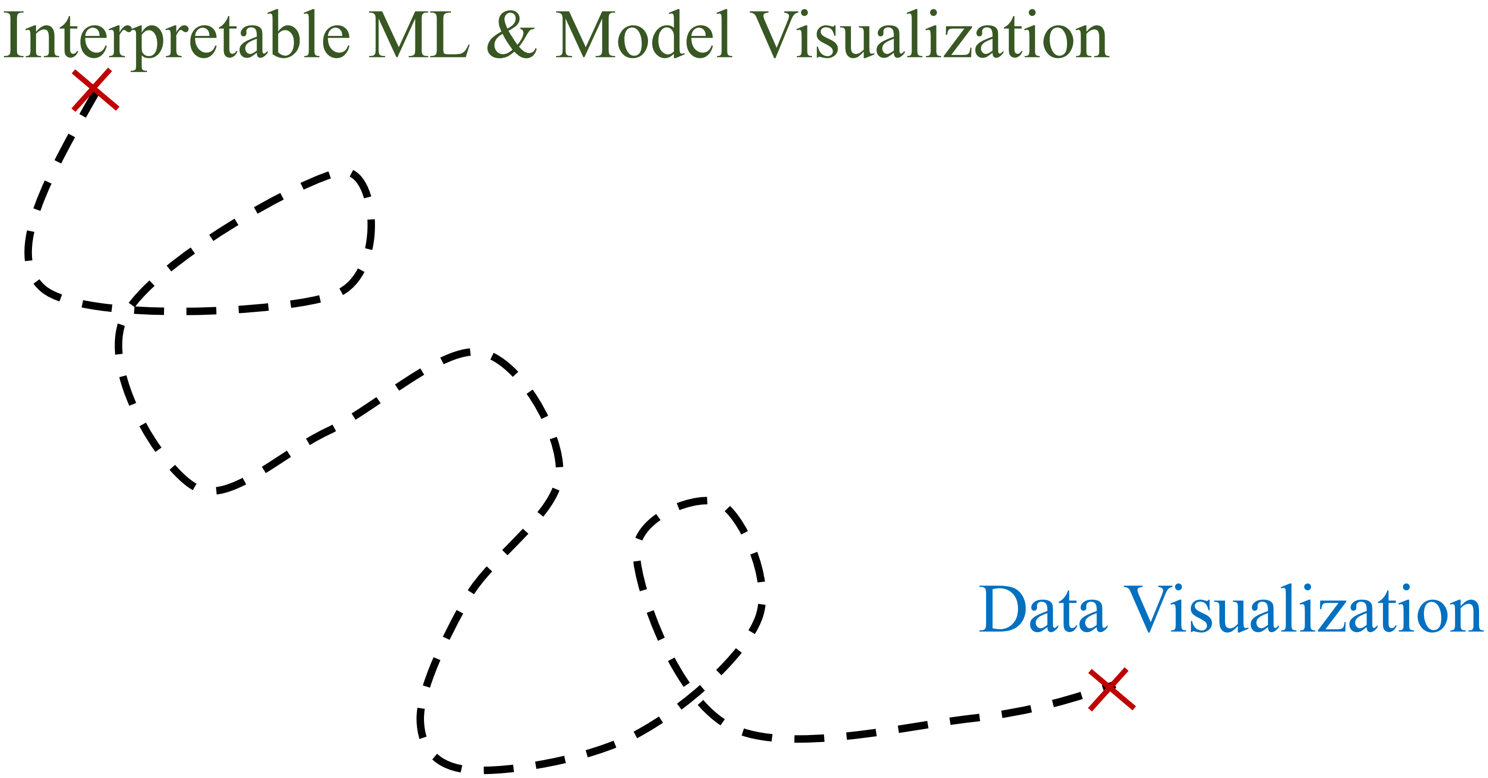
Understanding Data and Models Through Visualization

Cynthia Rudin

Professor of Computer Science, Electrical and Computer Engineering, and Statistical Science

Duke University

Outline


Interpretable ML & Model Visualization
Data Visualization

Interpretable Machine Learning

- An interpretable machine learning model obeys a domain-specific set of constraints.
- My technical definition: An interpretable machine learning model is constrained in model form so that it is either useful to someone, or obeys structural knowledge of the domain, such as monotonicity, causality, structural (generative) constraints, additivity, or physical constraints that come from domain knowledge.
- There’s a spectrum.

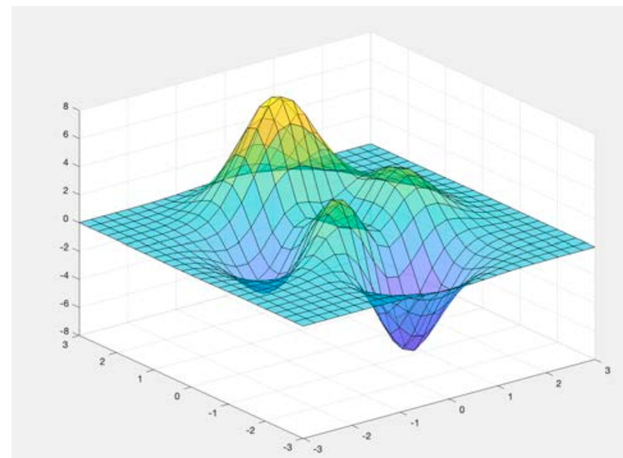
What is the difference between an “interpretable model” and an “explanation model”?

- Interpretable models don’t need to be “explained.”

The 2HELPS2B score
(predicts seizures in critically ill patients, Struck et al 2017)

1.	Any cEEG Pattern with Frequency **2 H**z	1 point	· · ·
2.	**E**pileptiform Discharges	1 point	+ · · ·
3.	Patterns include [**L**PD, LRDA, BIPD]	1 point	+ · · ·
4.	**P**atterns Superimposed with Fast or Sharp Activity	1 point	+ · · ·
5.	Prior **S**eizure	1 point	+ · · ·
6.	**B**rief Rhythmic Discharges	**2** points	+ · · ·
		SCORE	= · · ·

SCORE	0	1	2	3	4	5	6+
RISK	<5%	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

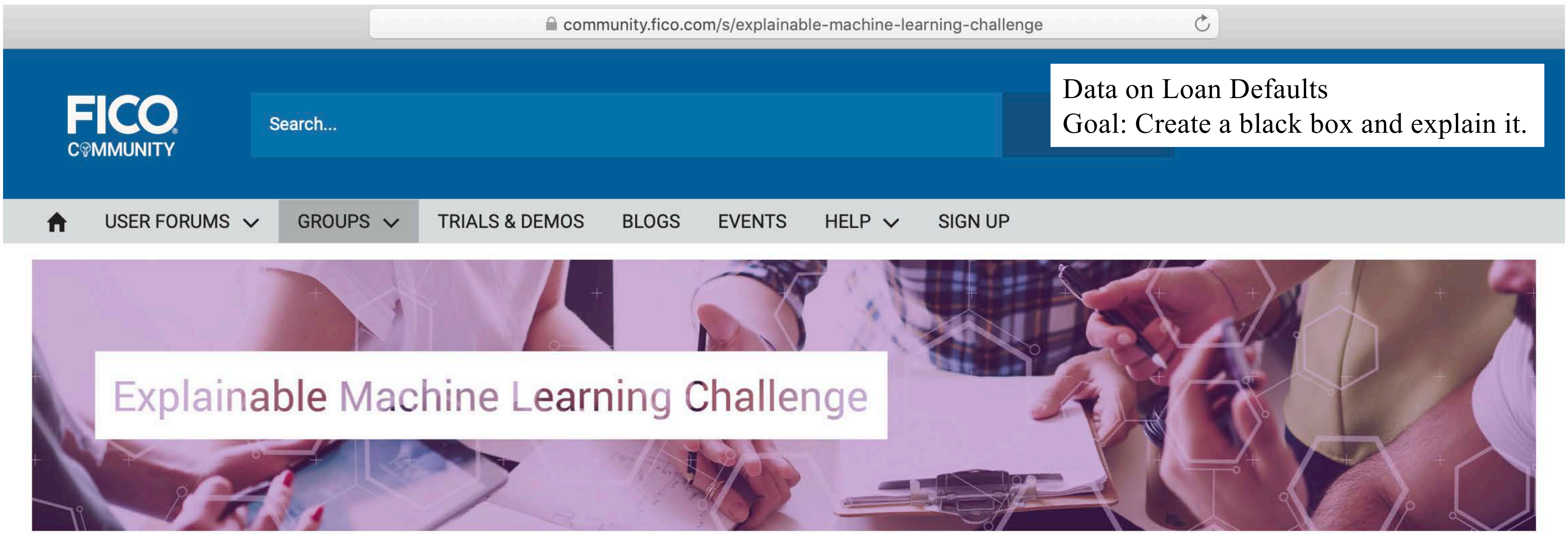


- An explanation model is often an approximation of a more complex model.

- If you can produce an interpretable model, why explain a black box?

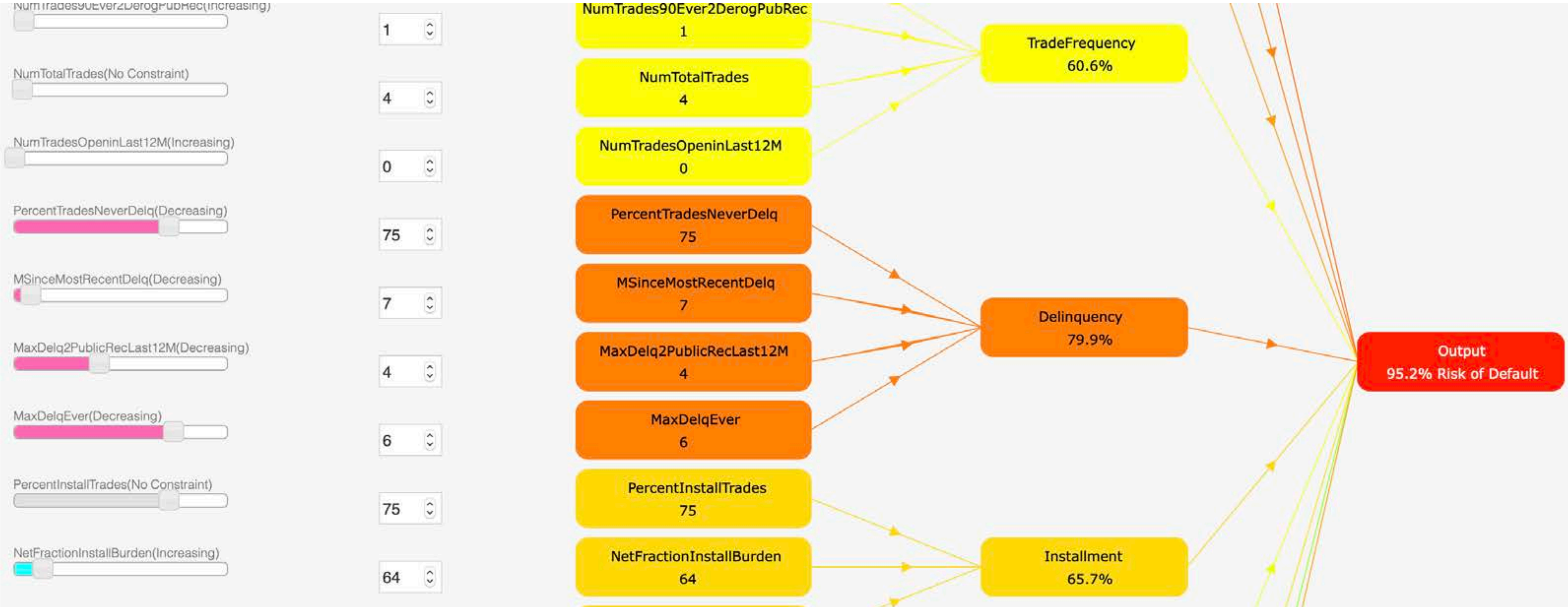
- Models do not need to be “simple” to be interpretable.

