CSE 564
Visualization & Visual Analytics

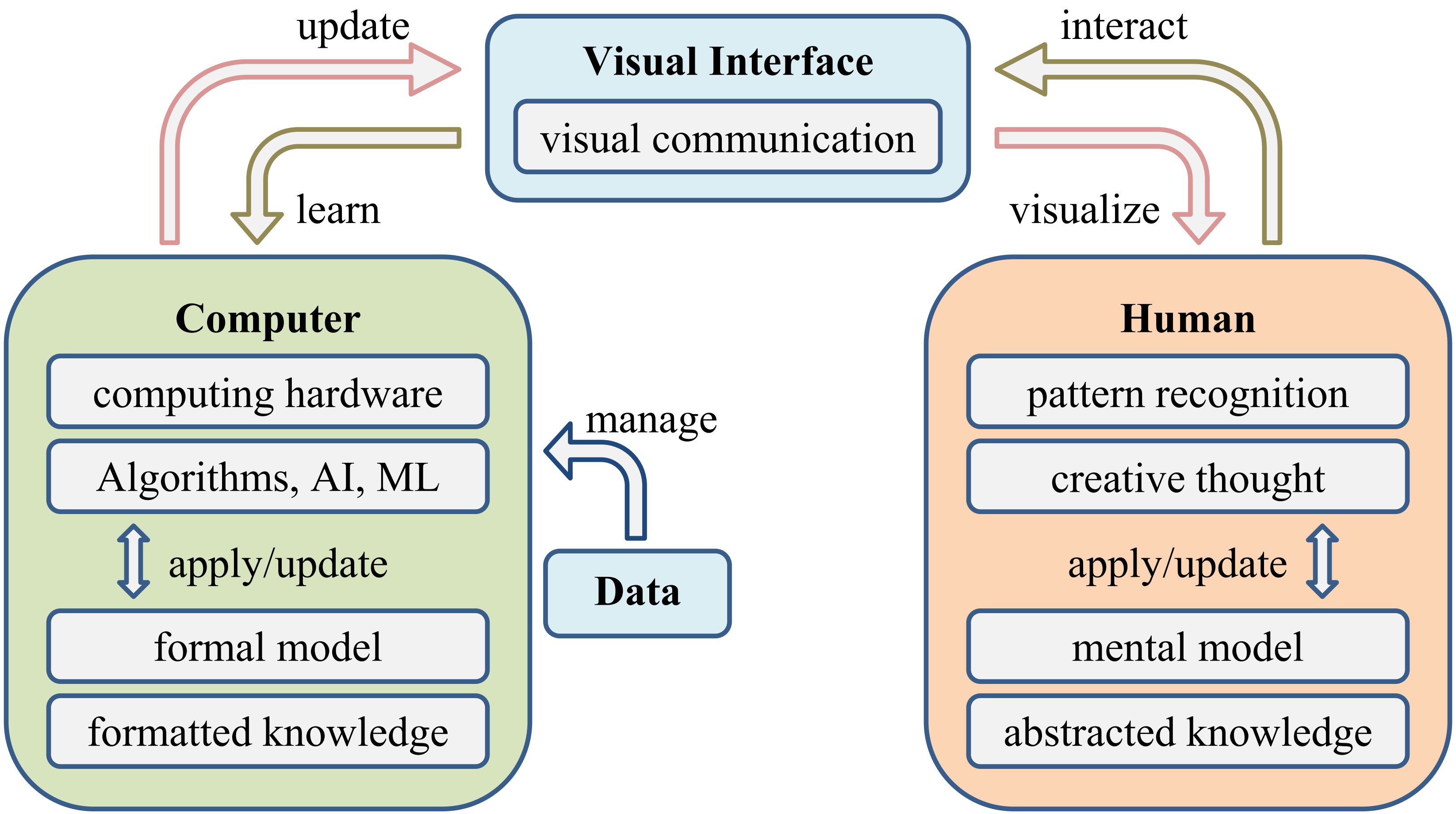
Visualization and AI

Klaus Mueller

Computer Science Department
Stony Brook University

Lecture	Topic	Projects
1	Intro and logistics	
2	Basic visualizations and tasks, data types, examples, ethical considerations	
3	Data preparation (cleaning, imputation, data set integration)	
4	AI-assisted coding for VIS applications (design, debugging, refactoring)	Project #1 out
5	Big data and data reduction (distance/sim metrics, intro to clustering)	
6	High-D data: concept, subspaces, dimension reduction, PCA	
7	Cluster analysis: hierarchical, density, model, embedding, temporal	
8	Perception and cognition (human visual system, color, contrast)	Project #2(a) out
9	Visual design and aesthetics	
10	Visualization of multivariate and high-D data: linear methods, projections	
11	Vis. of multivariate and high-D data: non-linear methods, embeddings	
12	Visualization and AI: mutual support and capabilities (VIS4AI, AI4VIS)	Project #2(b) out
13	Principles of interaction: drive what is visualized, analyzed & how (HCI4VIS)	
14	Visual analytics (VA), human-centered AI, mixed-initiative system	
15	Midterm #1 (tentative date)	
16	VA system design and evaluation, collaborative VA, uncertainty, provenance	
17	Midterm #1 discussion (tentative date)	Final proj. proposal call out
18	Visualization of hierarchical data	
19	Visualization of maps and data with geo-reference	
20	Visualization of graphs, networks (incl. derivation of causal networks)	Final project proposal due
21	Vis. of time-varying, time-series, streaming data, progressive visualization	
22	Visualization of text, LLMs, and semantic data	
23	Ed Tufte revisited: principles, critiques and limits, responsible visualization	
24	Design of effective infographics	Final proj. prelim report due
25	Foundations scientific and medical visualization, intro to volume rendering	
26	Scientific visualization	Bonus project out (Vol Ren)
27	Story telling with data, data journalism	
28	Midterm #2 (tentative date)	
Final	Final project demo on zoom (public)	All final proj. materials due

Human-AI Collaboration



K. Mueller, S. Garg J. Nam, T. Berg, K. McDonnell, "Can Computers Master the Art of Communication? A Focus on Visual Analytics, IEEE CG&A, 2011

VIS4AI

If AI becomes more powerful, will visualization become less important or more important?

Becomes more important → VIS4AI

- Model complexity grows and becomes difficult to interpret or debug
- Decision processes become opaque ("black-box" models)
- Output volume increases beyond what humans can inspect manually
- Trust and accountability are required → visualization provides transparency
- Human oversight remains essential for validation and control

AI4VIS

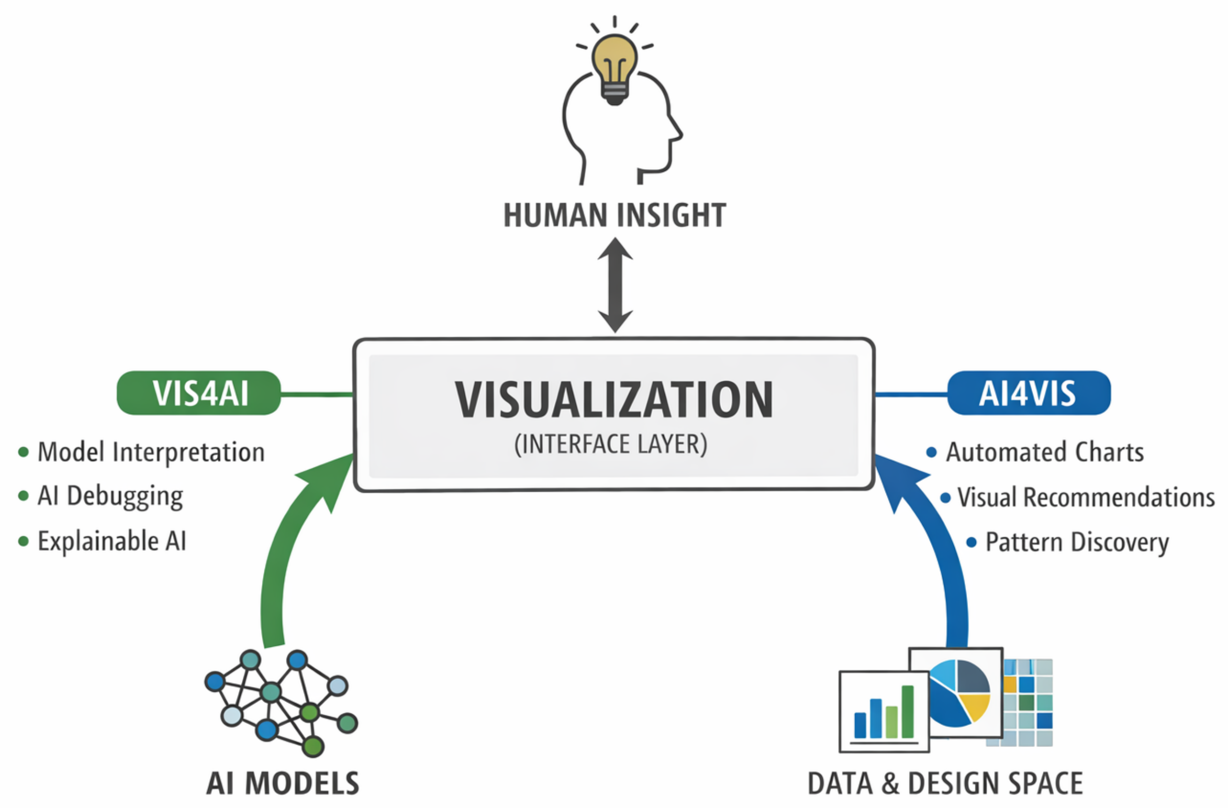
If datasets become larger and more complex, can humans design all visualizations themselves?

AI can help → AI4VIS

- Large datasets create enormous visualization design spaces
- AI can recommend effective charts and layouts
- Natural language interfaces allow users to generate visualizations quickly
- AI can detect patterns, anomalies, and insights automatically
- Analysts can focus on task specification and interpretation rather than manual chart design

Human-AI Collaboration in the Age of AI

Visualization is the communication layer between humans and intelligent systems



Visualization for AI (VIS4AI)

Why Visualization for AI?

