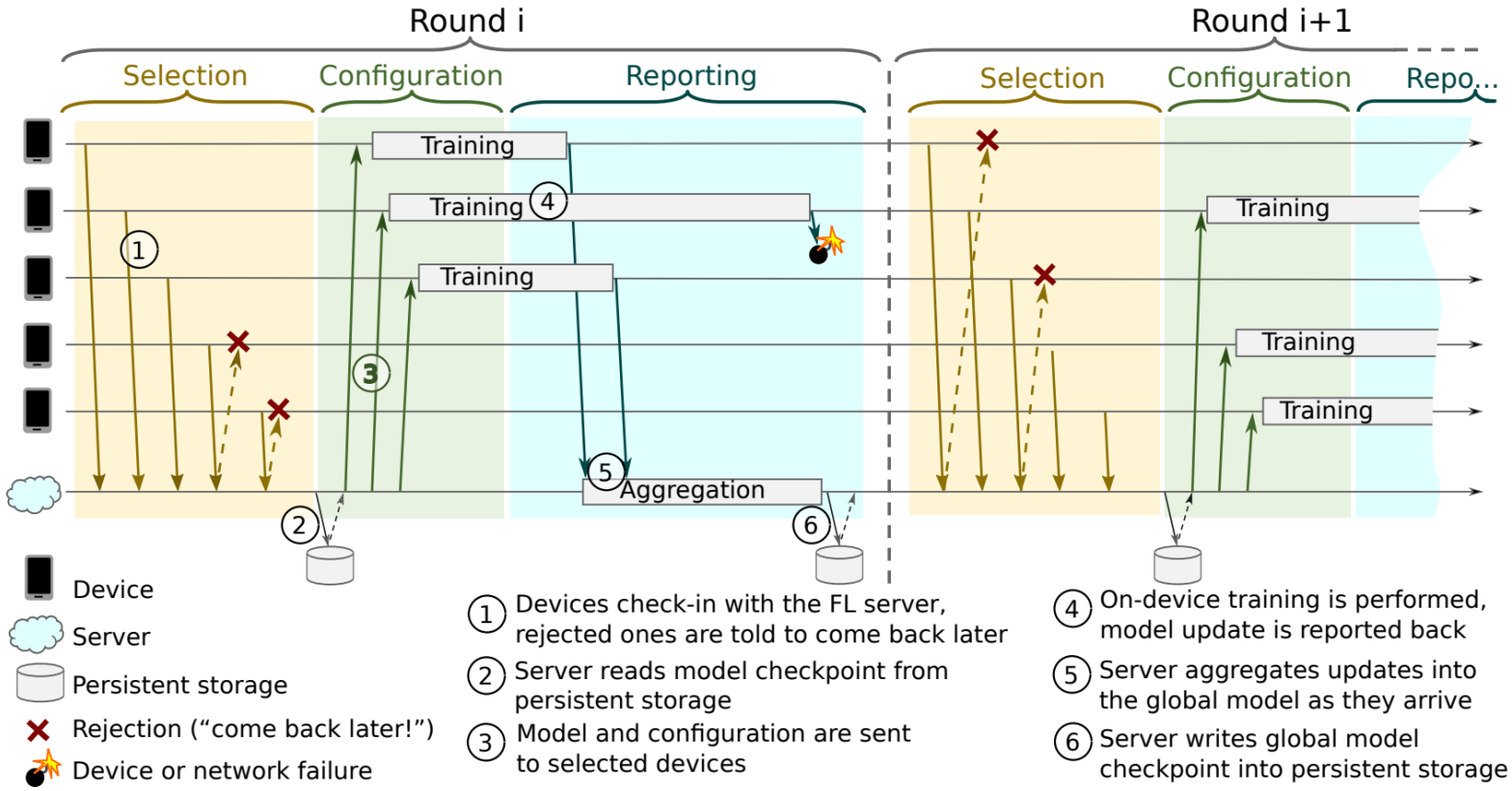


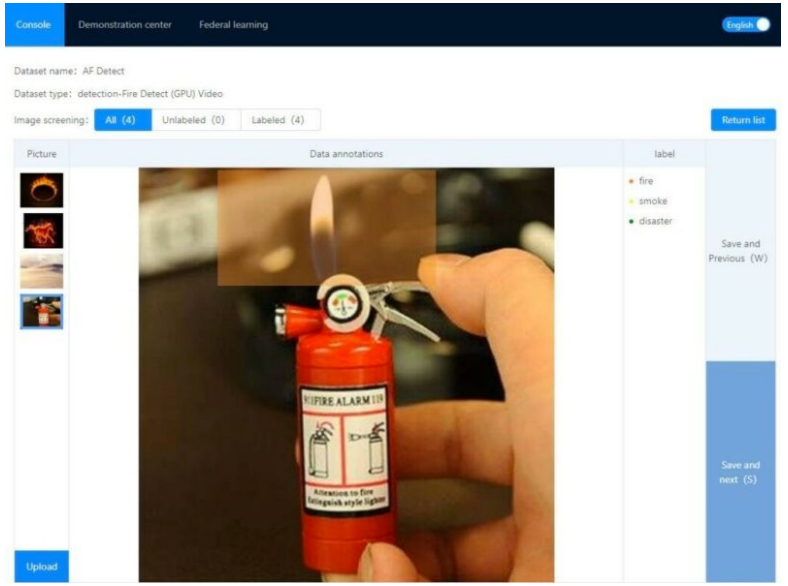


(b) Collaborative Learning



Observational Study and Requirements





Literature Survey

R.1 Understand the HFL process

R.2 Correlate client information in one/different round(s)

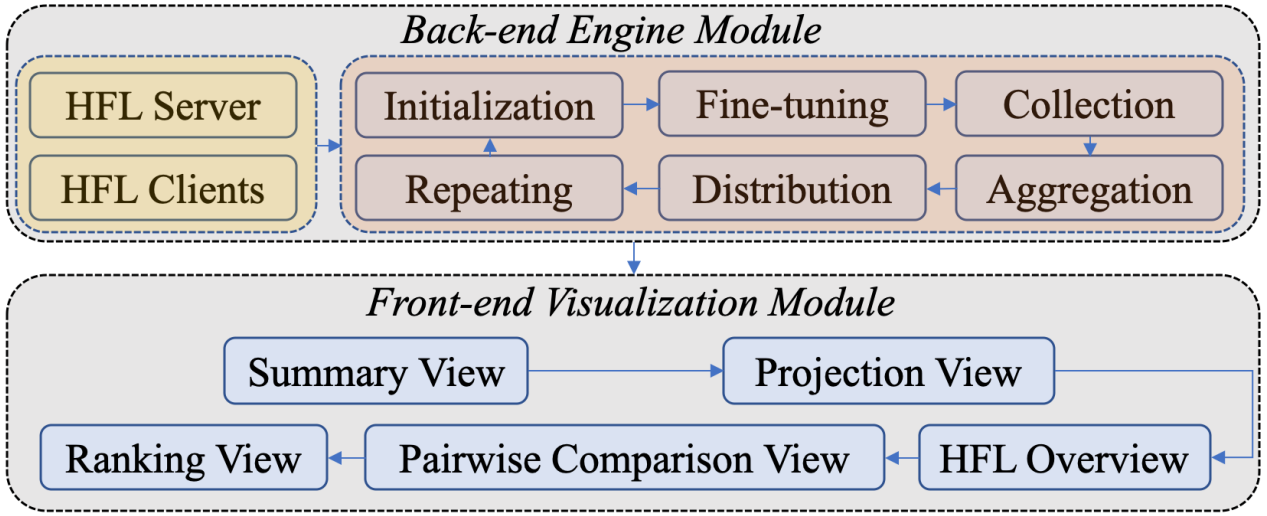
R.3 Inspect “clusters” and “outlier” of clients

R.4 Observe potential anomalies

R.5 Assess the contribution of clients

Domain experts from an AI solution provider

Approach Overview



	Feature Name	Symbol
Client	Client ID	k
	Round ID	r
	Local Sample Count	$N_k^{(r)}$
	Model Weights (Protected)	$W_k^{(r)}$
	Local Test performance	$P_k^{(r)}$
	Start Time	$t_k^{start(r)}$
	End Time	$t_k^{end(r)}$
	Model Weight Histogram	$H_k^{model(r)}$
	Model Gradient Histogram	$H_k^{gradient(r)}$
Server	Test Performance on Server	$P_{server}^{(r)}$
	Model Histogram on Server	$H_{server}^{model(r)}$
	Model Gradient Histogram on Server	$H_{server}^{gradient(r)}$

- The back-end engine module **initializes**, **optimizes**, and **collects** the log data during the HFL running process
- The front-end visualization module **visualizes** the extracted log data and supports an **interactive** data analysis