FICO Explainable ML Challenge 2018

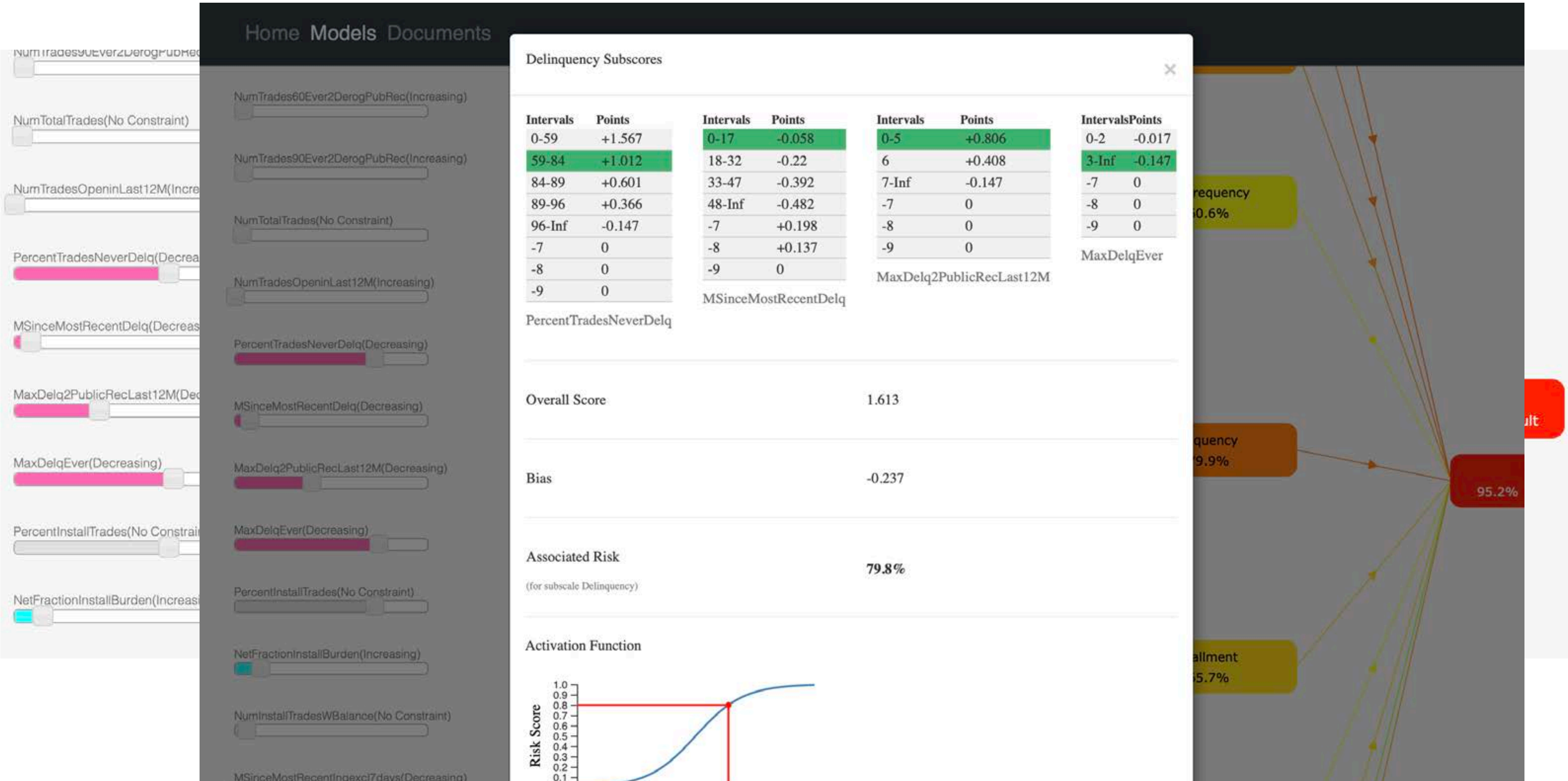


Data on Loan Defaults
Goal: Create a black box and explain it.

Play around with our entry at http://dukedatasciencefico.cs.duke.edu



Play around with our entry at http://dukedatasciencefico.cs.duke.edu



Play around with our entry at http://dukedatasciencefico.cs.duke.edu

Delinquency Subscores

Intervals	Points
0-59	+1.567
59-84	+1.012
84-89	+0.601
89-96	+0.366
96-Inf	-0.147
-7	0
-8	0
-9	0

PercentTradesNeverDelq

Intervals	Points
0-17	-0.058
18-32	-0.22
33-47	-0.392
48-Inf	-0.482
-7	+0.198
-8	+0.137
-9	0

MSinceMostRecentDelq

Intervals	Points
0-5	+0.806
6	+0.408
7-Inf	-0.147
-7	0
-8	0
-9	0

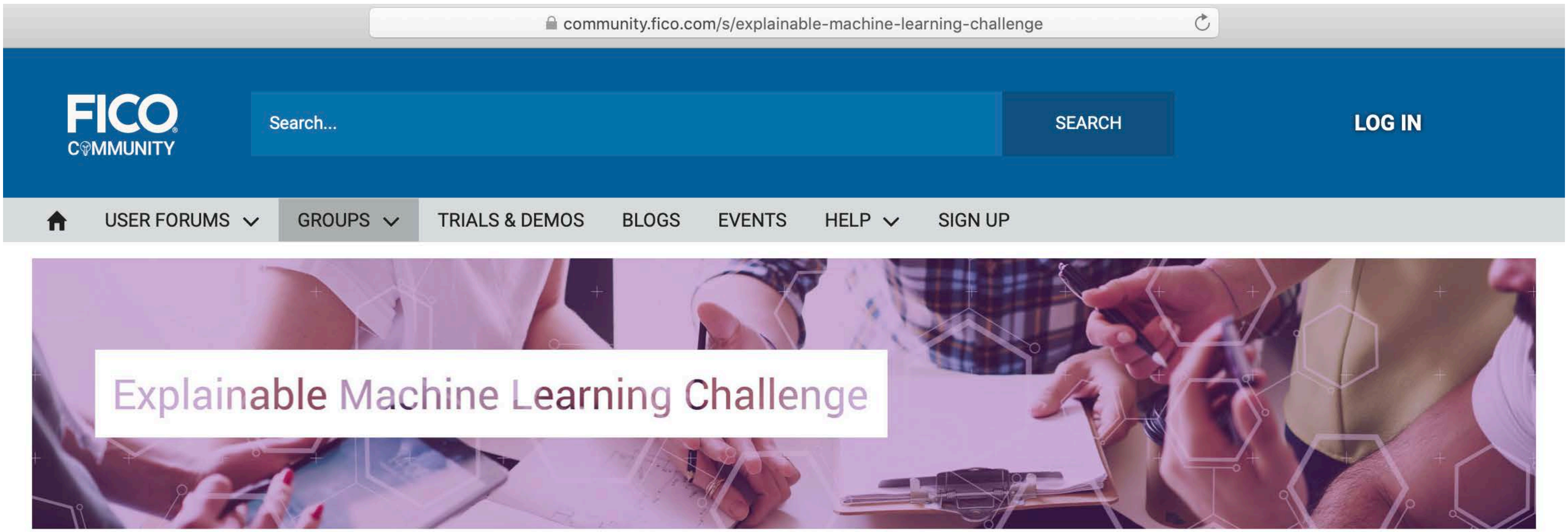
MaxDelq2PublicRecLast12M

Intervals	Points
0-2	-0.017
3-Inf	-0.147
-7	0
-8	0
-9	0

MaxDelqEver

Play around with our entry at http://dukedatasciencefico.cs.duke.edu

FICO ML Challenge



Play around with our entry at http://dukedatasciencefico.cs.duke.edu

Results

The team representing Duke University, which included Chaofan Chen, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang and Tong Wang, received the FICO Recognition Award acknowledging their submission for going above and beyond expectations with a fully transparent global model and a user-friendly dashboard to allow users to explore the global model and its explanations.

Dear XXX@Stanford.edu,

<blah>… We don't know whether our paper fits into the scope of the special issue. It's not a traditional methodology paper, it's contribution is an analysis of the FICO data, including a machine learning model that is interpretable, ... The paper's content actually won the FICO Challenge Award for the first Explainable Machine Learning Challenge.

Response:

Dear Cynthia,

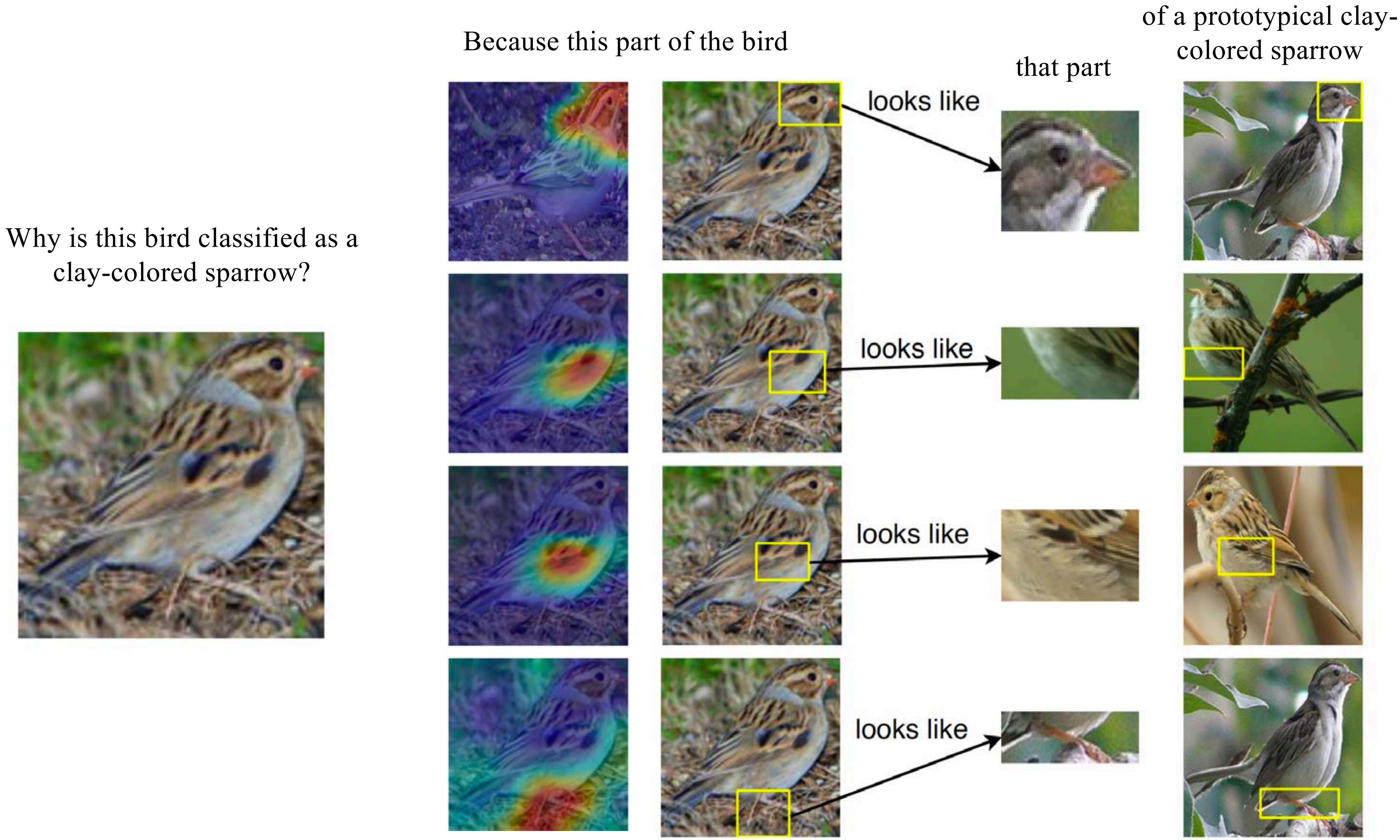
Thanks for reaching out. This is an interesting paper… . But I'm afraid its not a good fit for the special issue. … Its also related to my own recent work on explainability of neural nets. … Is the FICO data still available? If so, could you share it?

