

# Overcoming Machine Learning's Data Bottlenecks

Fred Sala

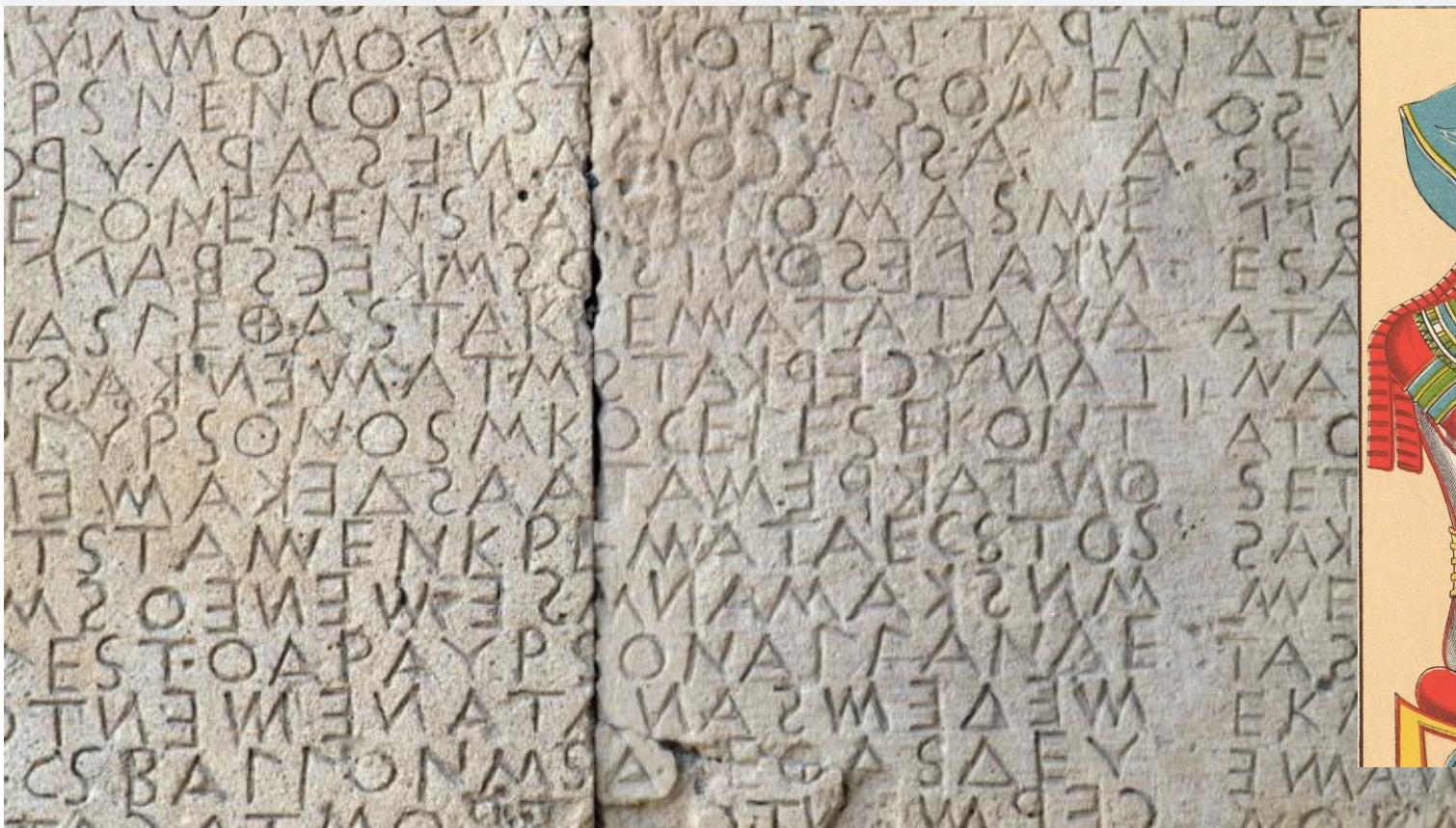


# ML Progress

WRITTEN IN STONE —