Modern AI systems are difficult to interpret:

- models contain millions or billions of parameters
- internal computations are opaque
- predictions can be difficult to explain

Visualization helps:

- reveal patterns in model behavior
- support debugging and model development
- communicate model decisions

Visualization turns AI systems from black boxes into analyzable systems

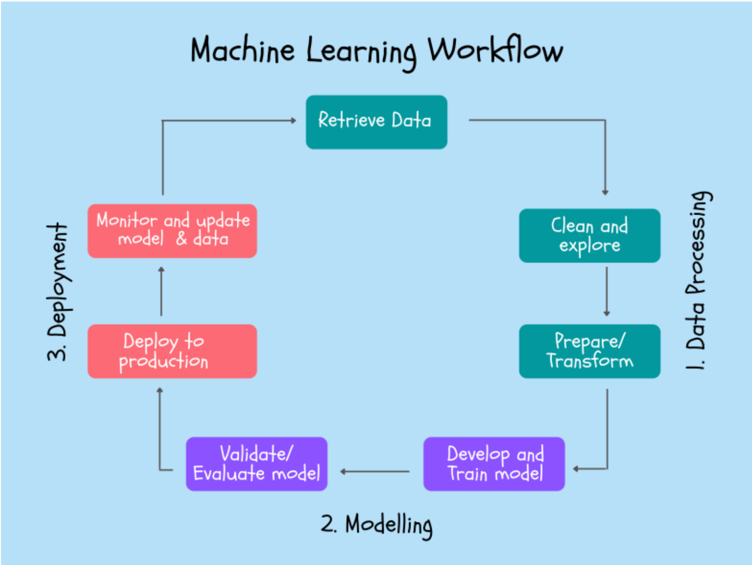
AI Model Development is Iterative

The typical machine learning workflow:

- Data → model training → evaluation → refinement → deployment

Visualization can support each of these steps:

- monitor model training
- compare models
- identify errors
- understand predictions



Monitor Model Training

Common visualizations:

- loss curves
- accuracy curves
- learning rate schedules
- parameter distributions

Can help detect:

- overfitting
- underfitting
- unstable training
- convergence problems

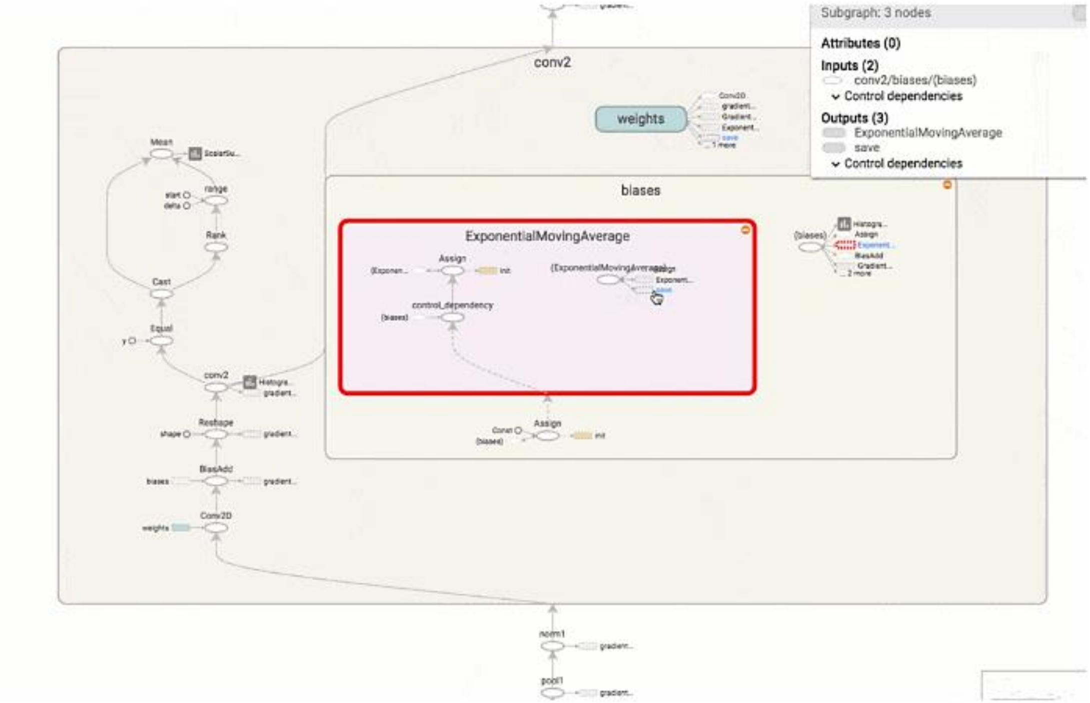
All images:

- [Google Tensorboard]

Example: Google tensorBoard

Good example for the general **visual training-monitoring paradigm**

- aside from these plots it also allows a view into the network itself



See this 5-minute videofor a summary

- A tensor is essentially a data matrix with 'depth' – a multi-dimensional array

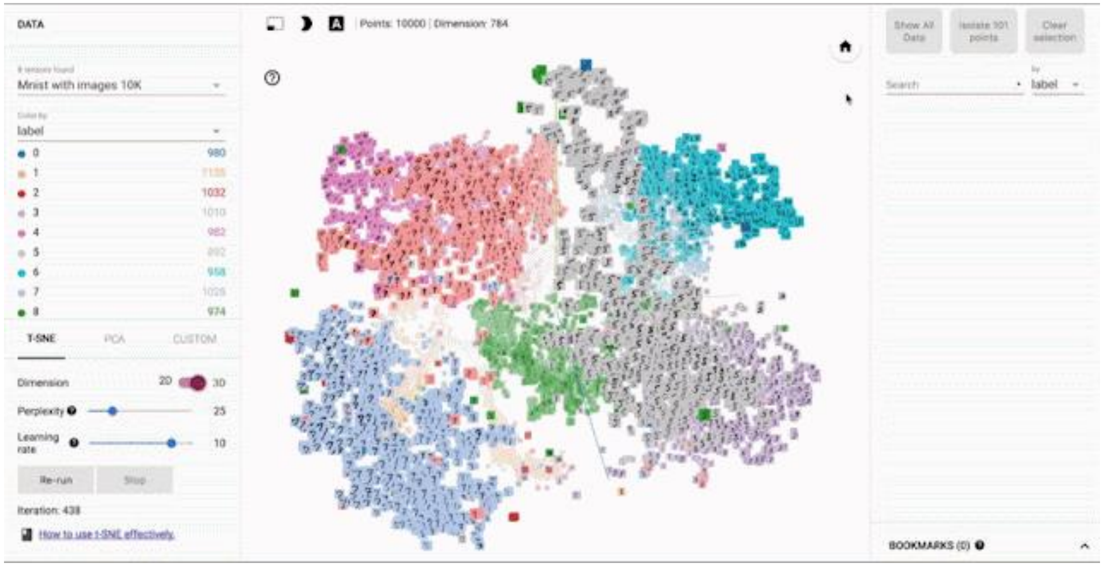
TensorBoard Embedding Projector

Embedding visualizations help developers:

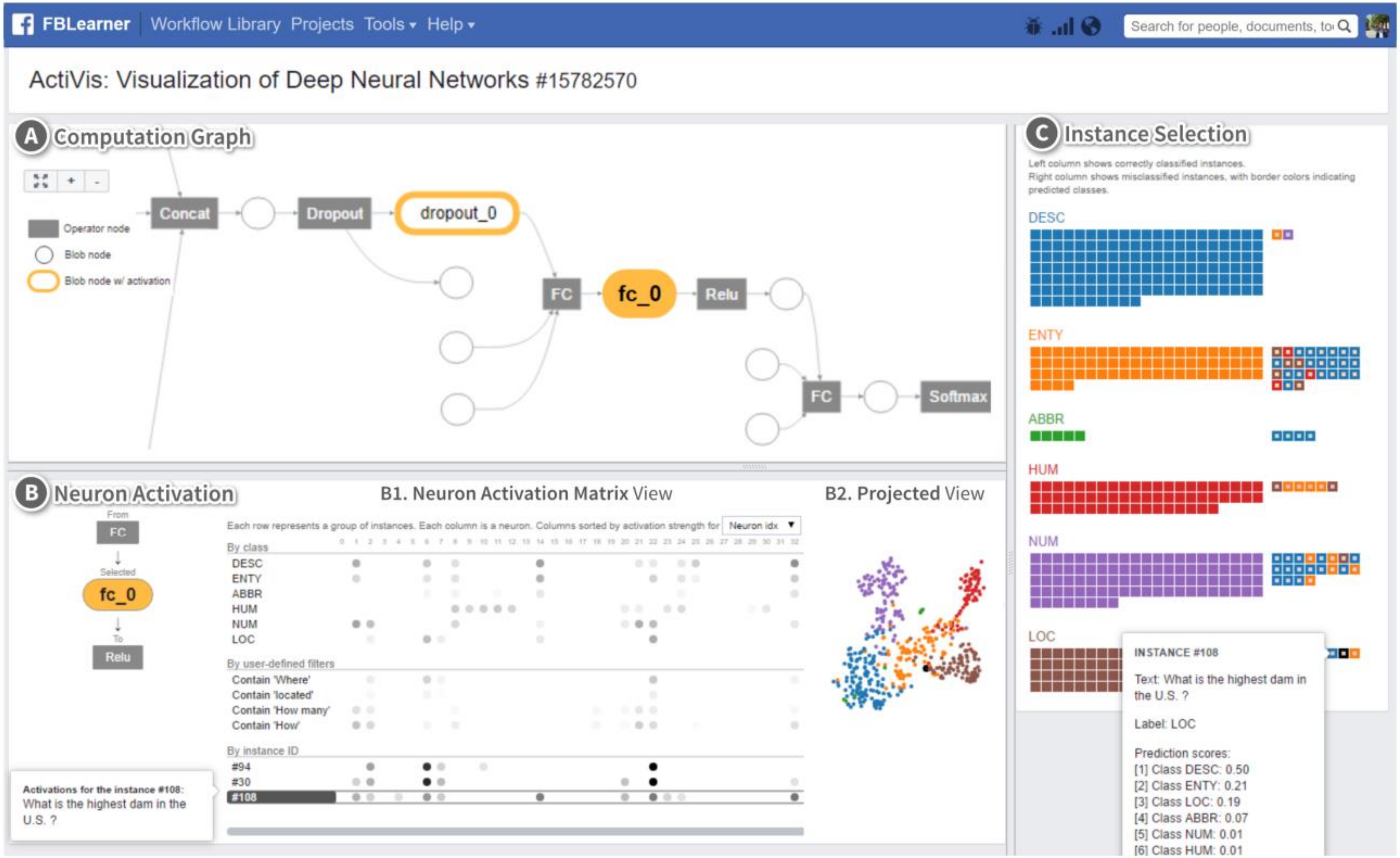
- evaluate representation quality
- identify mislabeled data
- detect model weaknesses → if two classes overlap heavily → misclassifications will likely occur
- understand the learned structure – the representation's geometry