My (silent) response: Whoa, can I frame this ridiculous email?

Understanding Data and Models Through Visualization

- Point 1: Interpretable models don’t need to be simple or small…
  if you can co-design interpretable models with their visualizations

An interpretable deep neural network?


Why is this bird classified as a clay-colored sparrow?
Because this part of the bird
looks like
that part
of a prototypical clay-colored sparrow
looks like
looks like
looks like

“*This* Looks Like *That:* deep learning for interpretable image recognition”

NeurIPS 2019 (spotlight)

arXiv.org > cs > arXiv:1806.10574

Search...

Help | Advanced S

Computer Science > Machine Learning

[Submitted on 27 Jun 2018 (v1), last revised 28 Dec 2019 (this version, v5)]

This Looks Like That: Deep Learning for Interpretable Image Recognition

Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, Cynthia Rudin

When we are faced with challenging image classification tasks, we often explain our reasoning by dissecting the image, and pointing out prototypical aspects of one class or another. The mounting evidence for each of the classes helps us make our final decision. In this work, we introduce a deep network architecture -- prototypical part network (ProtoPNet), that reasons in a similar way: the network dissects the image by finding prototypical parts, and combines evidence from the prototypes to make a final classification. The model thus reasons in a way that is qualitatively similar to the way ornithologists, physicians, and others would explain to people on how to solve challenging image

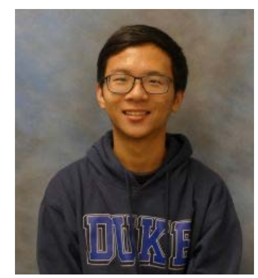
- Adds a “prototype” layer to a black box, forces the network to do case-based reasoning.
- Prototypes are learned during training.