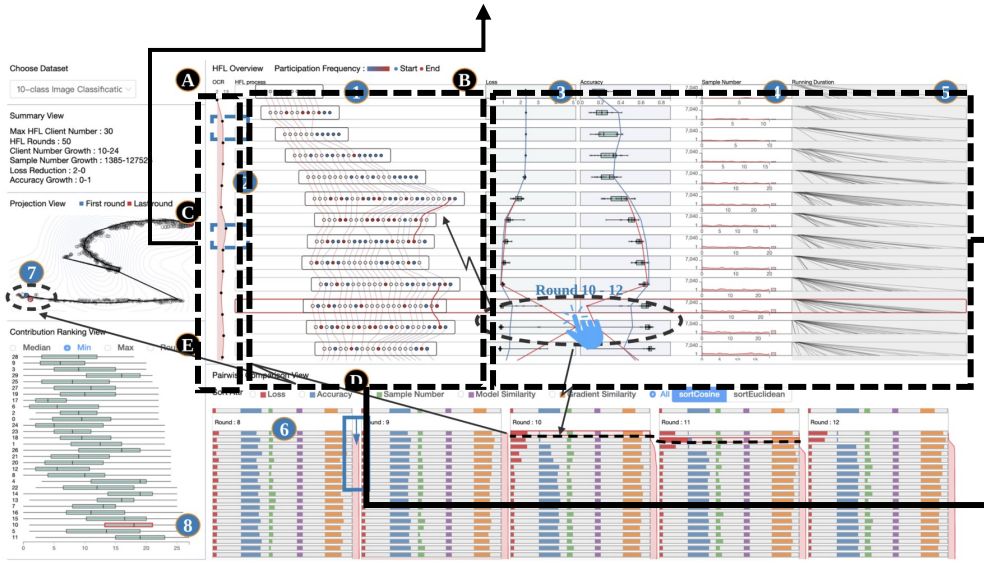
Information recorded in each HFL communication round

HFLens Interface

HFL Overview for Inspecting HFL Running Process

上海科技大学 ShanghaiTech University

Overall Change Ratio to measure HFL network change

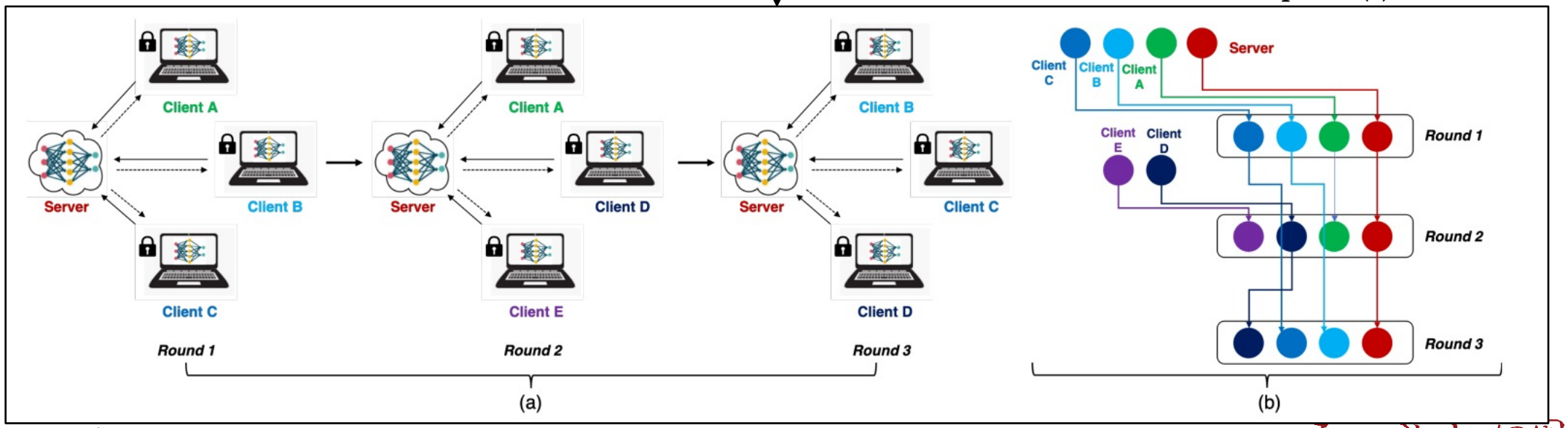


Auxiliary Information, including

1. The accuracy and loss of the local fine-tuned model in each communication round
2. the sample count for local training
3. the running duration of each participated client

An illustration of HFL overview design for inspecting HFL running process based on the example of (a)