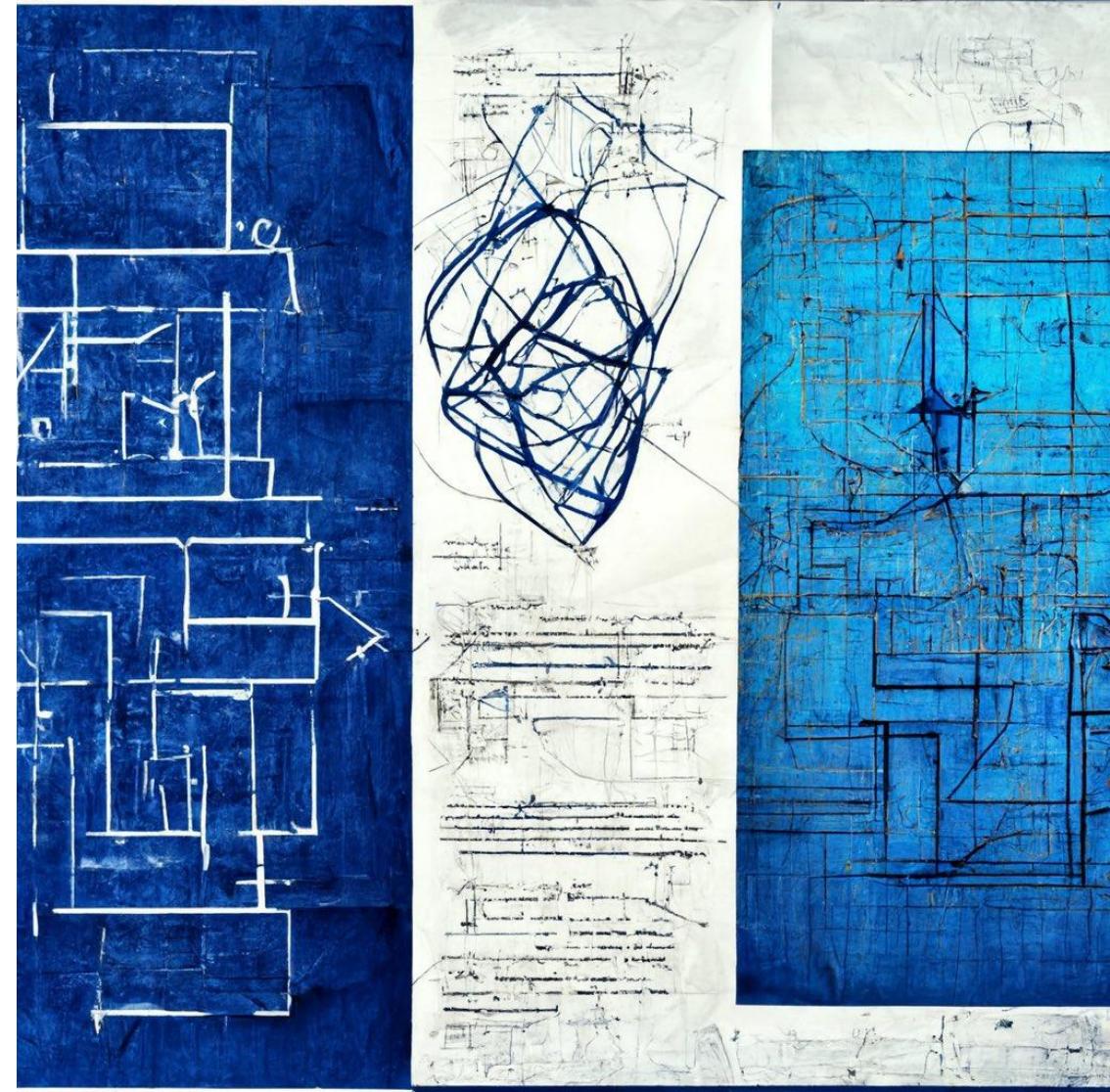
DeepMind's new AI gives historians a powerful new tool to interpret the past