Can use Word2Vec, t-SNE, or any other

Shown here is the MNIST dataset



Example: ActiVis



ActiVis Figure Caption

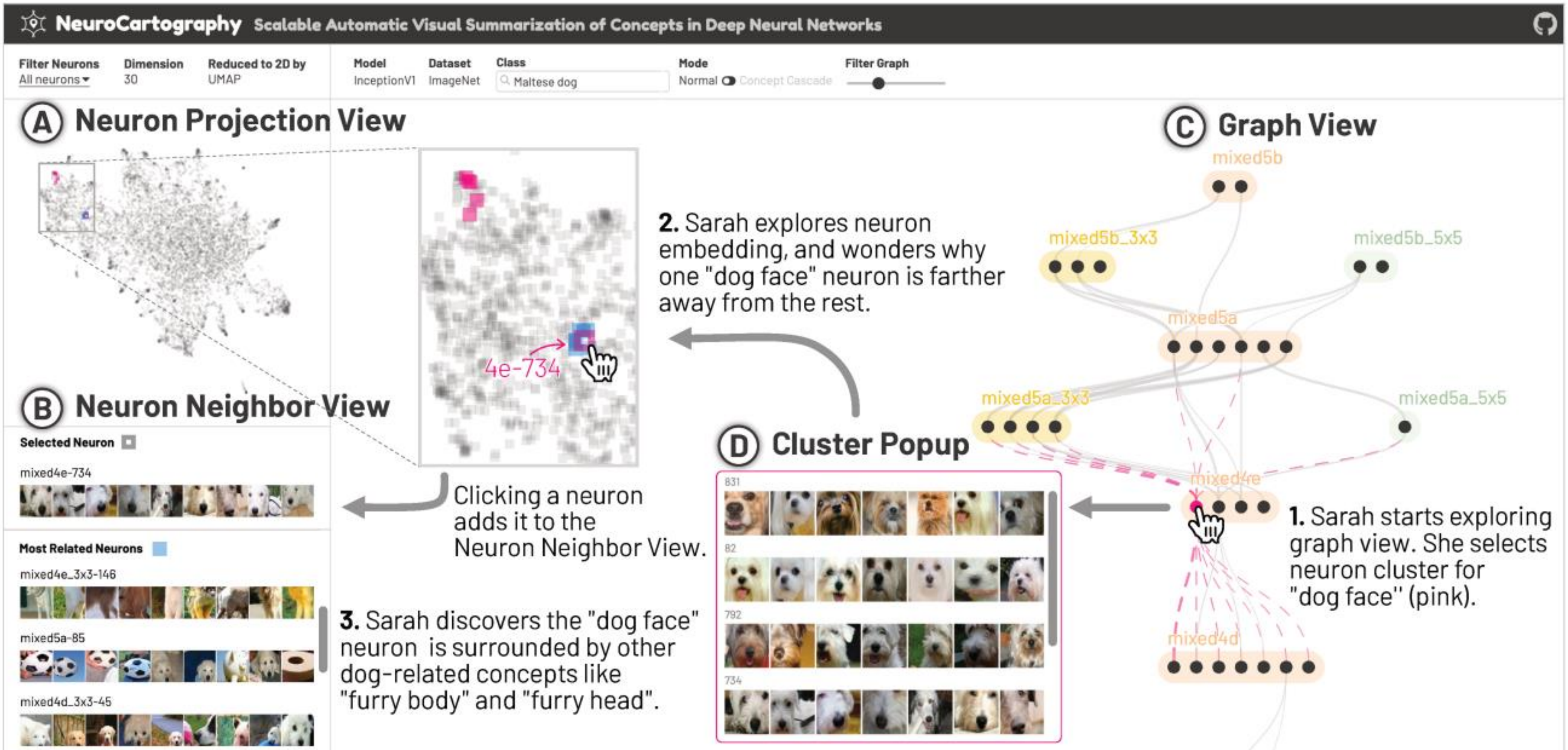
Fig. 2. ACTIVIS integrates multiple coordinated views.

A. The computation graph summarizes the model architecture (the neural network)
B. The neuron activation panel’s matrix view displays activations for instances, subsets, and classes (at B1), and its projected view shows a 2-D t-SNE projection of the instance activations (at B2).
C. The instance selection panel displays instances and their classification results; correctly classified instances shown on the left, misclassified on the right. Clicking an instance adds it to the neuron activation matrix view.

The dataset used is from the public TREC question answering data collections The trained model is a word-level convolutional model.

Kahng, Minsuk, et al. "ActiVis: Visual exploration of industry-scale deep neural network models." *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017): 88-97.

Example: NeuroCartography



Park, Haekyu, et al. ": Scalable automatic visual summarization of concepts in deep neural networks." *IEEE Transa*Neurocartography*ctions on Visualization and Computer Graphics* 28.1 (2021): 813-823. [video]

Take Aways

Visualization dashboards like these allow analysts to:

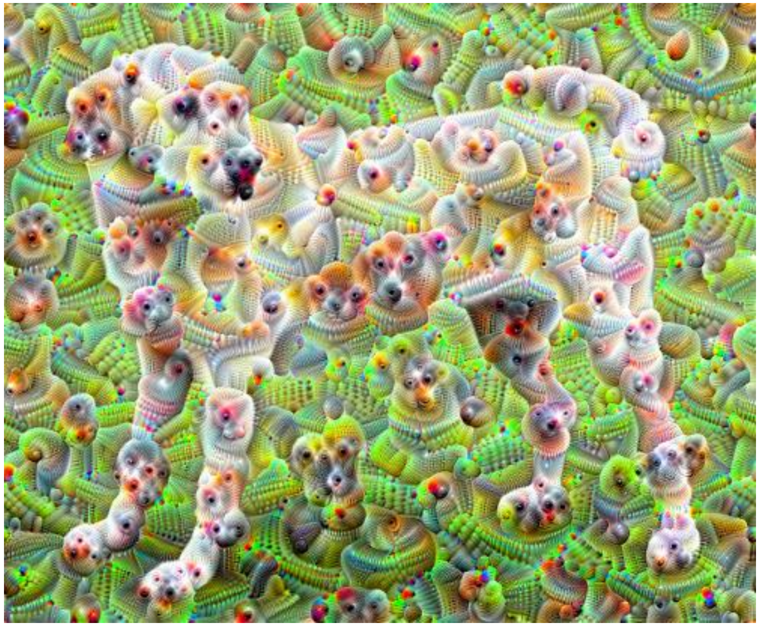
- inspect neuron activations
- Identify neurons that never activate
- see which layer failed
- identify misleading features

They help with:

- model interpretability
- debugging deep networks
- understanding feature hierarchies

Visualize on the Image Level

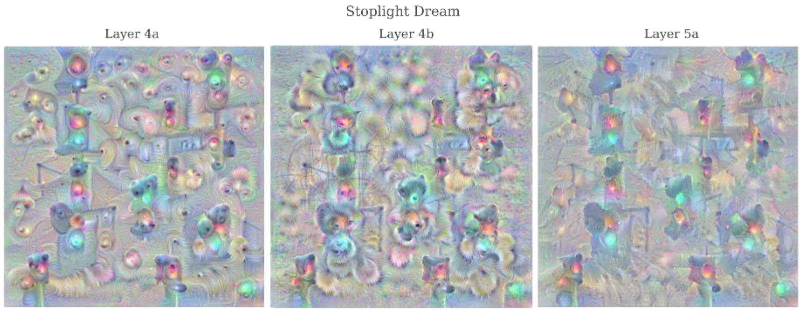
Google "DeepDream"



DeepDream– What Happens

Instead of visualizing the network structure, this technique visualizes the features neurons detect.

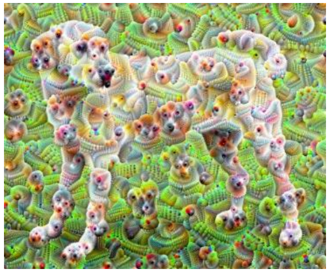
- Choose a neuron in a neural network
- Modify the input image to **maximize its activation**
- Observe what patterns emerge



Layers:

- Early layers detect simple patterns, edges, lines, color gradients
- Middle layers detect textures, fur, feathers, shapes
- Later layers detect objects, dog faces, wheels, buildings
- DeepDream exaggerates these patterns until they become visible

So Why So Many Dog Faces?



DeepDream amplifies patterns that strongly activate neurons in the network.

Several factors make dog features dominate:

- **Training data bias:** the network was trained on ImageNet, which contains many dog breeds → many dog detectors
- **Strong learned dog features**: the model learned to detect dog parts such as eyes, snouts, fur textures
- **Feature amplification**: DeepDream modifies the image to increase neuron activation, strengthening dog-like patterns
- **Recursive reinforcement**: once a dog-like feature appears, the algorithm enhances it again → dog faces appear inside other dog faces

DeepDream reveals what the network is most inclined to see (has learned most strongly) based on its training data → dog patterns

AI for Visualization (AI4VIS)

Why Use AI for Visualization?

Modern datasets are becoming:

- larger
- higher dimensional
- more complex

This creates a challenge:

- The space of possible visualizations becomes enormous.

AI can help by:

- recommending visualizations
- detecting patterns automatically
- assisting analysts in exploring data

The Visualization Design Space

The Visualization Design Space is Huge

Consider a dataset with:

- 10 attributes
- several aggregation choices
- many chart types

Possible visualization combinations can be thousands or millions

Example choices:

- which variables to plot
- which chart type to use
- how to encode color/size/shape
- what aggregation to apply

AI can help search this design space

Visualization Recommendation Systems

Try to automatically suggest useful visualizations for a dataset

Rule-Based Systems (Early VIS Research: Voyager, ShowMe)

- The system generates charts that satisfy design rules
  - quantitative variable → x-axis | categorical variable → color
  - avoid pie charts for many categories| prefer bar charts for comparisons

Constraint-Based Systems (example: Draco)

- Dataset → generate candidate visualizations → evaluate constraints (perceptual rules) → select best visualization

Machine Learning Systems (Recent VIS Work: Data2VIS, VizML)

- Learn from large datasets + example visualizations
- Try to predict good charts automatically
- Essentially this is AI-driven visualization design

Rule-Based Systems: Voyager

Checks whether visual encodings match the data types

- Quantitative variables → position on axes
- Categorical variables → color or grouping
- Time variables → line charts

Favors visualizations that reveal interesting statistical patterns

- correlations between variables
- strong trends
- clusters and outliers

Uses perceptual and cognitive rules for effective visual encoding

- prefer position, length (recall concept of visual variables)
- avoid relying on color hues and area for precise comparison

Encourages

- different types of views for multiple perspectives
- visualizations that minimize clutter and an encoding of too many variables

Encodes visualization knowledge that experienced analysts normally apply manually

Wongsuphasawat, Kanit, et al. "Voyager: Exploratory analysis via faceted browsing of visualization recommendations." *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015): 649-658.