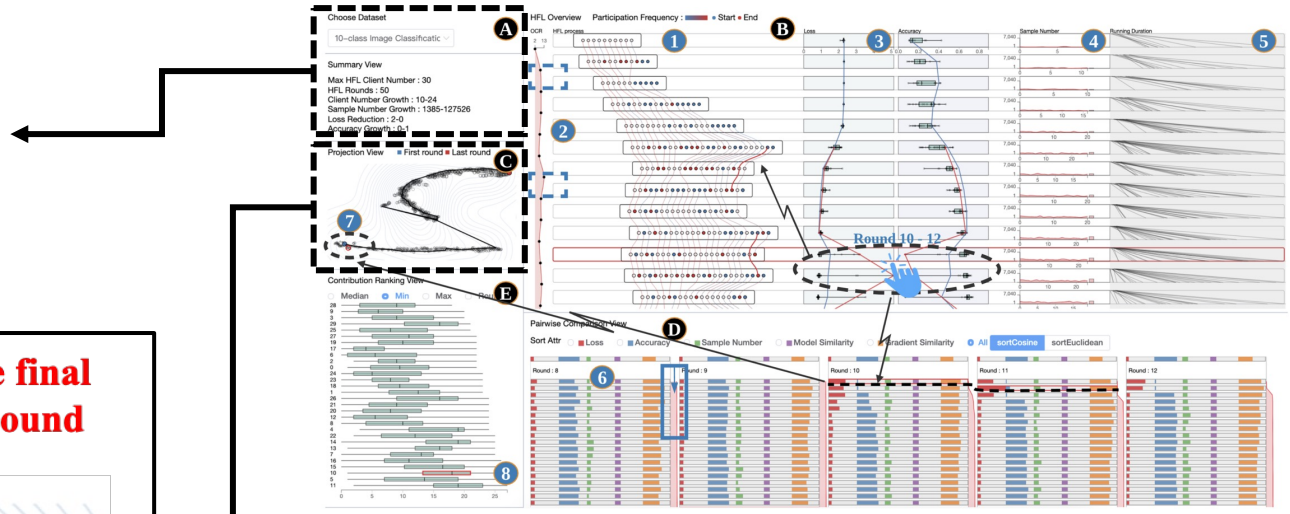
An example of a HFL process in three subsequent communication rounds

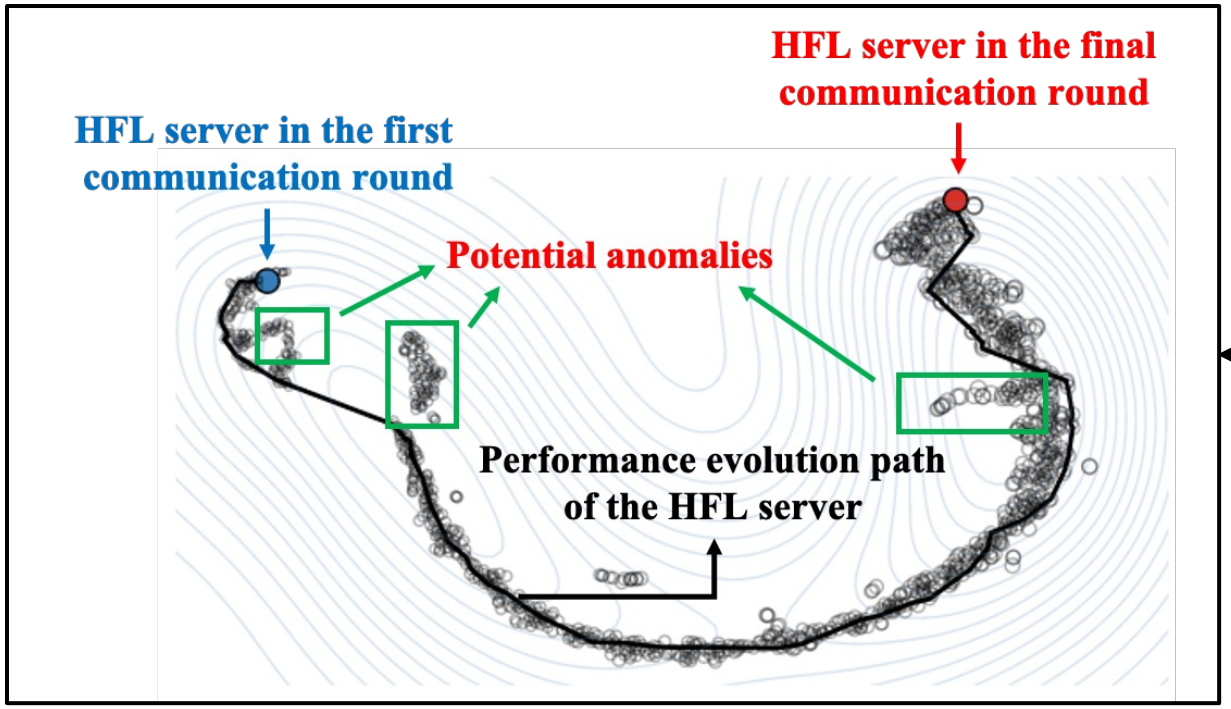


Summary View & Projection View

Statistical information:

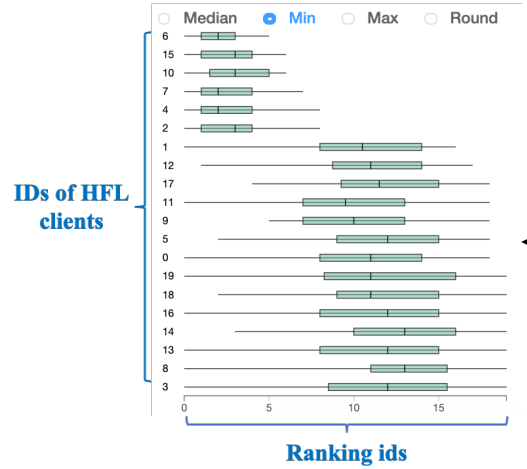
1. the number of HFL clients
2. the number of communication rounds
3. The sampling count
4. the loss reduction
5. the accuracy growth from the initial state to the final one



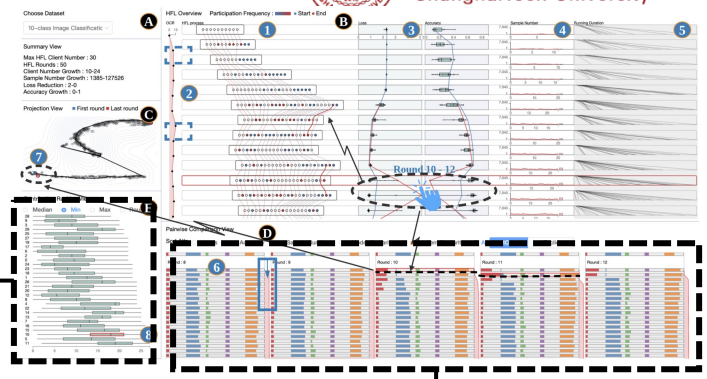


The projection view visualizes each HFL entity as a point in a 2D space which presents a pair of "client id/server – round id"

Pairwise Comparison View & Ranking View

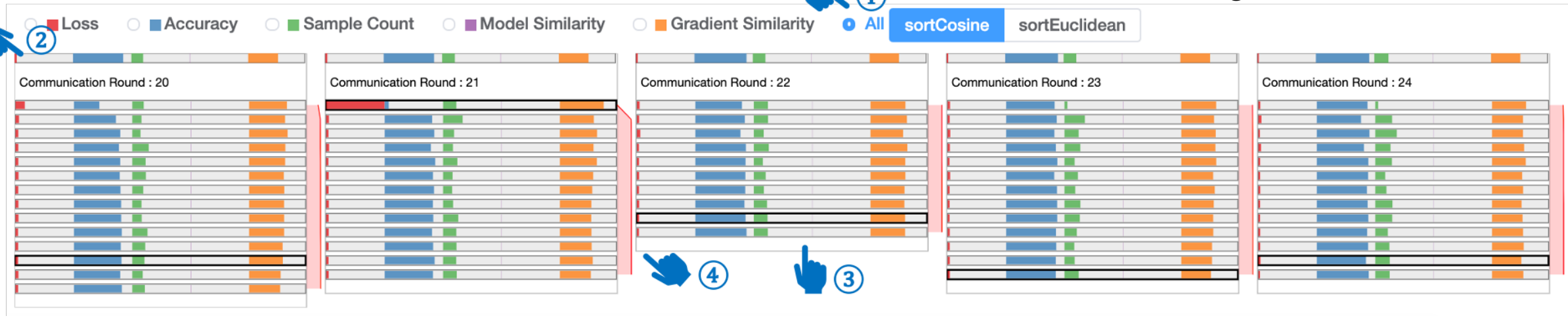


Ranking View
for Contribution Analysis



The top row always represents the metrics of the HFL server in different communication rounds

Users can sort the rankings of the clients



The curve area visualizes the similarity change between two adjacent HFL clients

- The columns represent the rankings of all the involved clients in the corresponding communication round
- The length of the color bars indicates the corresponding normalized value of the metric