Chaofan

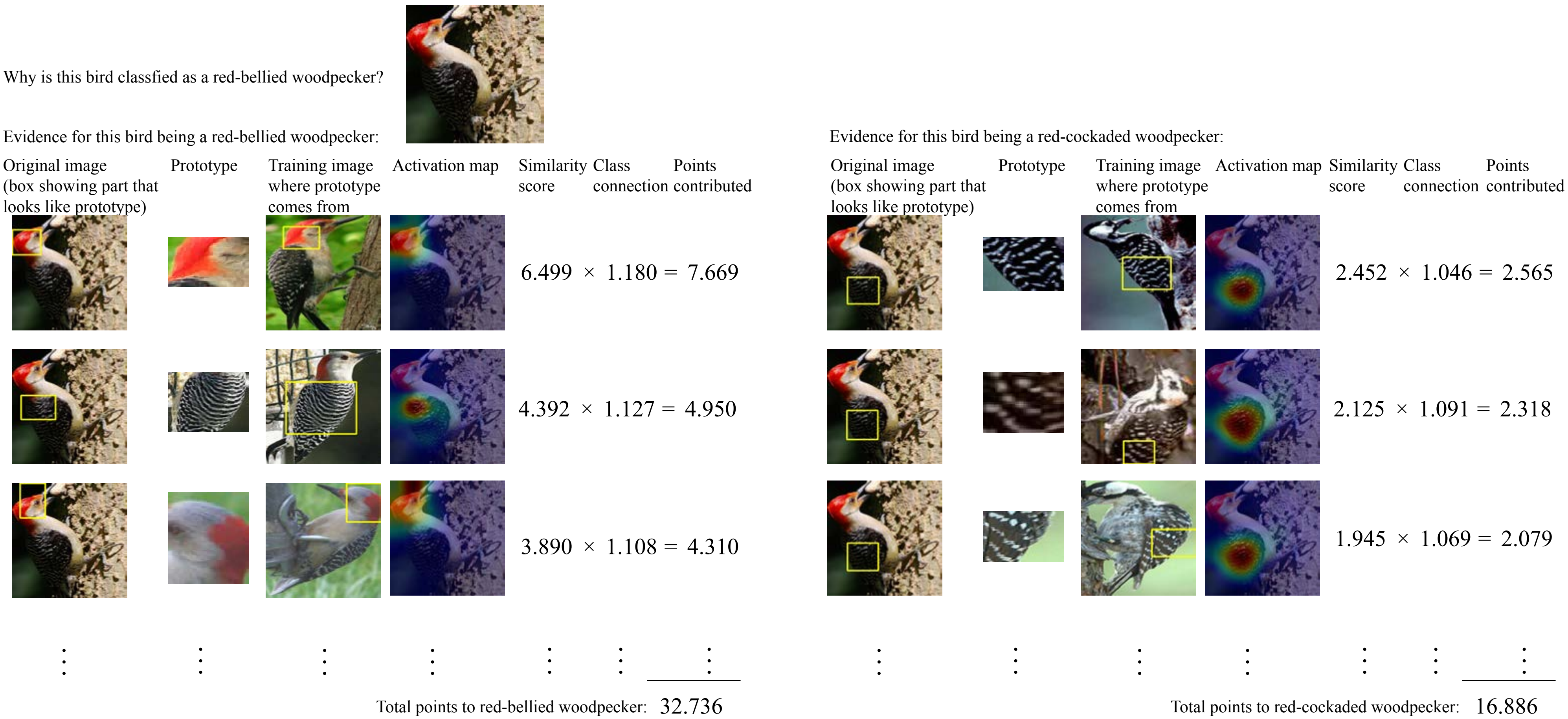
Oscar

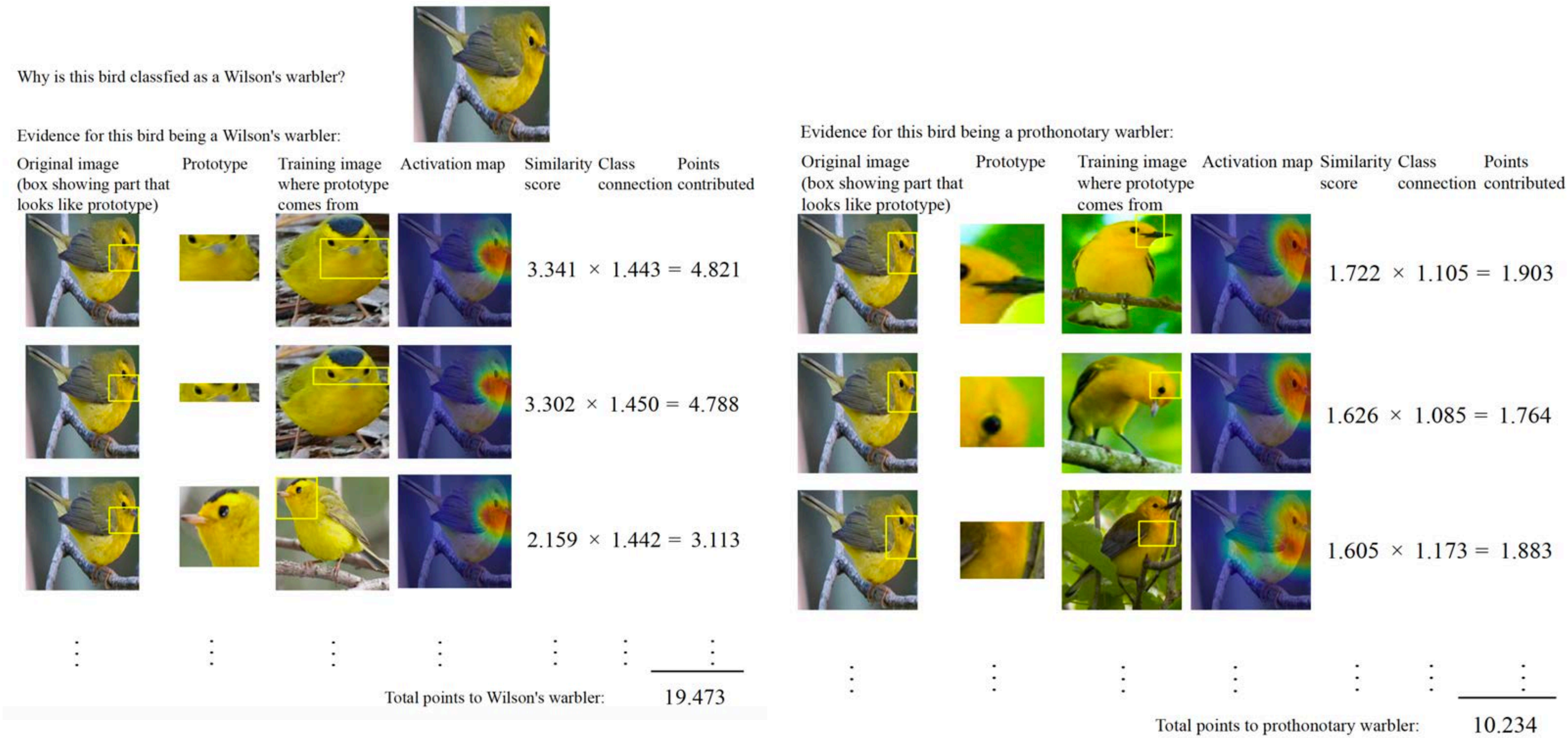
Alina

Jonathan

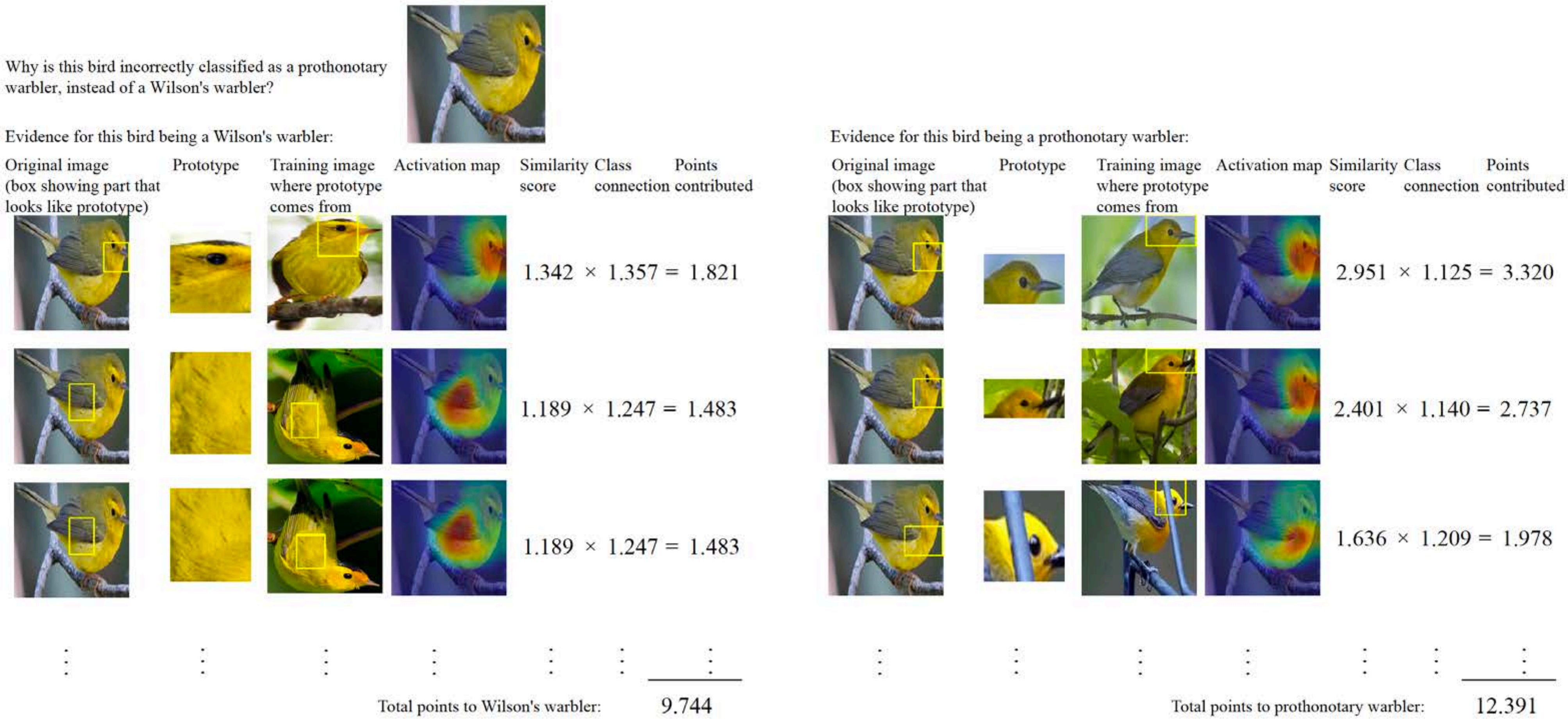


Daniel


Why is this bird classfied as a red-bellied woodpecker?
Evidence for this bird being a red-bellied woodpecker:
Original image (box showing part that looks like prototype)
Prototype
Training image where prototype comes from
Activation map
Similarity score
Class connection
Points contributed
6.499 × 1.180 = 7.669
4.392 × 1.127 = 4.950
3.890 × 1.108 = 4.310
Total points to red-bellied woodpecker: 32.736
Evidence for this bird being a red-cockaded woodpecker:
Original image (box showing part that looks like prototype)
Prototype
Training image where prototype comes from
Activation map
Similarity score
Class connection
Points contributed
2.452 × 1.046 = 2.565
2.125 × 1.091 = 2.318
1.945 × 1.069 = 2.079
Total points to red-cockaded woodpecker: 16.886



Base model: VGG-16



Base model: DenseNet161

CUB-200

- 200 classes of birds
- Original black box accuracy between 74.6% (VGG16) and 82.3% (Res34).
- Interpretable model's accuracy between 76.1% (VGG) and 80.2% (Dense121). Combining several interpretable networks together yields 84.8%, and still yields an interpretable model.

So even for computer vision, we can still have an interpretable model of the same accuracy as a black box.

Understanding Data and Models Through Visualization

- Point 1: Interpretable models don't need to be simple or small… if you can co-design interpretable models with their visualizations

- Point 2: When using data visualization tools, it's useful to have an understanding of their strengths and weaknesses

Visualizing data using t-SNE
L Maaten, G **Hinton** - Journal of machine learning research, 2008 - jmlr.org
We present a new technique called" t-SNE" that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002) that is much easier to optimize …
Cited by 13929 Related articles All 44 versions

t-SNE is a dimension reduction algorithm.

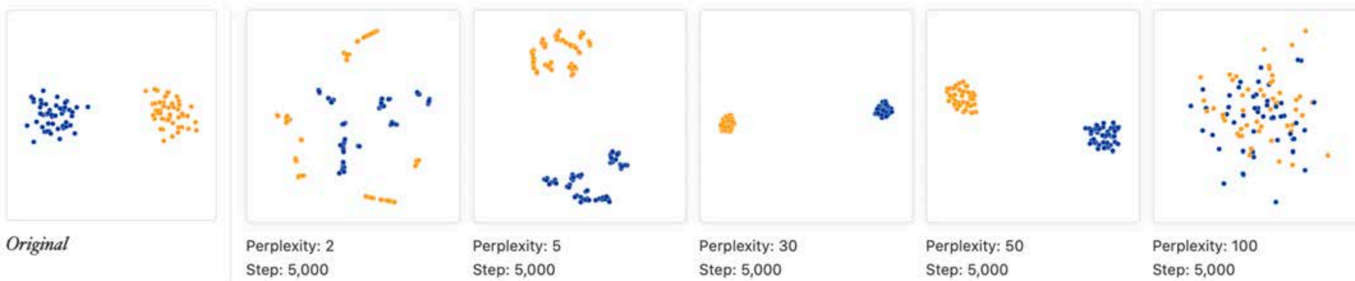
Input: high-dimensional data
Output: low-dimensional data that preserves…
- the graph structure?
- local neighborhoods?
- global structure?

Visualizing data using t-SNE
L Maaten, G **Hinton** - Journal of machine learning research, 2008 - jmlr.org
We present a new technique called" t-SNE" that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002) that is much easier to optimize …
☆ 99 Cited by 13929 Related articles All 44 versions »

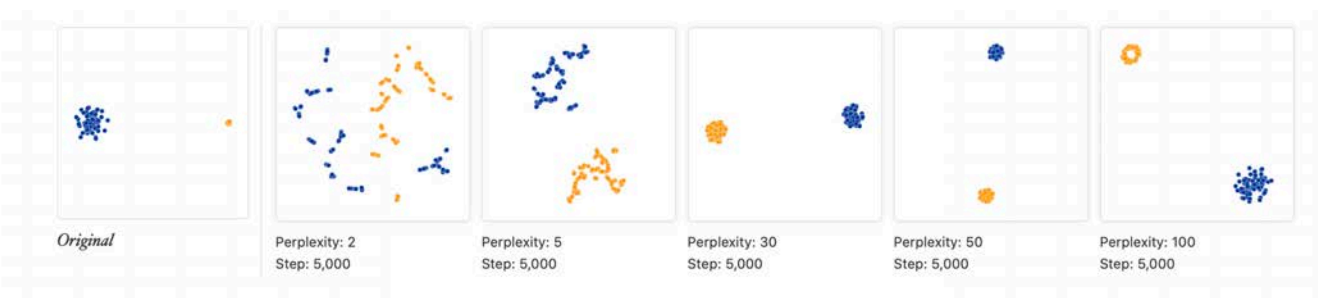
How to Use t-SNE Effectively

MARTIN WATTENBERG Google Brain
FERNANDA VIÉGAS Google Brain
IAN JOHNSON Google Cloud
Oct. 13 2016

1. Those hyperparameters really matter



2. Cluster sizes in a t-SNE plot mean nothing



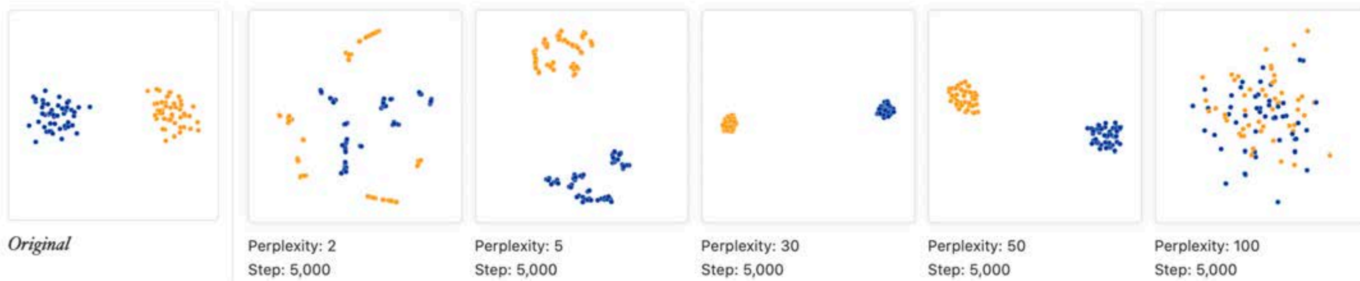
Visualizing data using t-SNE
L Maaten, G **Hinton** - Journal of machine learning research, 2008 - jmlr.org
We present a new technique called" t-SNE" that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002) that is much easier to optimize …
Cited by 13929 Related articles All 44 versions

How to Use t-SNE Effectively

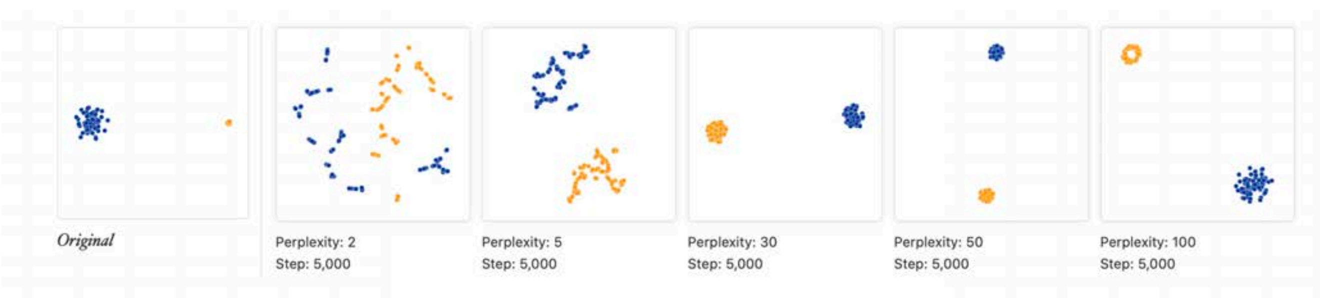
MARTIN WATTENBERG Google Brain
FERNANDA VIÉGAS Google Brain
IAN JOHNSON Google Cloud