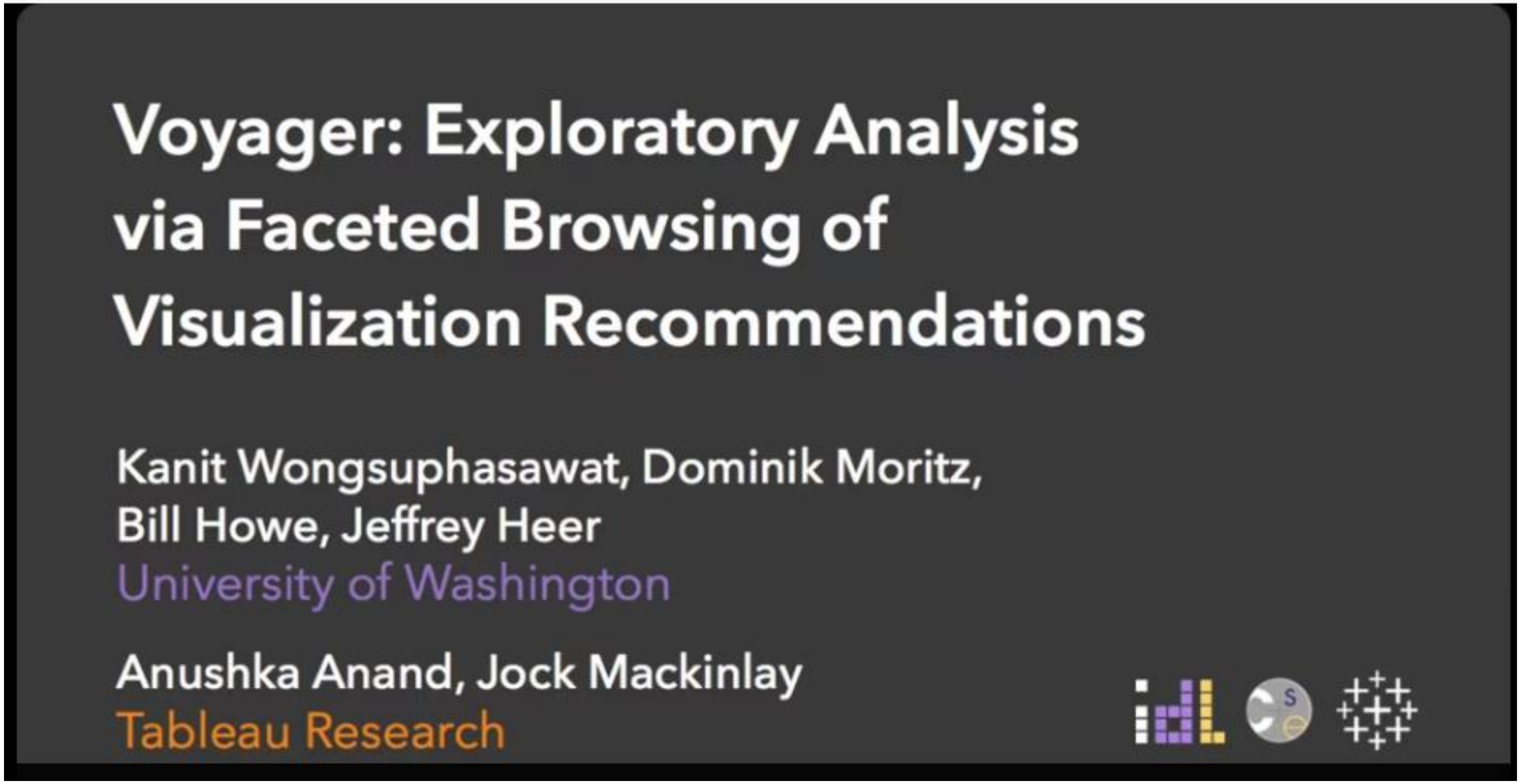
Voyager Dashboard

The **schema panel** that lists data variables selectable by users

The **main gallery** that presents suggested visualizations of different variable subsets and transformations.

Voyager Video

https://vimeo.com/135417594



Constraint-Based Systems: Draco

Draco turns visualization design into an optimization problem

- The best visualization is the one that minimizes violations of visualization principles
- Essentially this is Voyager but with an optimization engine

Examples:

- quantitative variables should map to position
- categorical variables should map to color or grouping
- avoid encoding too many variables simultaneously
- prefer perceptually effective visual encodings

Draco searches for visualizations that best satisfy these constraints

Moritz, Dominik, et al. "Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco." *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018): 438-448.

Draco Specifics

Rules describe relationships between:

- data fields
- visual encodings
- chart types
- design guidelines

Examples of rules Draco evaluates:

<u>Prefer:</u>

- quantitative variables → x or y axis
- categorical variables → color or grouping

<u>Avoid:</u>

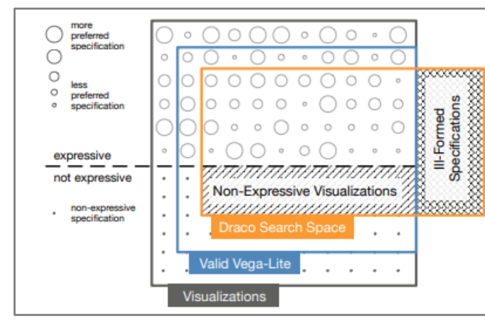
- mapping many variables to the same visual channel
- using weak visual encodings for precise comparisons

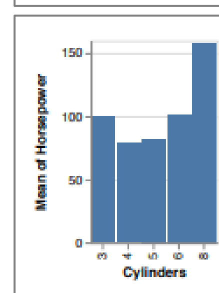
Each rule has a cost or penalty

The system finds visualizations with lowest total cost

Workflow:

dataset
↓
generate candidate visualization specs
↓
apply visualization design constraints
↓
score candidate visualizations
↓
recommend best designs





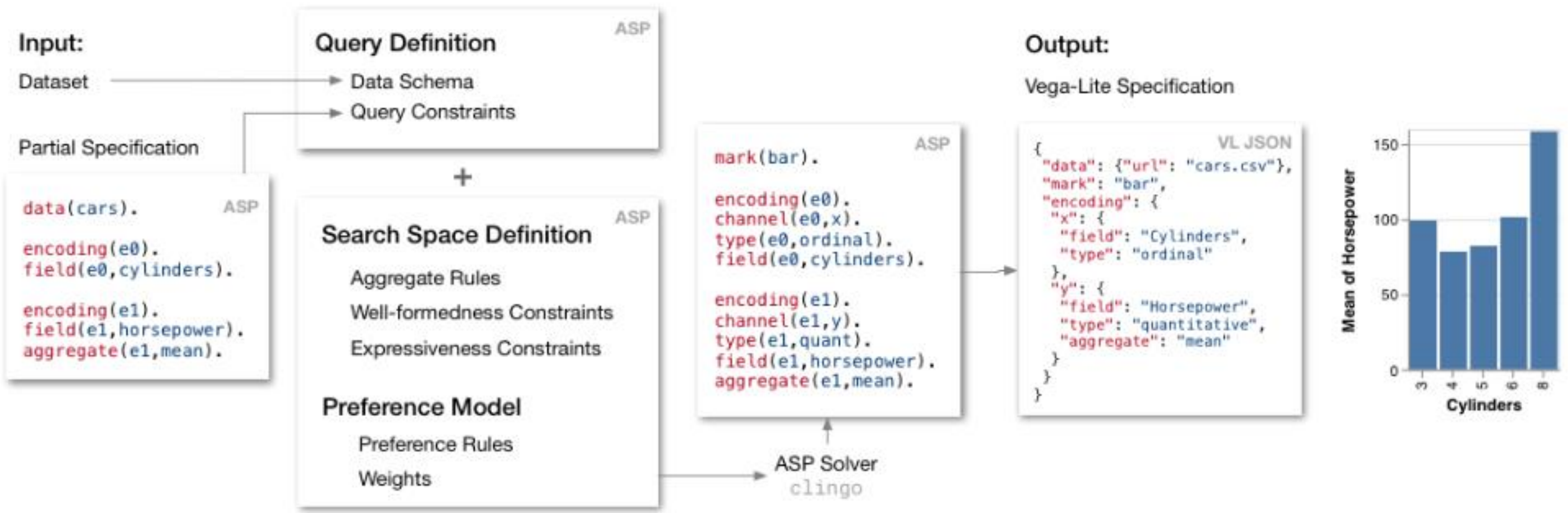
ASP

```
mark(bar).

encoding(e0).
channel(e0,x).
type(e0,ordinal).
field(e0,cylinders).

encoding(e1).
channel(e1,y).
type(e1,quant).
field(e1,horsepower).
aggregate(e1,mean).
```

Draco Implementation



Implementation of the optimal encoding search process using constraints. Draco compiles a user query (including the dataset, the partial specification, and the task) into a set of rules and combines them with the existing knowledge base to form an ASP program. Draco then calls Clingo to solve the program to obtain the optimal answer set. Finally, Draco translates the answer set into a Vega-Lite specification.