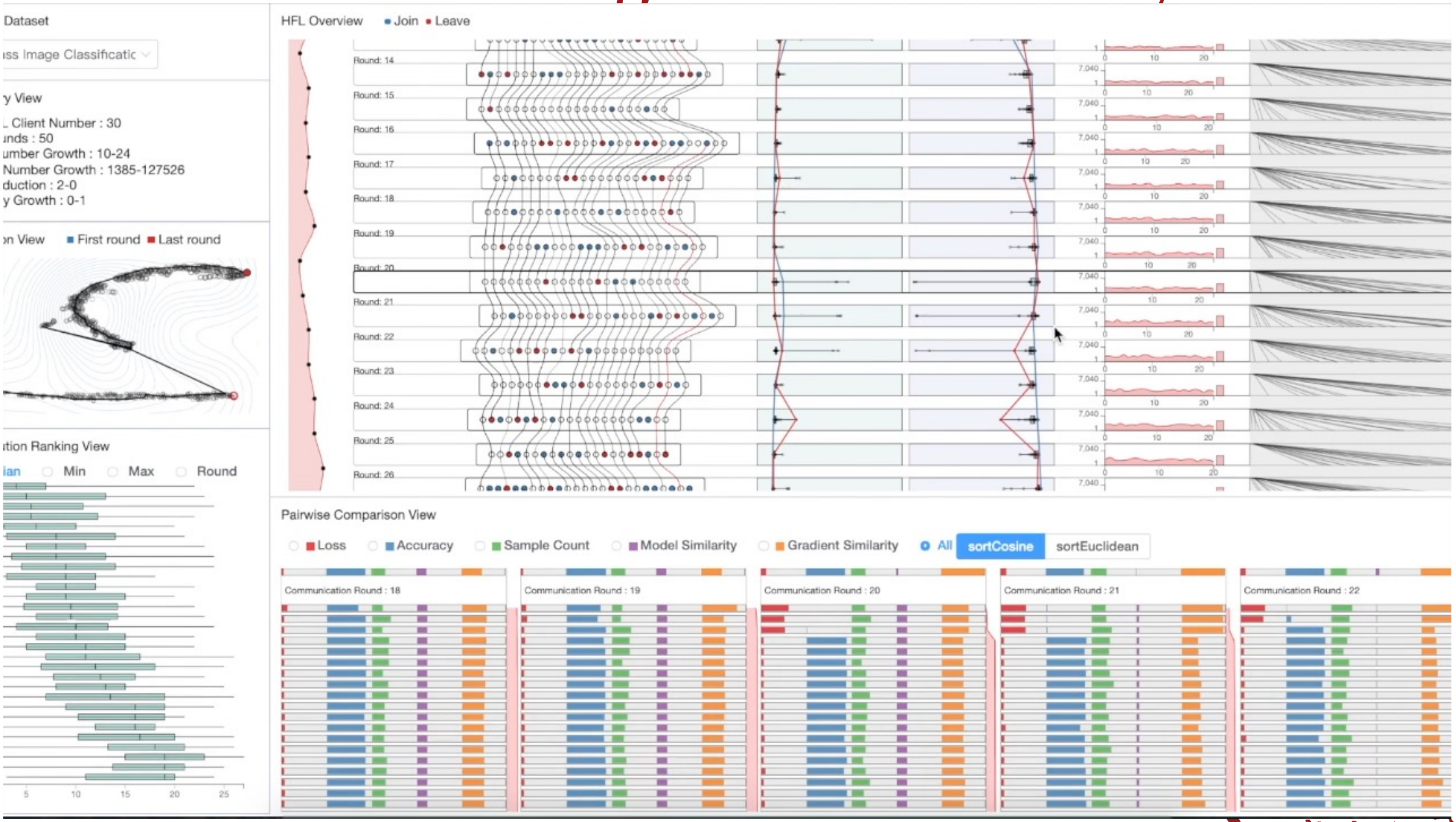
Case One: 10-class Image Classification by HFL