Oct. 13 2016

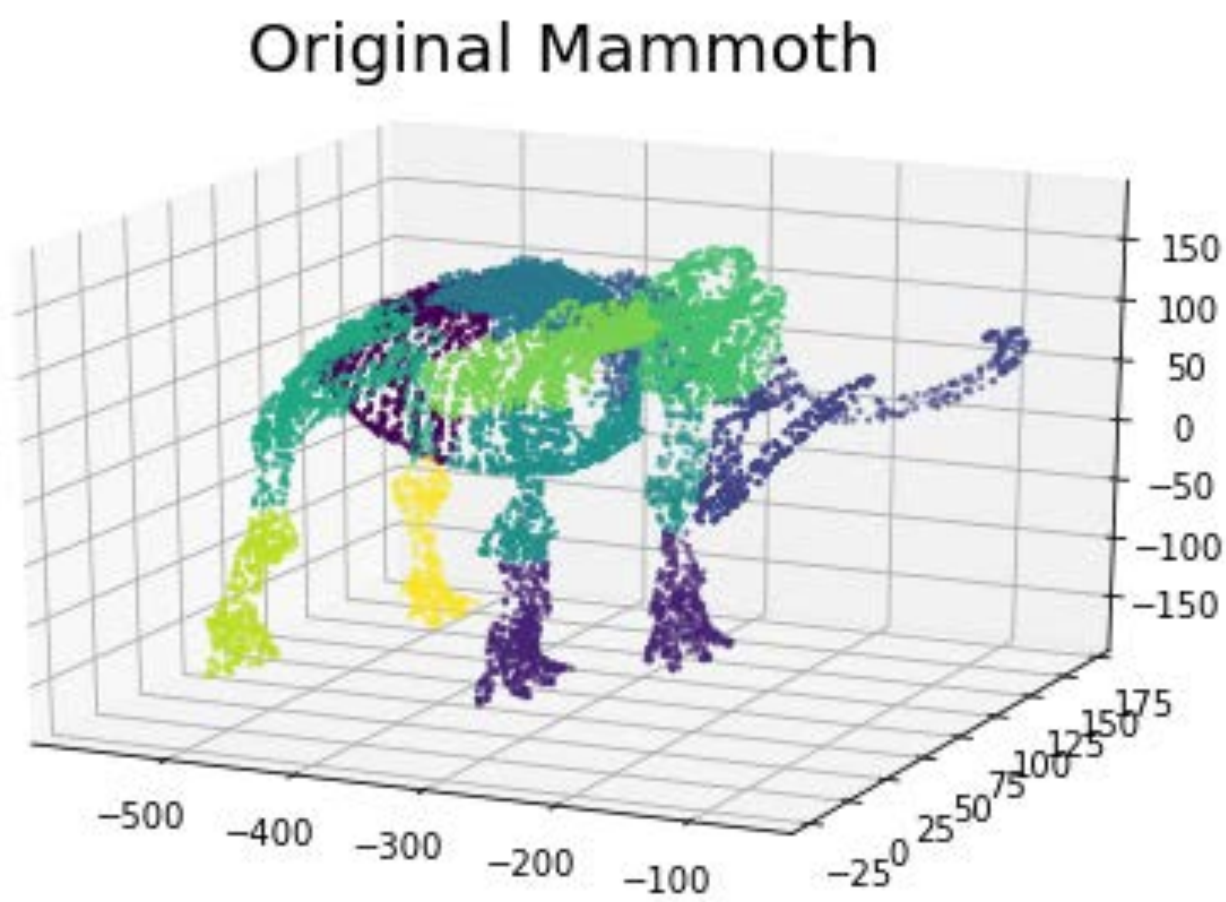
1. Those hyperparameters really matter



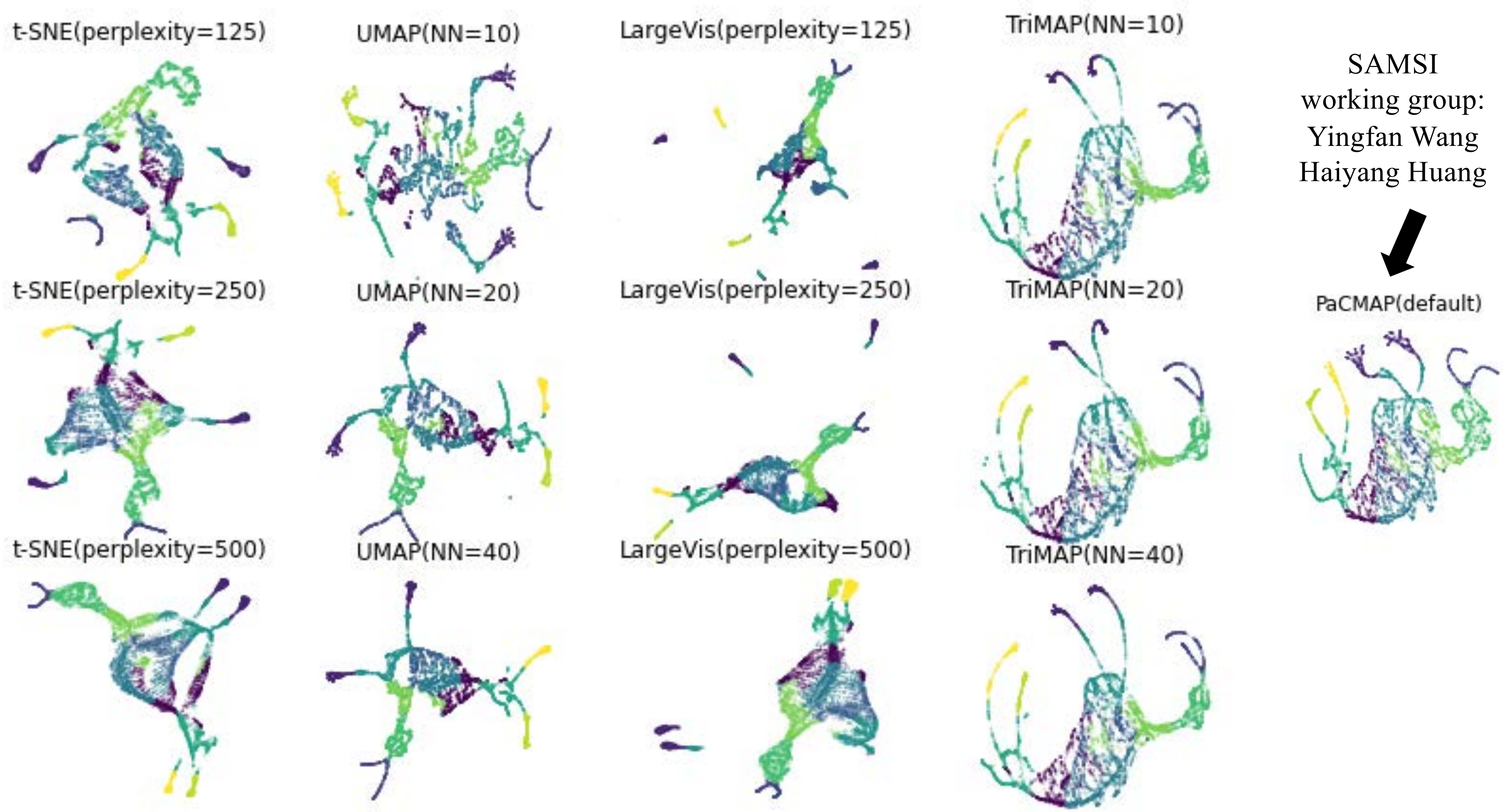
2. Cluster sizes in a t-SNE plot mean nothing

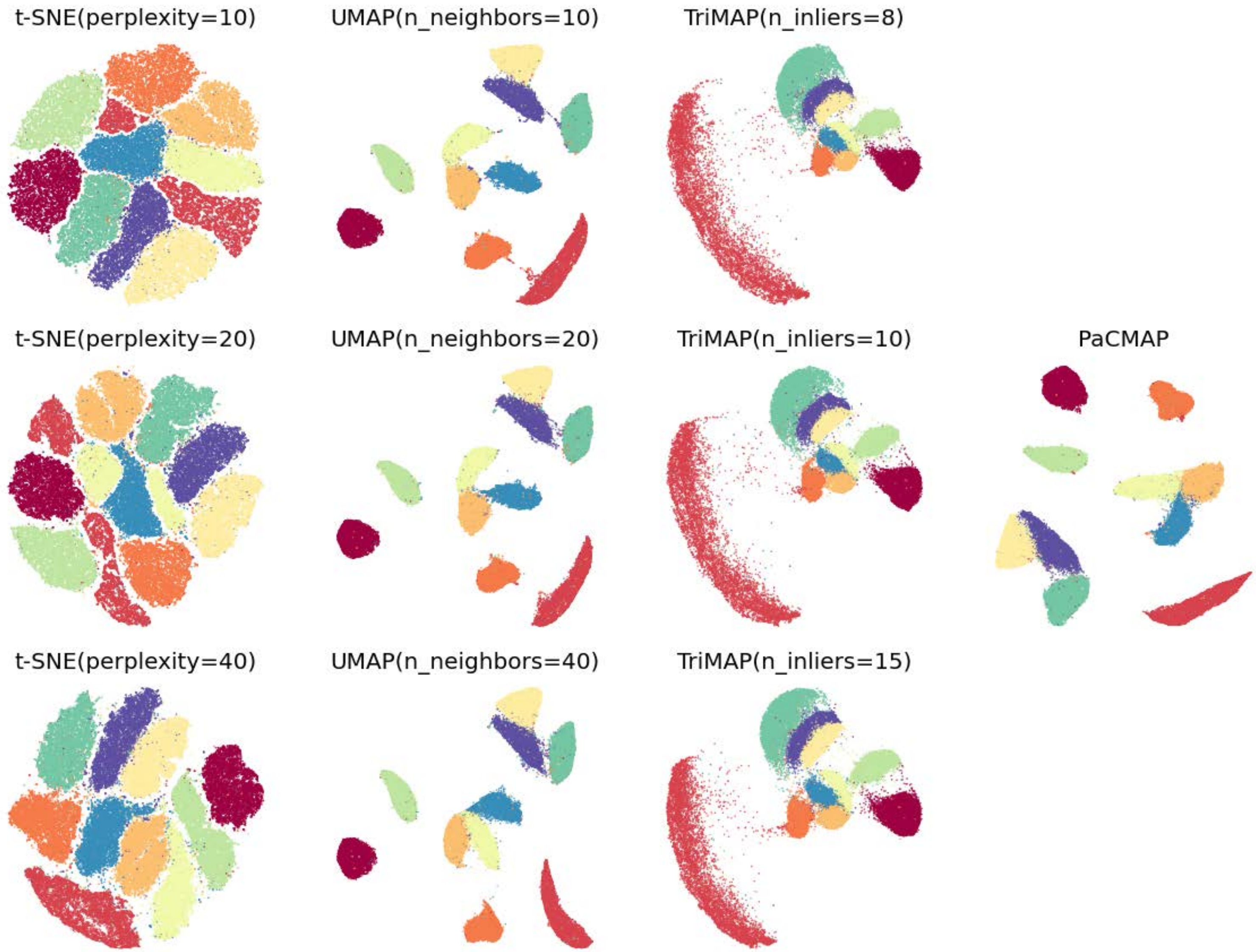


SAMSI Interpretable Deep Learning Working Group
Spring 2020



Task: 3d to 2d.
Global structure is important here!


t-SNE(perplexity=125)
UMAP(NN=10)
LargeVis(perplexity=125)
TriMAP(NN=10)
SAMSI working group: Yingfan Wang Haiyang Huang
t-SNE(perplexity=250)
UMAP(NN=20)
LargeVis(perplexity=250)
TriMAP(NN=20)
PaCMAP(default)
t-SNE(perplexity=500)
UMAP(NN=40)
LargeVis(perplexity=500)
TriMAP(NN=40)


t-SNE(perplexity=10)
UMAP(n_neighbors=10)
TriMAP(n_inliers=8)
t-SNE(perplexity=20)
UMAP(n_neighbors=20)
TriMAP(n_inliers=10)
PaCMAP
t-SNE(perplexity=40)
UMAP(n_neighbors=40)
TriMAP(n_inliers=15)

Algorithm	Graph component	Loss function
t-SNE	Edges (i, j)	$\text{Loss}^{\text{t-SNE}}_{i,j} = p_{ij} \log \frac{p_{ij}}{q_{ij}}$, where $q_{ij} = \frac{(1+\lVert \mathbf{y}_i - \mathbf{y}_j \rVert^2)^{-1}}{\sum_{k \neq l}(1+\lVert \mathbf{y}_k - y_l \rVert^2)^{-1}}$
UMAP	Edges (i, j)	$\text{Loss}^{\text{UMAP}}_{i,j} = \begin{cases} \bar{w}_{i,j} \log \left(1 + a \left(\lVert \mathbf{y}_i - \mathbf{y}_j \rVert_2^2\right)^b\right)^{-1} & i, j \text{ neighbors} \\ (1 - \bar{w}_{i,j}) \log \left(1 - \left(1 + a \left(\lVert \mathbf{y}_i - \mathbf{y}_j \rVert_2^2\right)^b\right)^{-1}\right) & \text{Otherwise} \end{cases}$
TriMAP	Triplets (i, j, k) where $\text{Distance}_{i,j} \leq \text{Distance}_{i,k}$	$\text{Loss}^{\text{TM}}_{i,j,k} = \omega_{i,j,k} \frac{s(\mathbf{y}_i, \mathbf{y}_k)}{s(\mathbf{y}_i, \mathbf{y}_j) + s(\mathbf{y}_i, \mathbf{y}_k)}$, where $s(\mathbf{y}_i, \mathbf{y}_j) = \left(1 + \lVert \mathbf{y}_i - \mathbf{y}_j \rVert^2\right)^{-1}$

UMAP (McInnes et al., 2018), TriMAP (Amid & Warmuth, 2019)

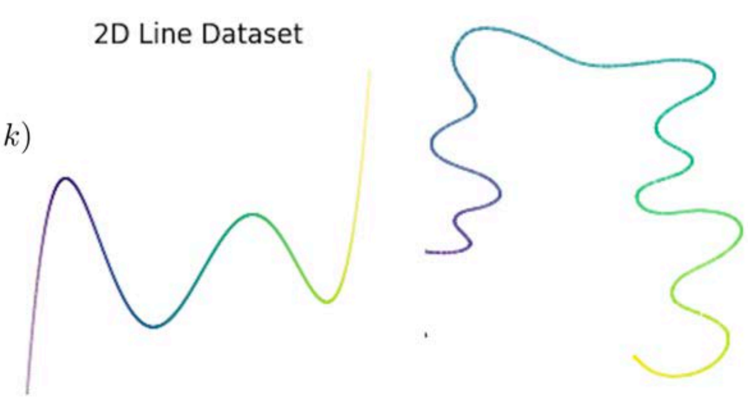
What elements of these algorithms are important?