GENERATING VISUALIZATIONS WITH DEEP LEARNING: DATA2VIS

The system learns to generate visualization via a neural network from many real-world examples

- learns the association: Dataset → Best Visualization Specification
- uses a sequence-to-sequence neural network (it was 2019)
- the model is trained on many examples of datasets paired with best practice visualization designs

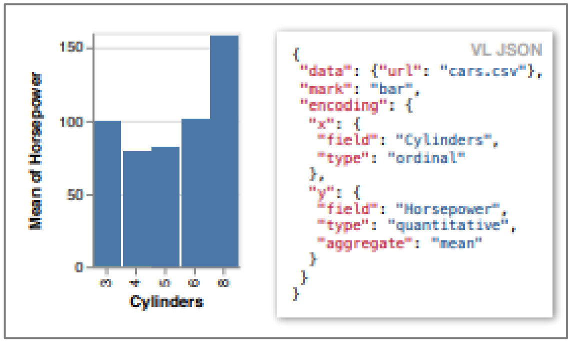
Training phase:

- datasets + example charts → train neural network
- used a dataset of 4k charts

Prediction phase:

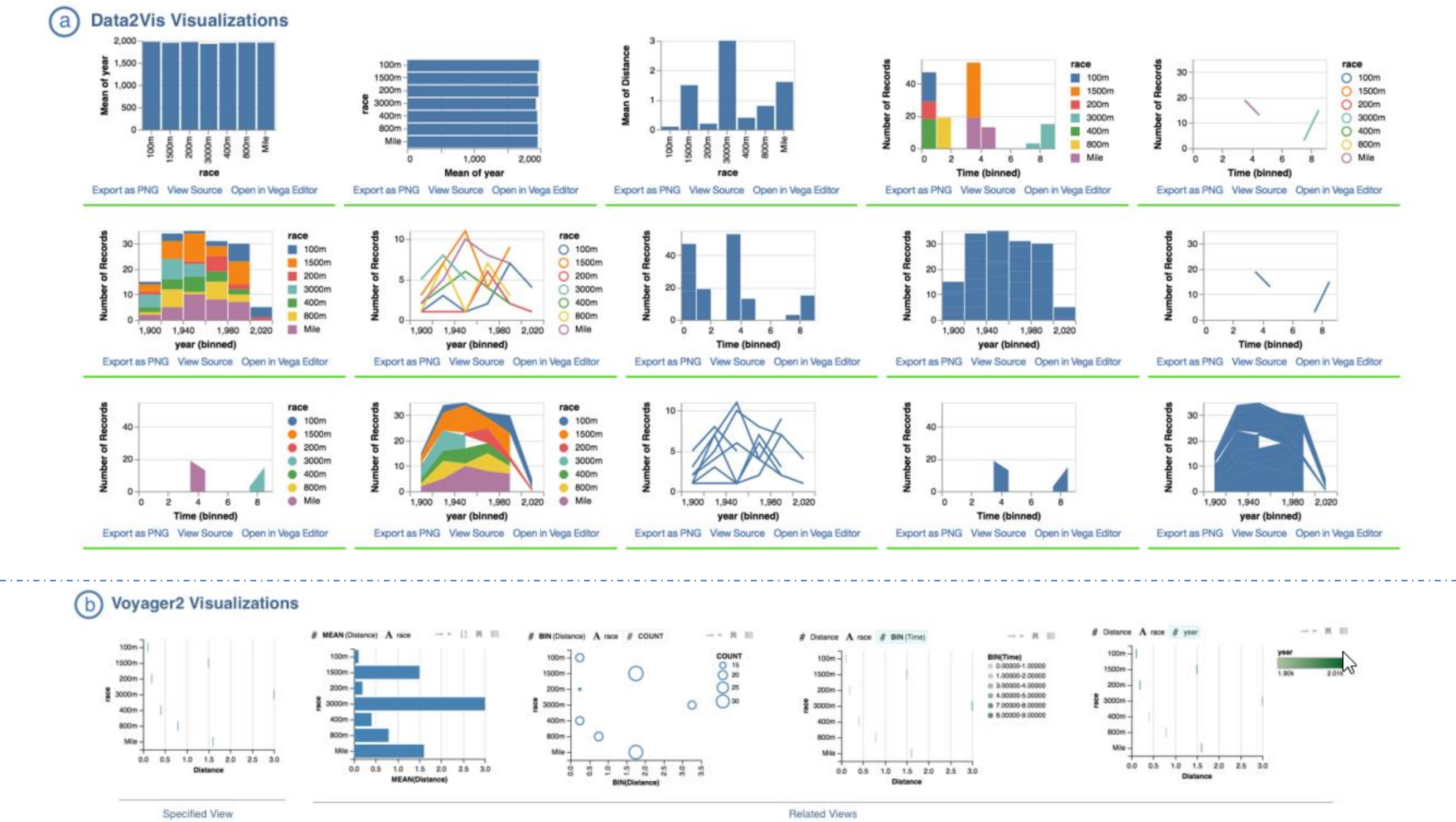
- new dataset → generate chart specification

The output is typically a Vega-Lite visualization



Dibia, Victor, and Çağatay Demiralp. "Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks." *IEEE Computer Graphics & Applications* 39.5 (2019): 33-46.

Data2Vis Generated Charts: More Diverse than Voyager 2



AI4VIS Summary

System	Year	Approach	Training Data Source	Approx. Size	Example Output
Voyager	2015	rule-based recommendation	visualization design heuristics	—	ranked candidate charts
Draco	2018	constraint-based optimization	visualization design constraints	—	optimized visualization specification
Data2Vis	2017	deep learning (seq2seq)	Vega-Lite example specifications	~4K visualizations	generated Vega-Lite chart
VizML	2018	machine learning prediction	Tableau Public dashboards	~1M visualizations	predicted chart type & encodings
MLViz systems	~2021	learned visualization design	multiple visualization repositories	hundreds of thousands	recommended visualization mappings
LLM-based systems	2023+	foundation models	large code and visualization corpora	millions–billions	visualizations generated from prompts

Visualization recommendation has shifted from expert rules to data-driven AI trained on large collections of human visualizations

VizML video is here https://www.youtube.com/watch?v=lUq35pi98yI

Automating Task Identification for Visualization

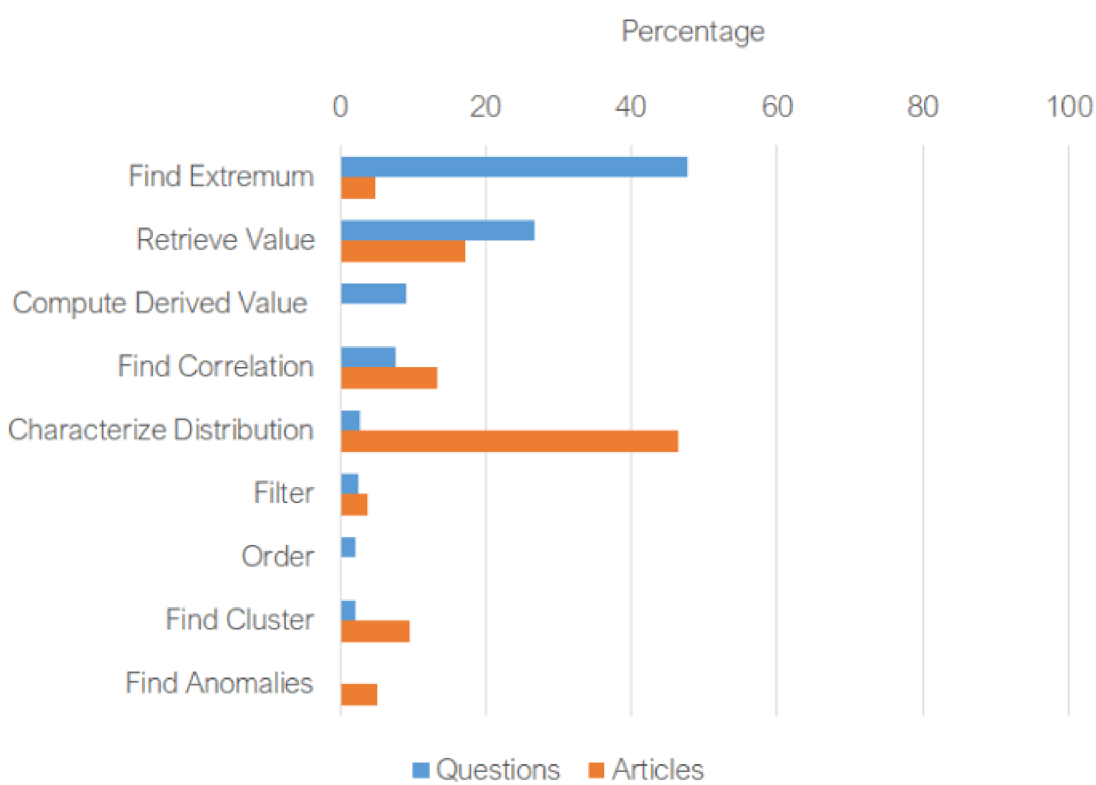
Visualization recommender systems help users choose appropriate charts for their data

- Require visualization design rules
- Require empirical models learned from data
- But require users to specify analytical tasks beforehand

Identifying the relevant analytical tasks often requires domain expertise

- **TaskFinder** aims to automatically identify relevant analytical tasks which can then be mapped to the best visualization type (see future lecture)
- analyzes domain-specific textual documents to detect relationships between data attributes and analytical tasks

Types of Tasks

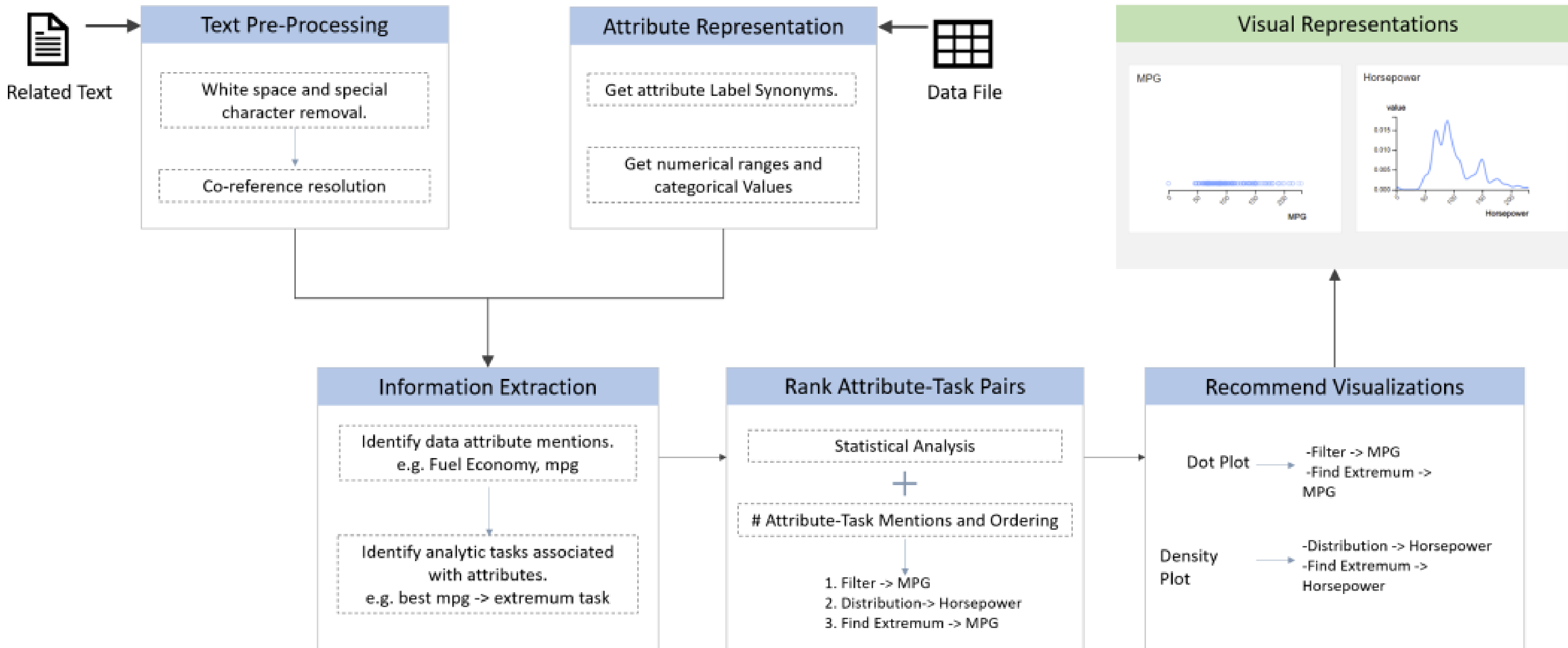


In a car dataset, a common question is "What is the maximum horsepower of all cars?"
'Maximum' → 'Find Extremum' task
Variable → Horsepower
Use a histogram over HP

"How does weight affect acceleration?"
'Affect' → 'Find Correlation' task
Variables → Weight, Acceleration
Use a scatterplot of acceleration (y) vs, weight (x)

Coelho, Darius, Bhavya Ghai, Arjun Krishna, Maria Velez-Rojas, Steve Greenspan, Serge Mankovski, and Klaus Mueller. "TaskFinder: A Semantics-Based Methodology for Visualization Task Recommendation." Analytics 3, 3 (2024): 255-275.