Experimental Setting

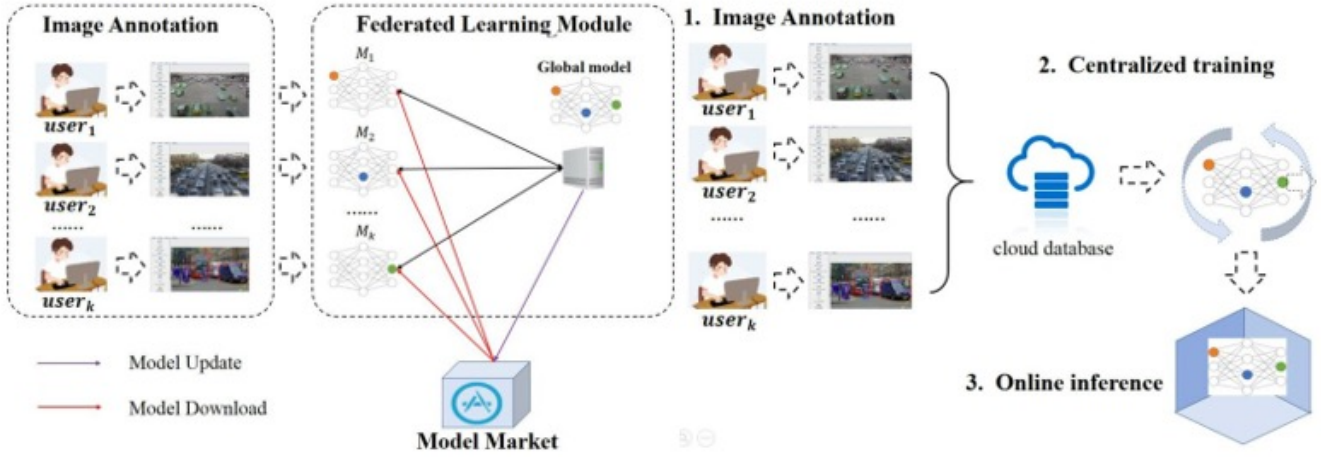
- Simulated **30 HFL clients** by using 30 terminals and ran a total of 50 HFL communication rounds
- Each client **randomly samples its local dataset** from the entire training data set and run an independent process as:

1. The clients can choose **when to join or leave** the HFL process
2. After receiving the aggregated model from the HFL server, each client **randomly chooses a time to start** the local finetuning process
3. The clients **randomly sample data** from the training data set and add them to their local training data set
4. Two kinds of client anomalies, **label-shuffle and adversarial-model-poison** happen during the process
5. Each client performs the local model fine-tuning process for 5 epochs