What we knew before:

- Certain properties of the loss function are important:
  - Attraction: neighbors should be attracted. But not too close! (Crowding)
  - Repulsion: farther points in original space should be far in low-dim space.

But that's not enough…

$$\sum_{(i,j)\in\mathcal{T}_{\text{neighbors}}} l^{\text{attract}}(i,j) + \sum_{(i,k)\in\mathcal{T}_{\text{further}}} l^{\text{repulse}}(i,k)$$



What we knew before:

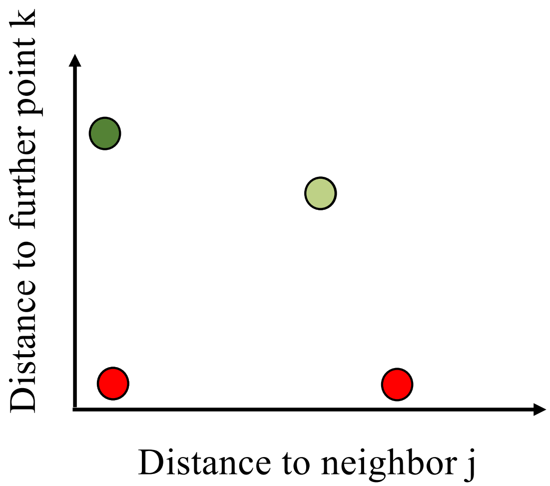
- Certain properties of the loss function are important:
  - Attraction: neighbors should be attracted. But not too close! (Crowding)
  - Repulsion: farther points in original space should be far in low-dim space.

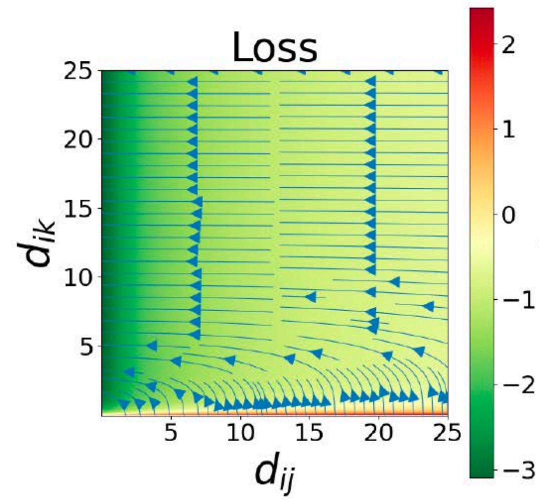
After a huge amount of experimentation, we found that:

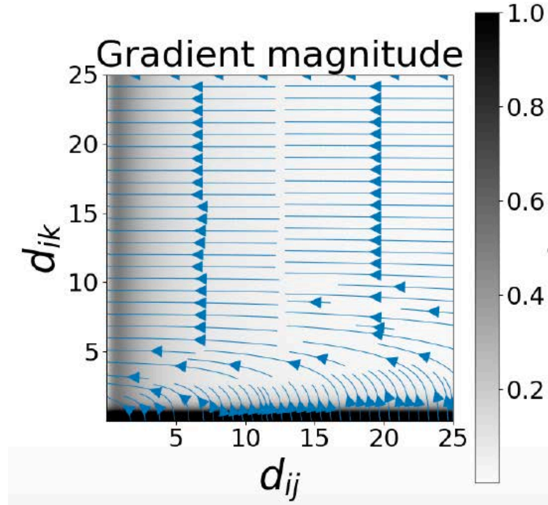
- Certain specific properties of the loss function are important.
- We must have forces on non-neighbors.
  - The choice of which graph components to preserve are important.

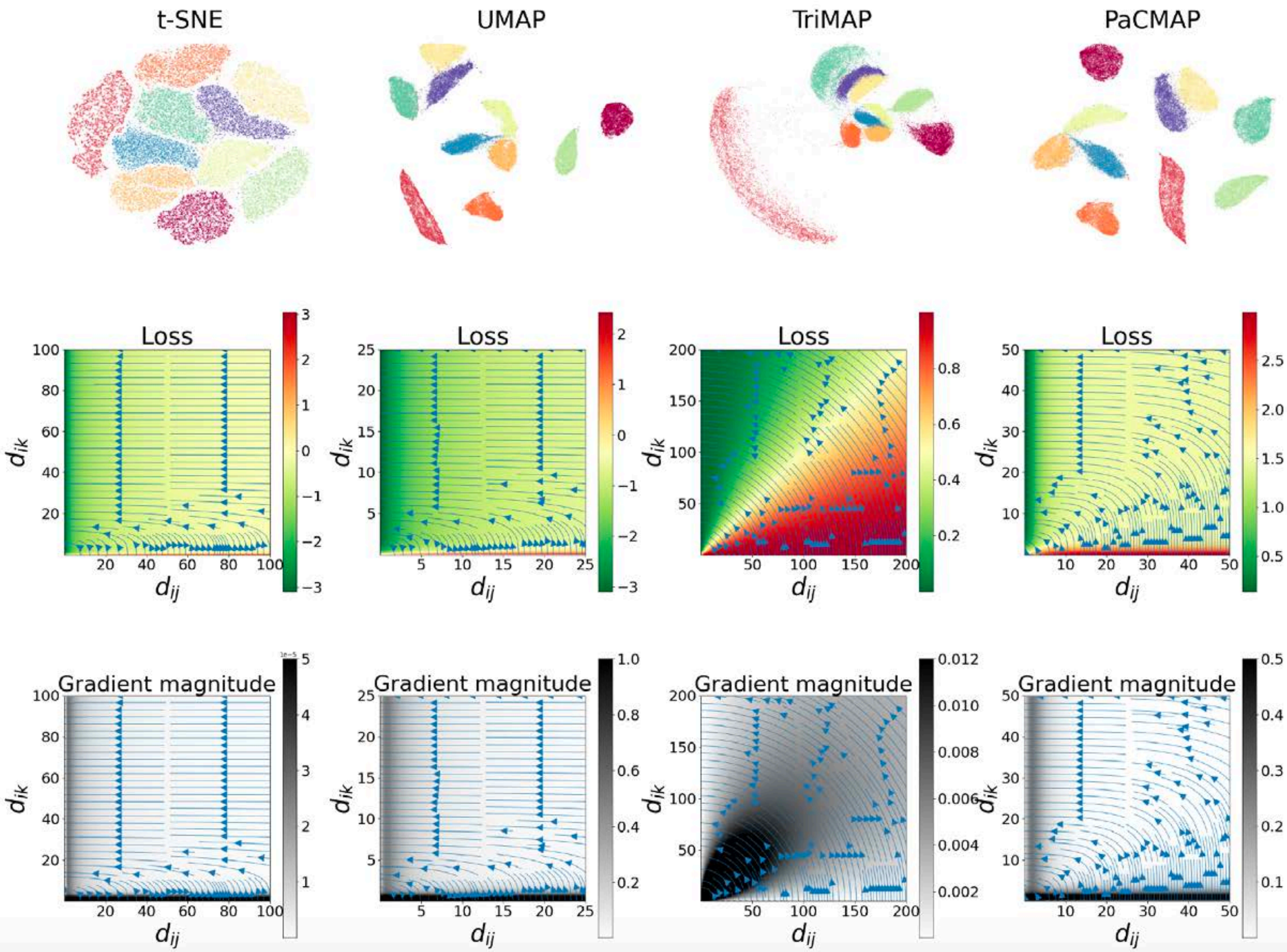
The “rainbow” plot

Triple i, j (neighbor), k (further)