TaskFinder Workflow



In and output

- User provides tabular data and a set of domain-related documents
- System recommends a visualization for each attribute–task pair
- This results in a list of visualizations

TaskFinder Dashboard

Users can select the tasks of interest

Users can also select the visualizations they are familiar with

Explainable AI

Visualization Techniques for Explainability

Visualization helps reveal how models make decisions

Feature importance visualization

- which variables most influence predictions

Saliency maps

- highlight regions of an image influencing predictions

Partial dependence plots

- show how predictions change when a feature varies

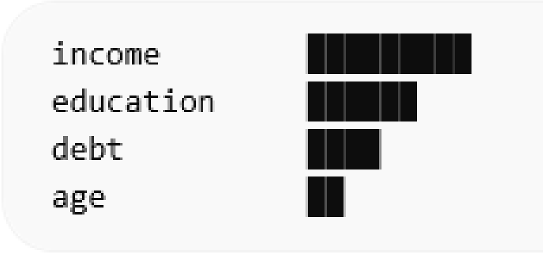
Causal models

- explain the path of reasoning

Feature importance visualization

A key question in ML and AI is:

- Which features most influence the model's prediction?



Feature importance methods estimate how strongly each feature affects the model output – allow analysts to:

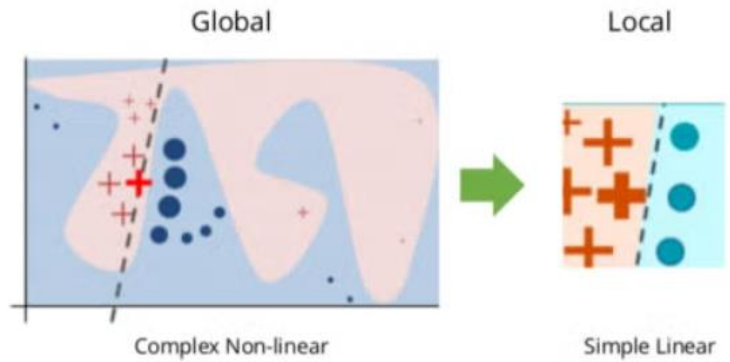
- understand model behavior
- identify influential variables
- detect unexpected relationships
- debug model predictions

Several techniques exist – see next

LIME: Local Interpretable Model-Agnostic Explanations

Core idea

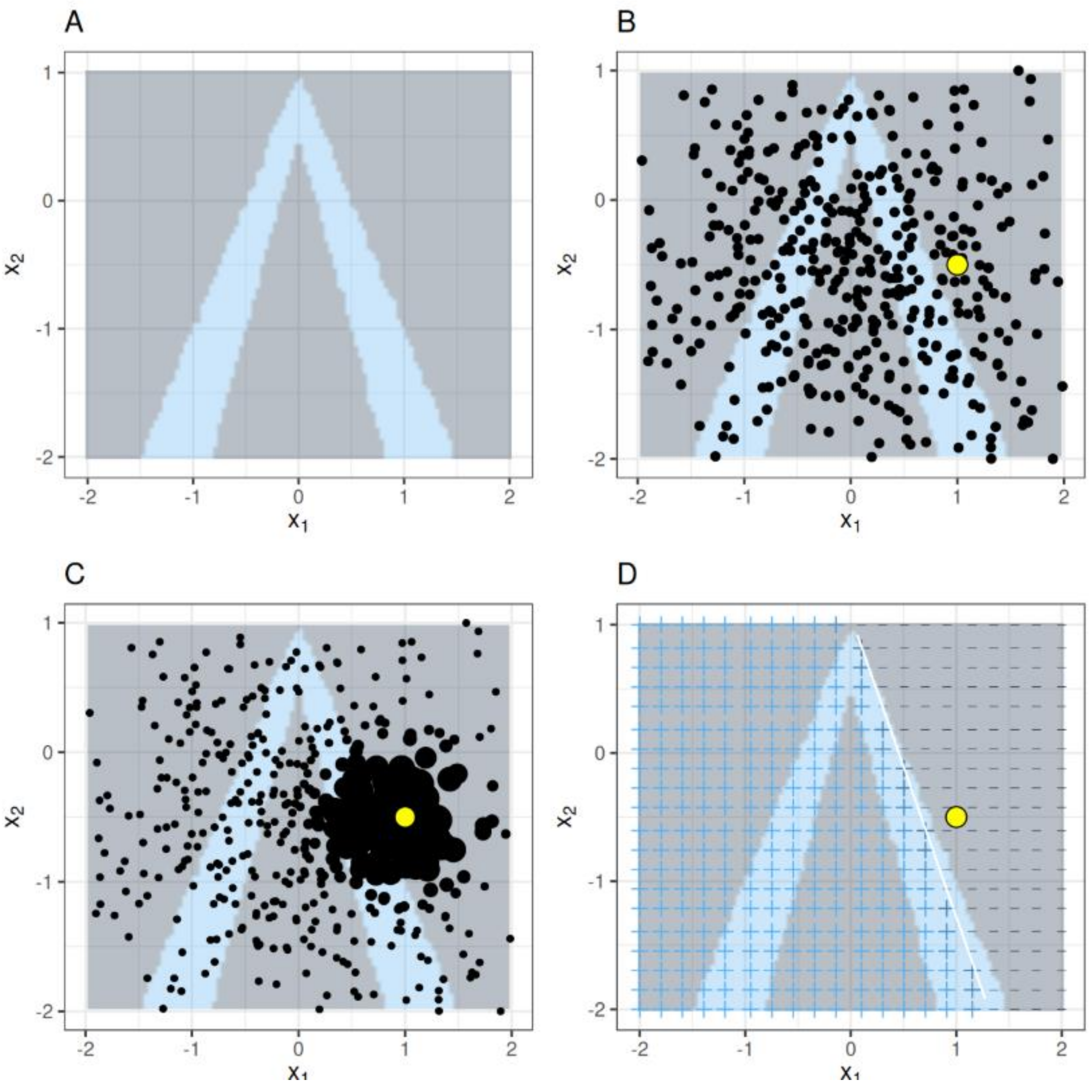
- Explain one specific prediction by approximating the model locally with a simple interpretable model
- Can explain any sampleable black box model



Method

- Select the instance to explain
- Create perturbed versions of the input
- Observe how predictions change
- Fit a simple model (often linear) near that instance

LIME: Detailed Explanation



LIME algorithm for tabular data.

A) Prediction surface given features x_1 and x_2. Predicted classes: 1 (dark) or 0 (light).

B) Instance of interest (big yellow dot) and sampled data (small black dots).

C) Assign weights based on distance to instance.

D) Signs (+/-) show the classifications of the locally learned model from the weighted samples. The white line marks the decision boundary (P(c=1) = 0.5).

SHAP: Shapley Additive Explanations

SHAP explains predictions by assigning contribution values to each feature based on Shapley values from game theory

- prediction = baseline + contribution(feature1) + contribution(feature2) + contribution(feature3)….

Algorithm

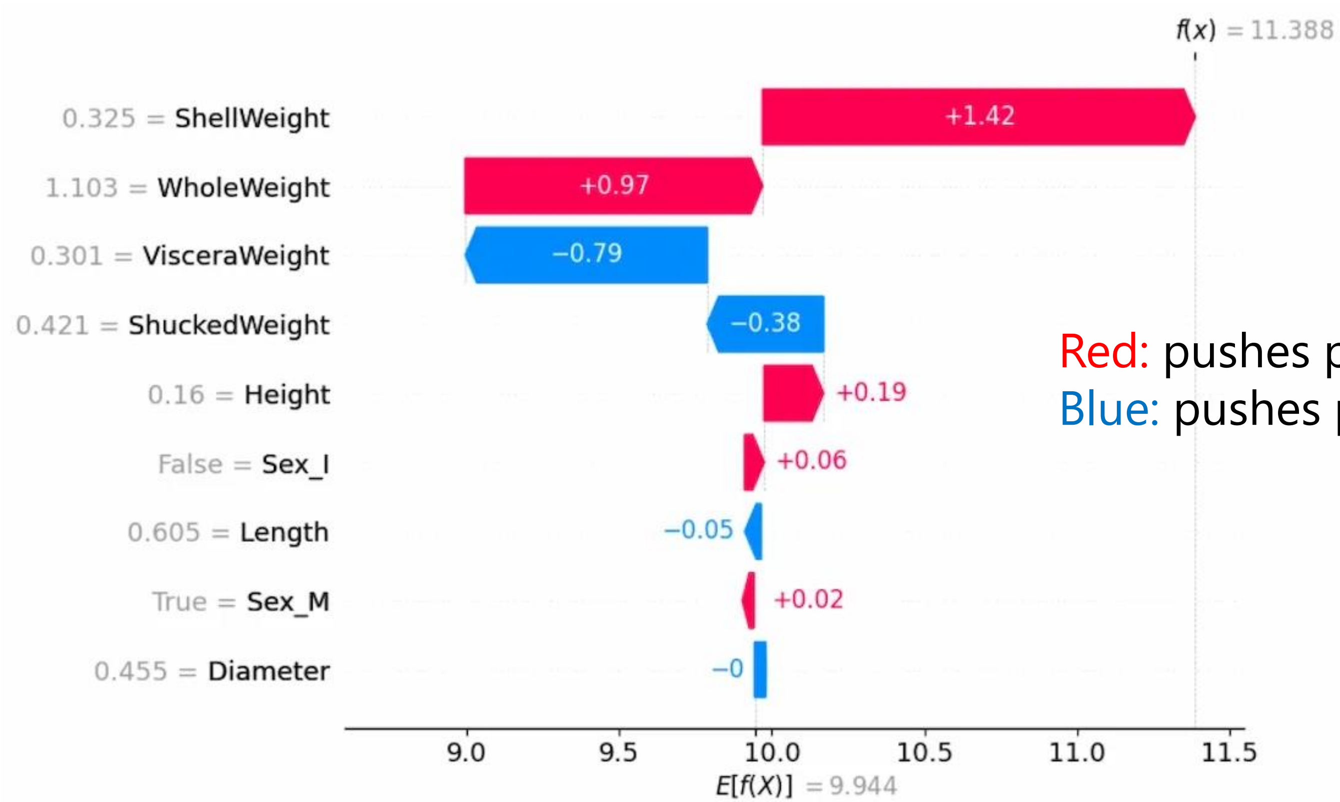
- Consider all possible subsets of features
- For each subset S:
  - compute the model prediction using only features in S
- Add feature i to subset S
- Measure the change in prediction
- Average this contribution across all subsets