Case One: 10-class Image Classification by HFL



Case Two: Visual Object Detection by HFL



Pipeline of FedVision

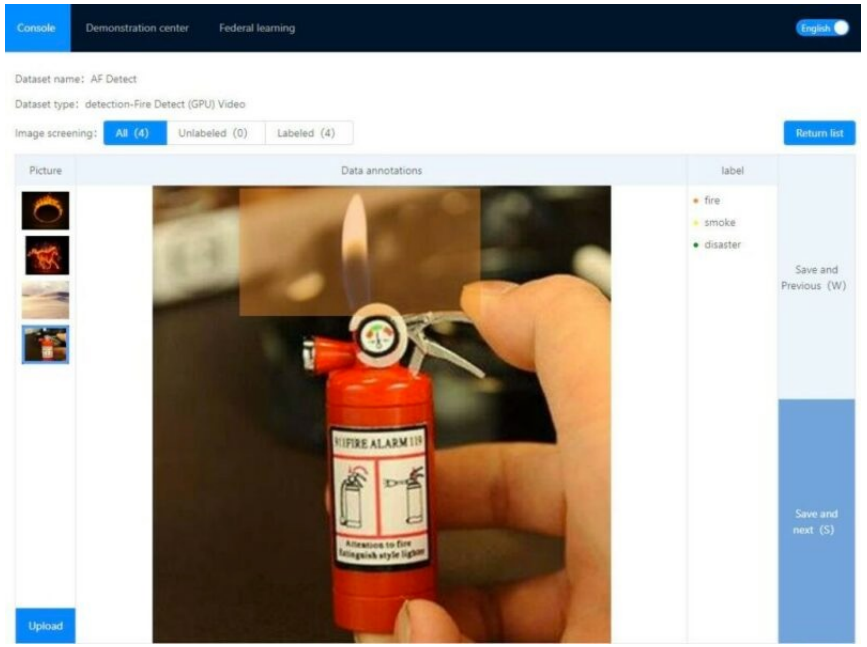
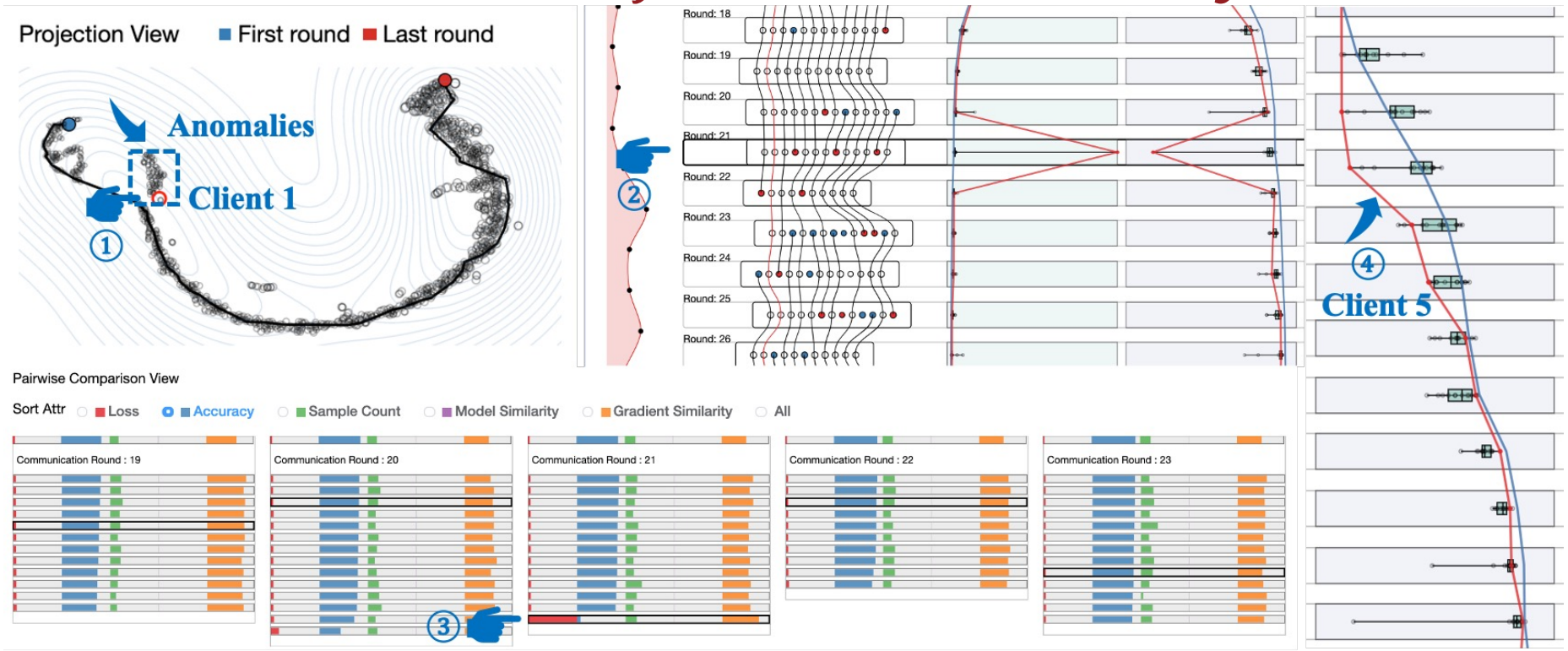


Image labeling

- An HFL-based video detection system that includes surveillance systems from several worksites
- The workers select some frames from the surveillance video for labeling
- Has gone through 40 communication rounds
- The total number of clients is 15

https://cn.fedai.org/cases/computer-vision-platform-powered-by-federated-learning/

Case Two: Visual Object Detection by HFL



Finding:

- The performance fluctuation of client 1 was caused by the improper operation of a newcomer
- Client 5 is a factory surveillance system, whereas the other clients are all construction worksites are all construction worksites

Action:

- Notified the relevant worksites and disseminated the proper labeling method to all the involved colleagues
- Adjusted the subsequent aggregation strategy in FedVision by lowering the aggregation weight of Client 5
- The HFL network stability of FedVision must be considered

Discussion and Limitation

Discussion

1. System Performance
2. Feedback and Takeaway
3. Visual Design and Interaction
4. Generalizability and Scalability

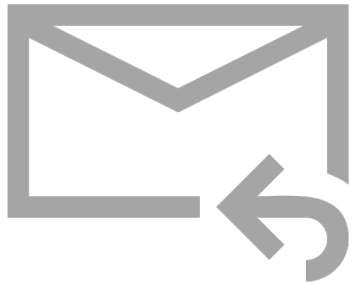
Limitation

- Only worked with a small team of experts
- New attacking algorithms, which do not have differentiable performance or histogram, may not be detected by our system
- Based on several metrics and more metrics can be included according to different scenarios
- Hardware utilization and clients' configuration are also important factors to be considered in future work

Conclusion

- Theory
  - Rigorous theory (cognition + CS) of explainability and explanation
  - Proper evaluation of explainability and the quality of an explanation
  - How to model the bias and variance of human
- Application
  - Real-world applications for end-users
  - Design guidelines
  - Human learn from AI?

Questions?
Thank you ☺



Quan Li

liquan@shanghaitech.edu.cn