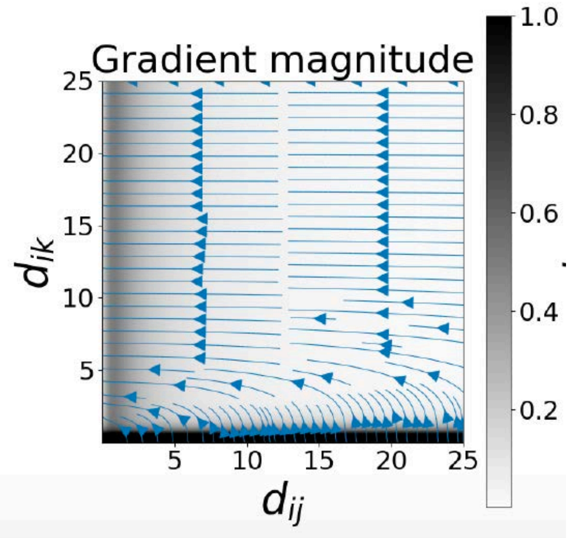


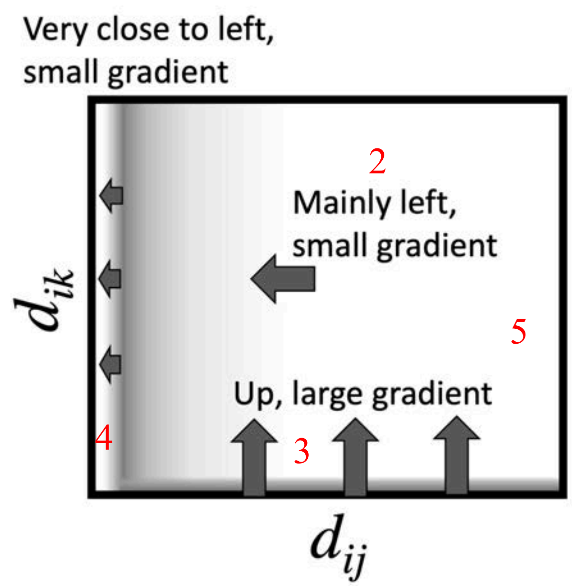


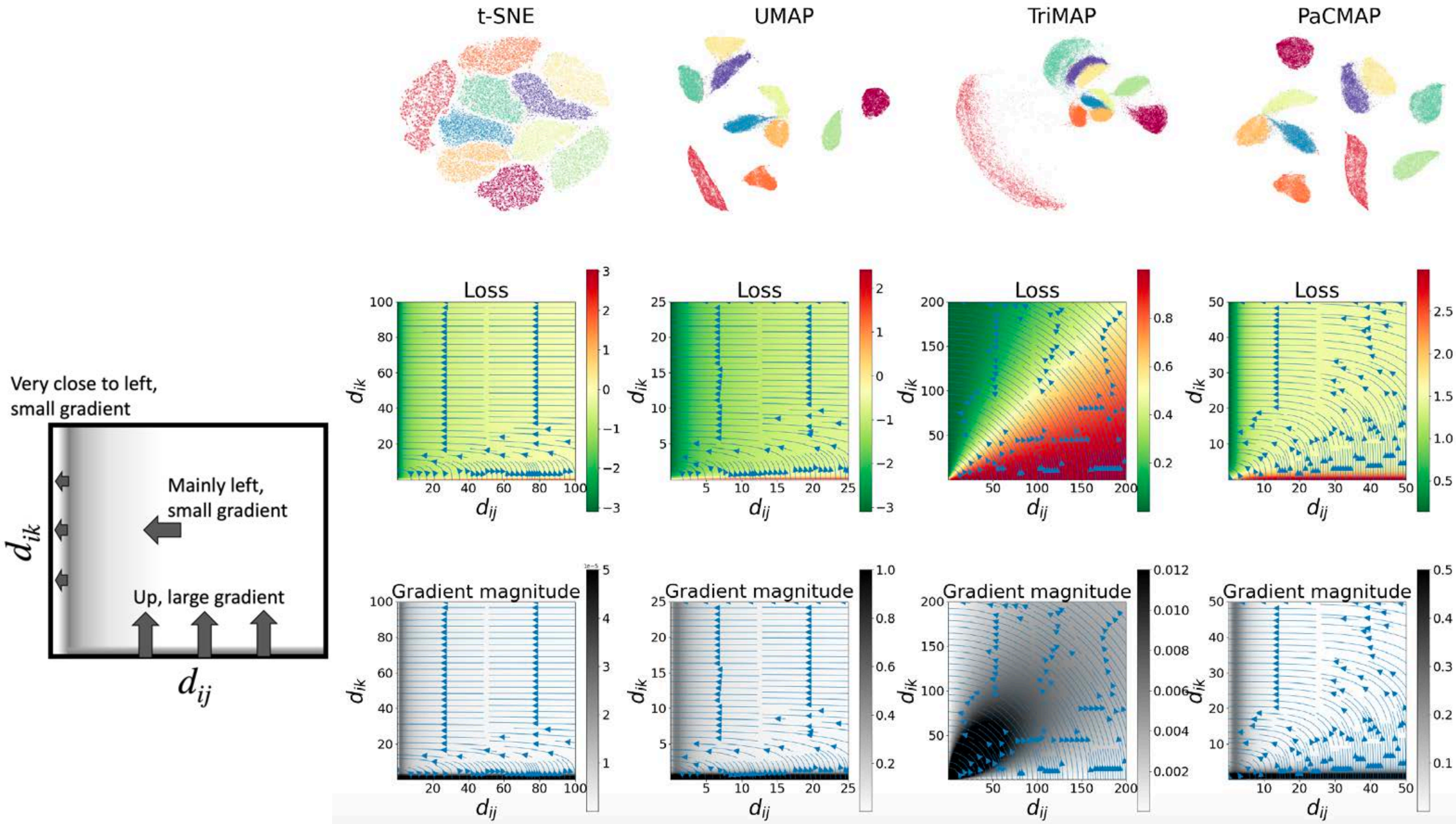

t-SNE
UMAP
TriMAP
PaCMAP
Loss
Gradient magnitude
d_{ij}
d_{ik}

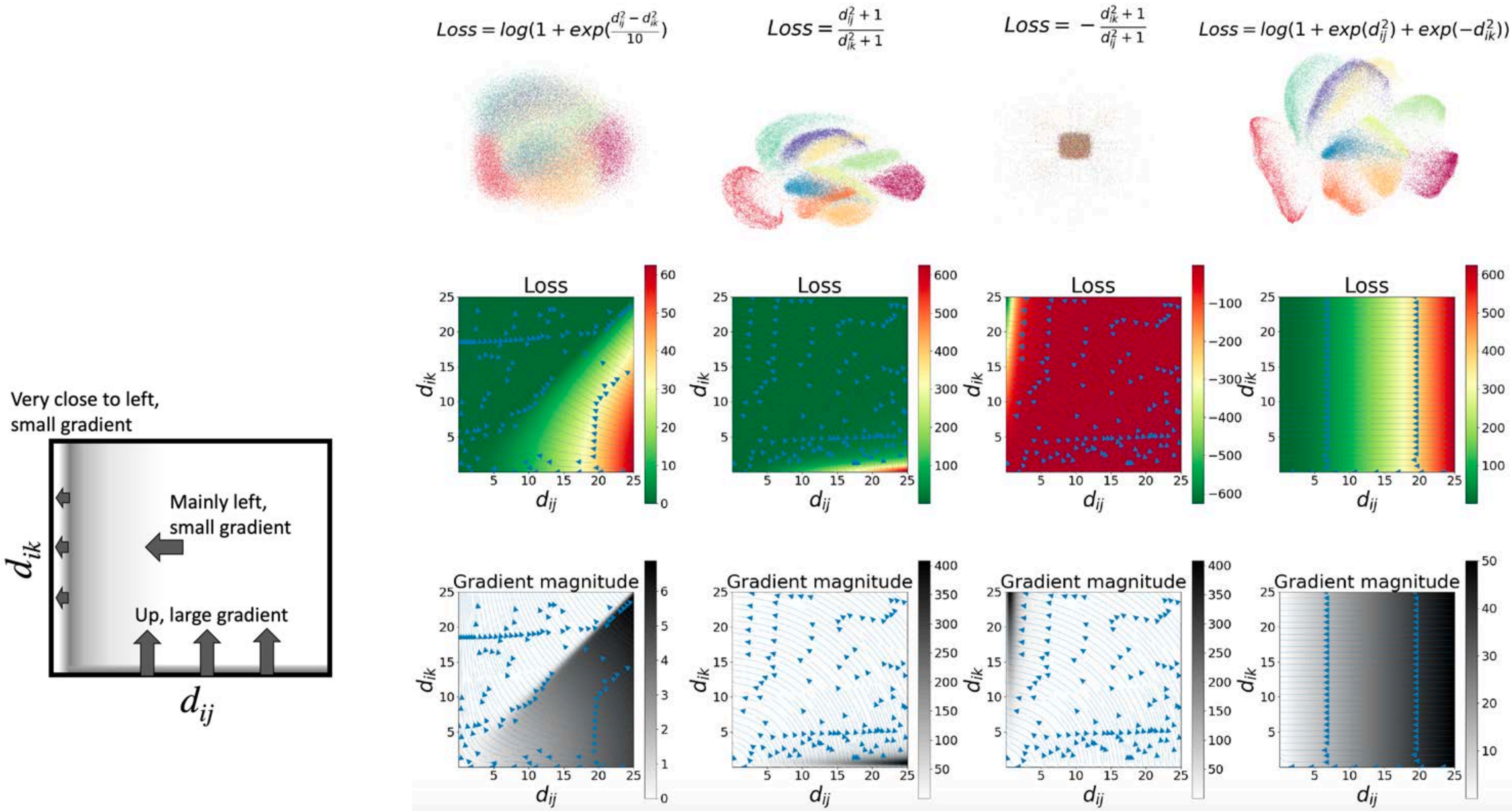
Principles for a good loss for DR

1) Monotonicity: pull neighbors closer, push farther points away (go left, go up)
2) Except along the bottom, gradient should go mainly to the left (broadly attract neighbors, further points are far enough), sufficient attraction
3) Along bottom, gradient goes mainly up (further point is too close) with large gradient
4) Along vertical axis, small magnitude (neighbor is close enough)
5) Weak pull on far neighbors: gradients should become small as distance to neighbor j becomes large






t-SNE
UMAP
TriMAP
PaCMAP
Loss
Gradient magnitude
Very close to left, small gradient
Mainly left, small gradient
Up, large gradient
d_{ik}
d_{ij}



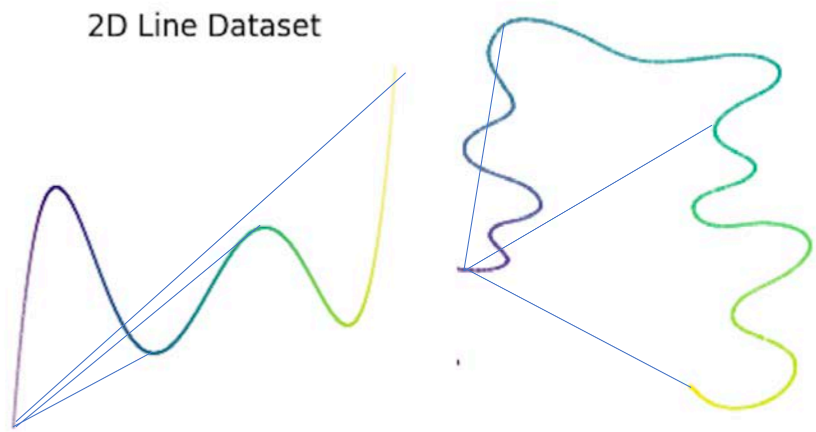
Too much repulsion | Insufficient attraction | No gradient on repulsion | Insufficient local attraction

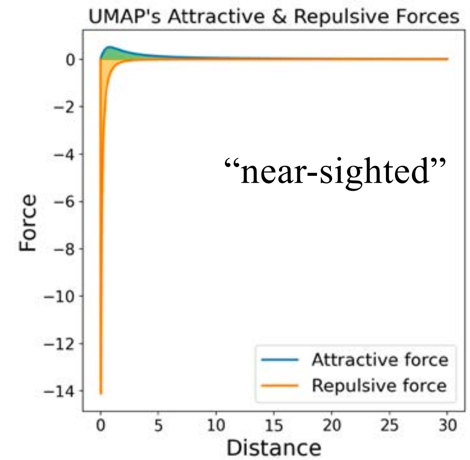
What we knew before:

- Certain properties of the loss function are important:
  - Attraction: neighbors should be attracted. But not too close! (Crowding)
  - Repulsion: farther points in original space should be far in low-dim space.

After a huge amount of experimentation, we found that:

- Certain specific properties of the loss function are important.
- We must have forces on non-neighbors.
  - The choice of which graph components to preserve are important.





$$\sum_{(i,j)\in\mathcal{T}_{\text{neighbors}}} l^{\text{attract}}(i,j) + \sum_{(i,k)\in\mathcal{T}_{\text{further}}} l^{\text{repulse}}(i,k)$$

After a huge amount of experimentation, we found that:

- Certain specific properties of the loss function are important.
- We must have forces on non-neighbors.
  - The choice of which graph components to preserve are important.

$$\text{Loss}^{\text{PaCMAP}} = w_{\text{neighbors}}\text{Loss}_{\text{neighbors}} + w_{MN}\text{Loss}_{MN} + w_{FP}\text{Loss}_{FP}$$

$$\text{Loss}_{\text{neighbors}} = \frac{\tilde{d}_{ij}}{10 + \tilde{d}_{ij}}, \quad \text{Loss}_{MN} = \frac{\tilde{d}_{ik}}{10000 + \tilde{d}_{ik}}, \quad \text{Loss}_{FP} = \frac{1}{1 + \tilde{d}_{il}}$$

Neighbors: attractive

Mid-near pairs: mild attractive

Further points: repulsive

After a huge amount of experimentation, we found that:

- Certain specific properties of the loss function are important.
- We must have forces on non-neighbors.
    - The choice of which graph components to preserve are important.

$$\mathrm{Loss}^{\mathrm{PaCMAP}} = w_{\mathrm{neighbors}}\mathrm{Loss}_{\mathrm{neighbors}} + w_{MN}\mathrm{Loss}_{MN} + w_{FP}\mathrm{Loss}_{FP}$$

$$\mathrm{Loss}_{\mathrm{neighbors}} = \frac{\tilde{d}_{ij}}{10 + \tilde{d}_{ij}}, \quad \mathrm{Loss}_{MN} = \frac{\tilde{d}_{ik}}{10000 + \tilde{d}_{ik}}, \quad \mathrm{Loss}_{FP} = \frac{1}{1 + \tilde{d}_{il}}$$

Neighbors: attractive

Mid-near pairs: mild attractive

Further points: repulsive

The weights change on a schedule:

Period 1: $w_{\mathrm{neighbors}}$ is medium, w_{MN} is huge, w_{FP} is medium

Period 2: $w_{\mathrm{neighbors}}$ is large, w_{MN} is small, w_{FP} is medium

Period 3: $w_{\mathrm{neighbors}}$ is medium, w_{MN} is 0, w_{FP} is medium

SVM accuracy (measures local structure preservation)