Each feature receives a Shapley value representing its contribution

SHAP Waterfall Model

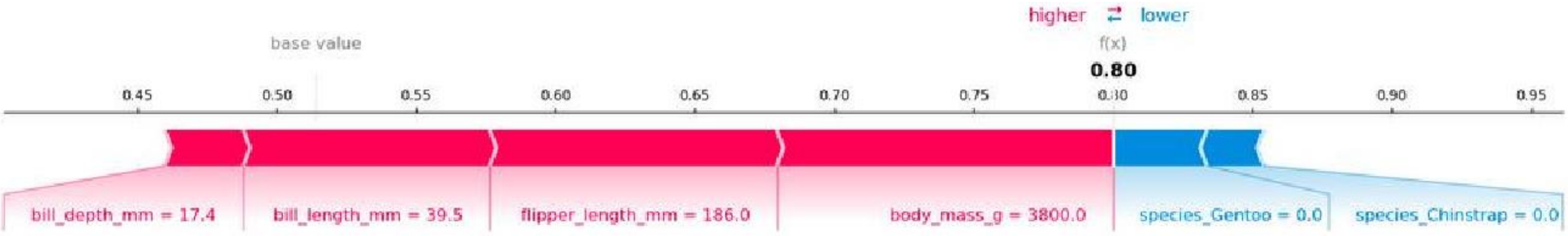
Local explanations

- Shows how each feature contributes to the difference between the model's base value and output prediction for **a specific instance**



Red: pushes prediction higher
Blue: pushes prediction lower

Alternative View



Global Model Explanations with SHAP

Compute SHAP values for many data points, then aggregate

Aggregating SHAP values reveals

- which features are most important overall
- how features influence predictions
- how effects vary across data points
- This allows analysts to study global model behavior

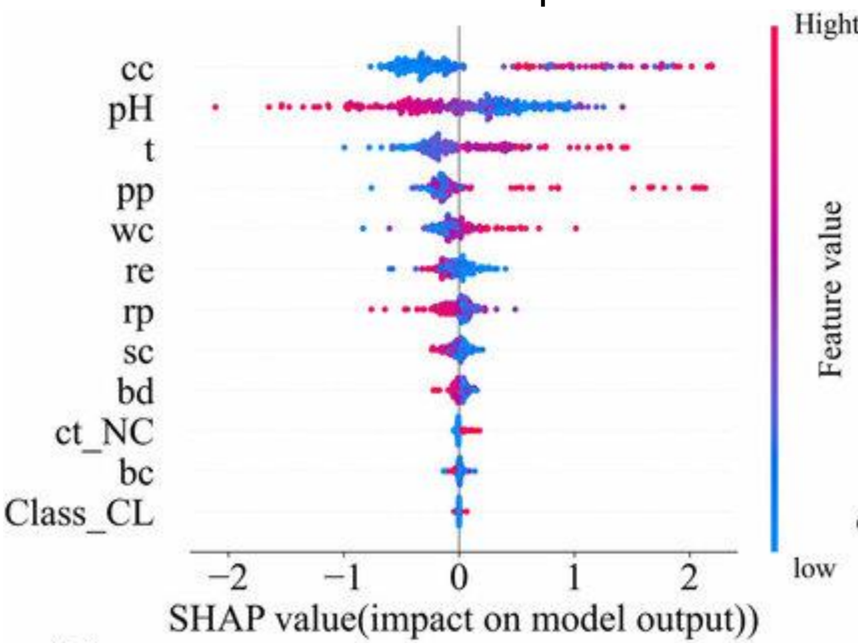
For each feature *i* compute the importance as the average absolute SHAP values

- For *j* can use the training instances x_j

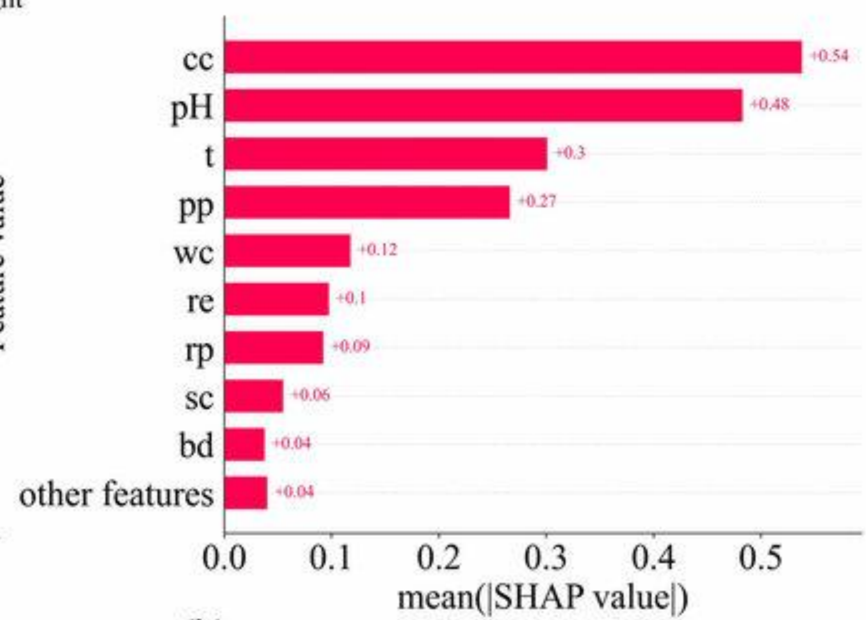
$$Importance(i) = \frac{1}{N}\sum_{j=1}^{N} |SHAP_i^{(j)}|$$

SHAP Global Model

SHAP Beeswarm plot



SHAP Global Feature Importance Plot



Horizontal Position = Impact on Prediction (the x-axis is the SHAP value)

- Right side (> 0) → feature increases the model prediction
- Left side (< 0) → feature decreases the model prediction

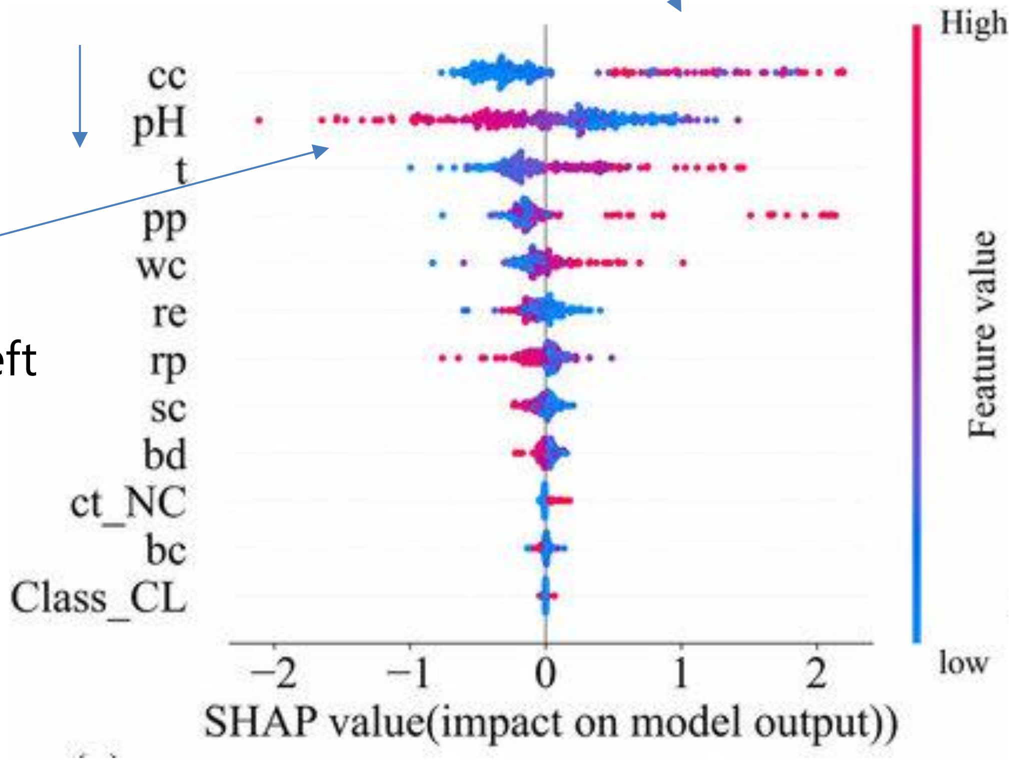
Color encodes the actual value of the feature

- Red: high feature value
- Blue: low feature value

The SHAP Beeswarm Plot

Red points appear mostly on the right
→ high values of **cc** tend to increase the prediction

cc is the most important
feature, followed by pH

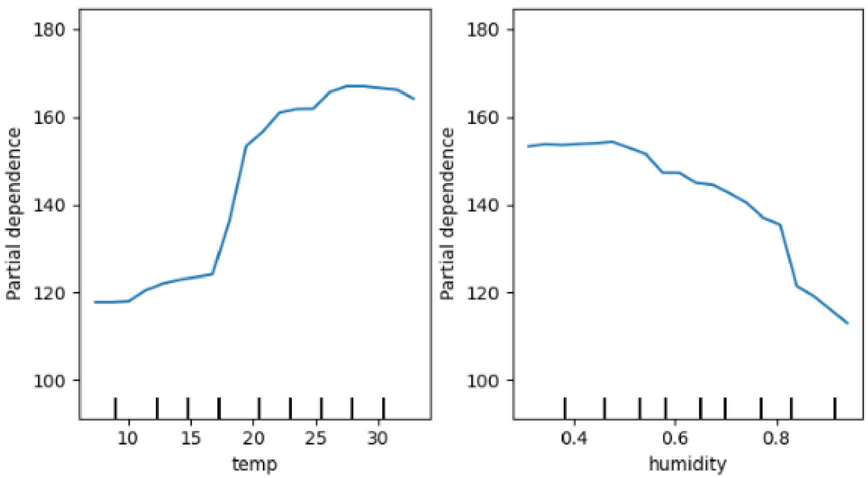


Red points appear mostly on the left
→ high values of **pH** tend to
decrease the prediction

Partial Dependence Plots (PDP)

What question do they answer?