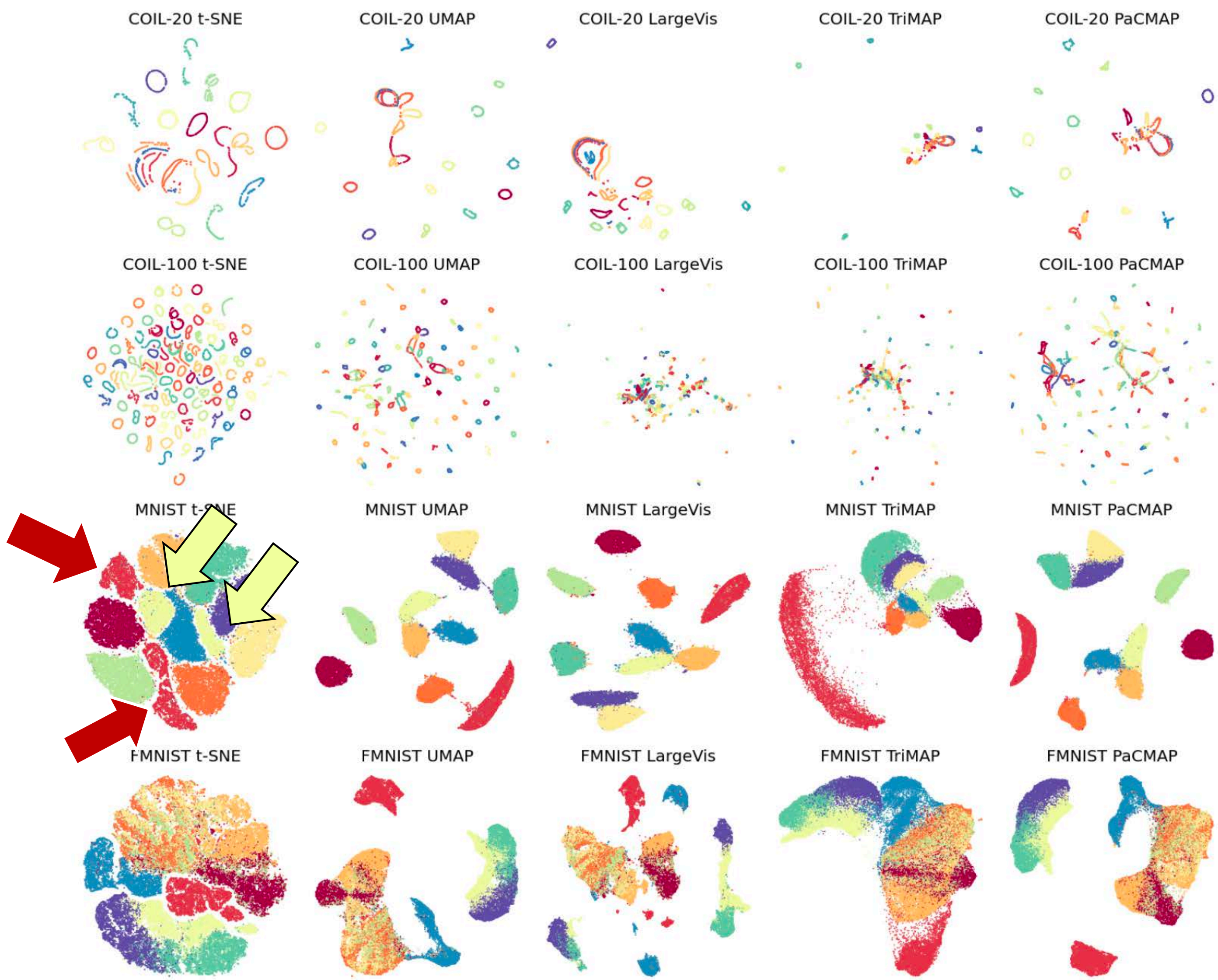
Dataset (Size)	Baseline	t-SNE	LargeVis	UMAP	TriMAP	PaCMAP
COIL-20 (1.4K)	0.972	0.909 ± 0.015	0.799 ± 0.020	0.844 ± 0.004	0.778 ± 0.010	**0.942** ± **0.009**
COIL-100 (7.2K)	0.989	0.911 ± 0.004	0.707 ± 0.014	0.879 ± 0.007	0.737 ± 0.019	**0.933** ± **0.009**
USPS (9K)	0.949	***0.959*** ± ***0.002***	*0.957* ± *0.001*	*0.956* ± *0.002*	0.946 ± 0.001	*0.958* ± *0.001*
Mammoth (10K)	0.961	0.927 ± 0.009	0.923 ± 0.011	***0.941*** ± ***0.003***	0.900 ± 0.004	*0.933* ± *0.004*
20Newsgroups (18K)	0.792	0.435 ± 0.014	*0.444* ± *0.012*	0.431 ± 0.013	0.410 ± 0.007	***0.447*** ± ***0.006***
MNIST (70K)	0.926	0.967 ± 0.002	0.965 ± 0.004	0.970 ± 0.001	0.960 ± 0.001	**0.974** ± **0.001**
F-MNIST (70K)	0.854	***0.754*** ± ***0.003***	*0.748* ± *0.003*	0.742 ± 0.003	0.729 ± 0.001	*0.752* ± *0.004*

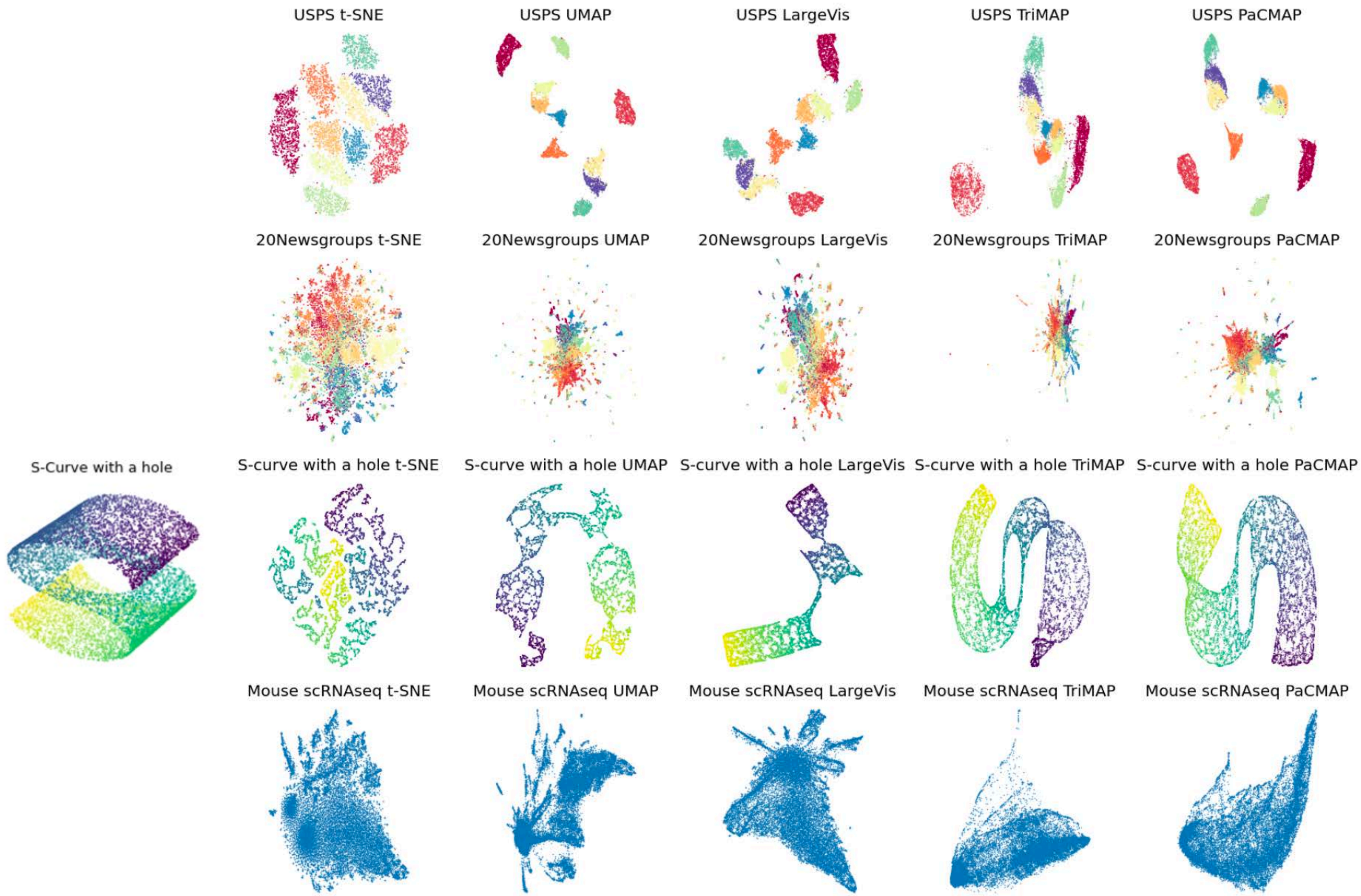
Random triplet accuracy (measures global structure preservation)

Dataset (Size)	t-SNE	LargeVis	UMAP	TriMAP	PaCMAP
COIL-20 (1.4K)	0.698 ± 0.016	**0.735** ± **0.011**	0.649 ± 0.014	0.659 ± 0.006	0.699 ± 0.007
COIL-100 (7.2K)	0.577 ± 0.012	0.630 ± 0.021	0.568 ± 0.011	0.633 ± 0.002	**0.718** ± **0.005**
USPS (9K)	0.654 ± 0.013	*0.668* ± *0.011*	***0.669*** ± ***0.002***	0.640 ± 0.002	*0.665* ± *0.002*
S-Curve with Hole (9.5K)	0.722 ± 0.045	0.834 ± 0.041	0.800 ± 0.013	0.838 ± 0.004	**0.866** ± **0.010**
Mammoth (10K)	0.701 ± 0.038	0.766 ± 0.024	0.816 ± 0.001	***0.874*** ± ***0.001***	*0.872* ± *0.003*
20Newsgroups (18K)	0.645 ± 0.002	0.632 ± 0.001	0.664 ± 0.002	**0.704** ± **0.002**	0.666 ± 0.003
Mouse scRNA-seq (20K)	0.715 ± 0.002	0.719 ± 0.003	*0.727* ± *0.002*	***0.728*** ± ***0.001***	*0.727* ± *0.001*
MNIST (70K)	0.600 ± 0.007	0.601 ± 0.007	0.614 ± 0.001	0.600 ± 0.001	**0.619** ± **0.001**
F-MNIST (70K)	0.679 ± 0.019	0.657 ± 0.011	0.740 ±0.001	**0.777** ± **0.001**	0.741 ± 0.002
Flow Cytometry (3M)	–	–	–	0.857 ± 0.001	**0.894** ± **0.005**
KDD Cup99 (4M)	–	–	–	0.660 ± 0.007	**0.752** ± **0.002**

(Ran out of memory or time, >24 hrs per run)

t-SNE version: opt-SNE (Belkina et al., 2019) built on Multi-core t-SNE (Ulyanov et al., 2016)


COIL-20 t-SNE
COIL-20 UMAP
COIL-20 LargeVis
COIL-20 TriMAP
COIL-20 PaCMAP
COIL-100 t-SNE
COIL-100 UMAP
COIL-100 LargeVis
COIL-100 TriMAP
COIL-100 PaCMAP
MNIST t-SNE
MNIST UMAP
MNIST LargeVis
MNIST TriMAP
MNIST PaCMAP
FMNIST t-SNE
FMNIST UMAP
FMNIST LargeVis
FMNIST TriMAP
FMNIST PaCMAP


USPS t-SNE
USPS UMAP
USPS LargeVis
USPS TriMAP
USPS PaCMAP
20Newsgroups t-SNE
20Newsgroups UMAP
20Newsgroups LargeVis
20Newsgroups TriMAP
20Newsgroups PaCMAP
S-Curve with a hole
S-curve with a hole t-SNE
S-curve with a hole UMAP
S-curve with a hole LargeVis
S-curve with a hole TriMAP
S-curve with a hole PaCMAP
Mouse scRNAseq t-SNE
Mouse scRNAseq UMAP
Mouse scRNAseq LargeVis
Mouse scRNAseq TriMAP
Mouse scRNAseq PaCMAP

Understanding Data and Models Through Visualization

- Point 1: Interpretable models don’t need to be simple or small… if you can co-design interpretable models with their visualizations

- Point 2: When using data visualization tools, it’s useful to have an understanding of their strengths and weaknesses

Thanks