- How does the model prediction change when a specific feature varies?
- Partial dependence plots show the average effect of one feature on the model prediction



Method

for feature x(i):

- vary x(i) → set it to different value within the range (say 10, 20, …)
- keep other features fixed
- average the predictions so obtained

Causal Explanations: Outcome Explorer

Visualizes how the outcome varies when changing a variable

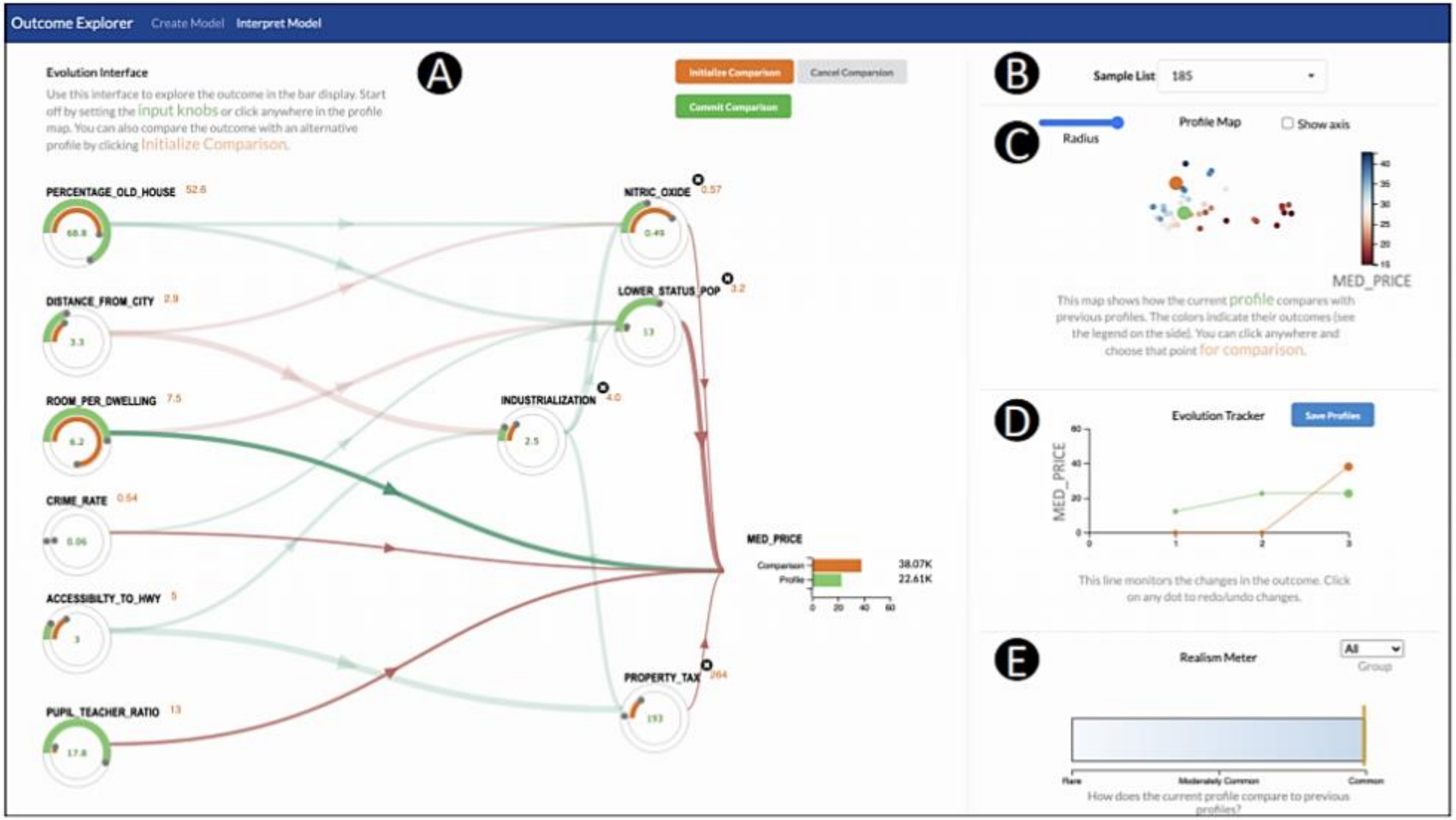


Figure Narration

A) Interactive causal DAG shows causal relations between variables. Each node includes two circular knobs (green and orange) to facilitate profile comparisons. The edge thickness and color depict the effect size and type of each edge.

B) Sample selection panel.

C) A biplot shows the position of green and orange profiles compared to nearest neighbors.

D) A line chart tracks the model outcome and to go back and forth between feature configurations.

E) Realism meter allows users to determine how common a profile is compared to other samples in the dataset.

Outcome Explorer Video

https://www.youtube.com/watch?v=ot4h2cXFhe8

Hoque, Md Naimul, and Klaus Mueller. "Outcome-explorer: A causality guided interactive visual interface for interpretable algorithmic decision making." *IEEE Transactions on Visualization and Computer Graphics* 28.12 (2021): 4728-4740.

Emerging Directions

Emerging Directions

LLM-assisted data analysis (NL4DV [TVCG 2020])

- Natural language interfaces that generate and refine visualizations

AI copilots for visualization (current systems ++)

- Systems that suggest insights, charts, and analytical workflows

Visualization of AI reasoning (current systems ++)

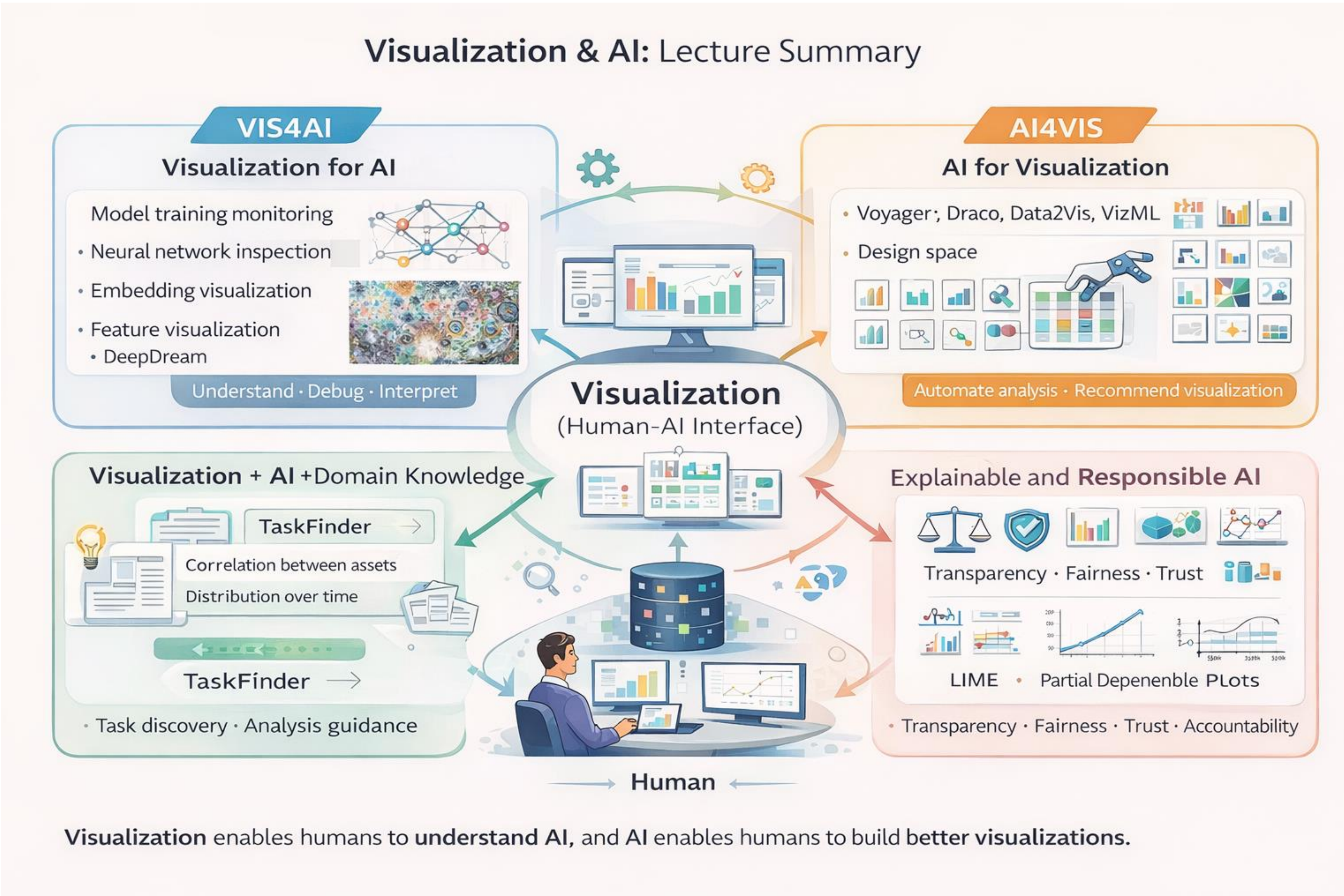
- Tools for exploring model behavior, agent decisions, and reasoning traces

Interactive human–AI collaboration (next lecture)

- Analysts guiding AI systems through interactive visual interfaces

Responsible and trustworthy AI (Silva [CHI, 2020], De-Bias [TVCG 2023])

- Visualization for fairness analysis, bias detection, and model accountability


Visualization & AI: Lecture Summary
VIS4AI
Visualization for AI
Model training monitoring
• Neural network inspection
• Embedding visualization
• Feature visualization
• DeepDream
Understand · Debug · Interpret
AI4VIS
AI for Visualization
• Voyager:, Draco, Data2Vis, VizML
• Design space
Automate analysis · Recommend visualization
Visualization
(Human-AI Interface)
Visualization + AI +Domain Knowledge
TaskFinder
Correlation between assets
Distribution over time
TaskFinder
· Task discovery · Analysis guidance
Explainable and Responsible AI
Transparency · Fairness · Trust
LIME · Partial Depenenble PLots
· Transparency · Fairness · Trust · Accountability
Human
Visualization enables humans to understand AI, and AI enables humans to build better visualizations.