Google launches hieroglyphics translator powered by AI



Google Translate / Life Collection

# ML Progress



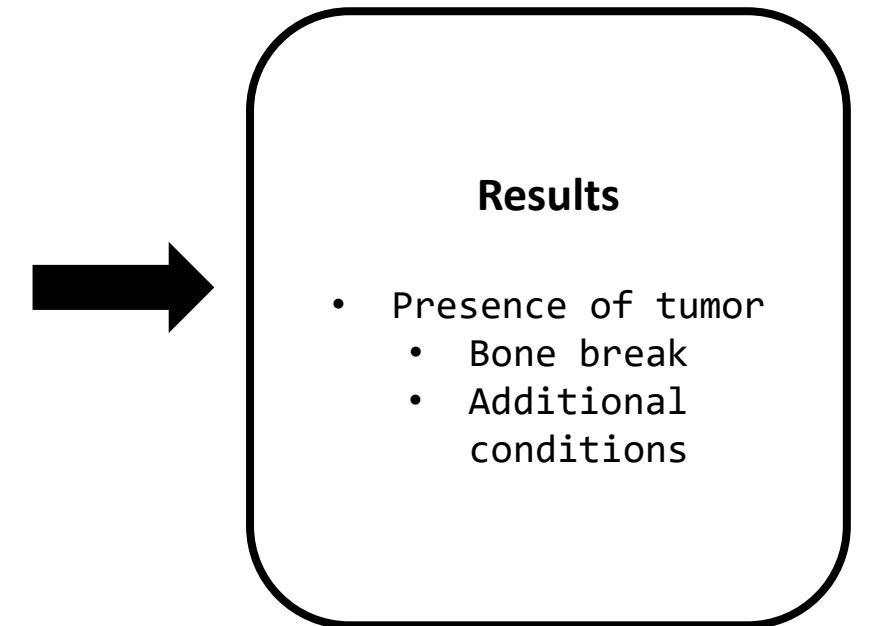
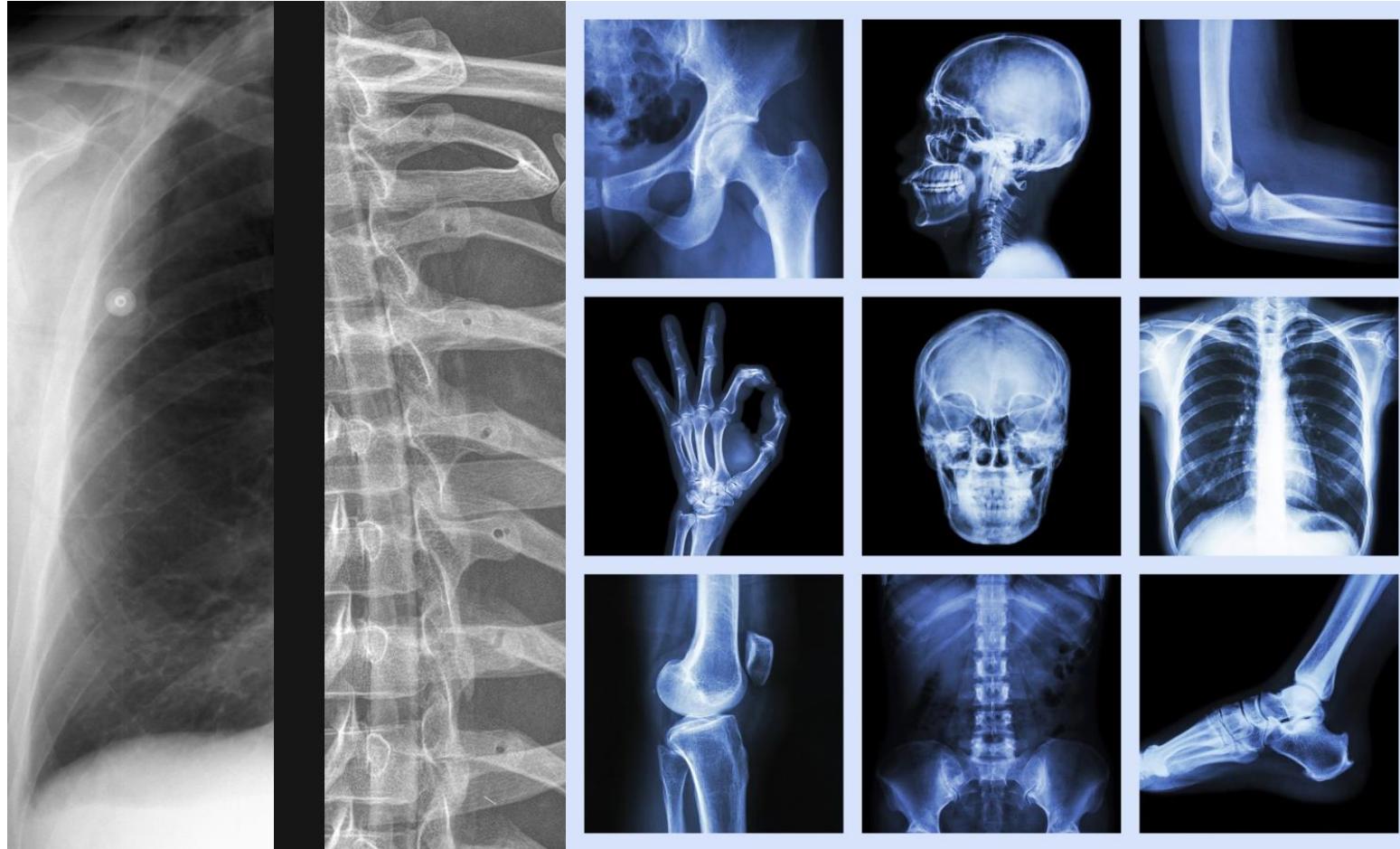
# ML Progress

The image shows a screenshot of the GitHub Copilot interface. On the left, there's a large white text area with the GitHub logo and the text "GitHub Copilot". Below it is a button labeled "Technical preview". The main area features a large white text "Your AI pair programmer" overlaid on a dark background with a grid pattern. On the right, there's a code editor window with two tabs: "JS fetch\_pic.js" and "push\_to\_git.py". The "fetch\_pic.js" tab contains the following code:

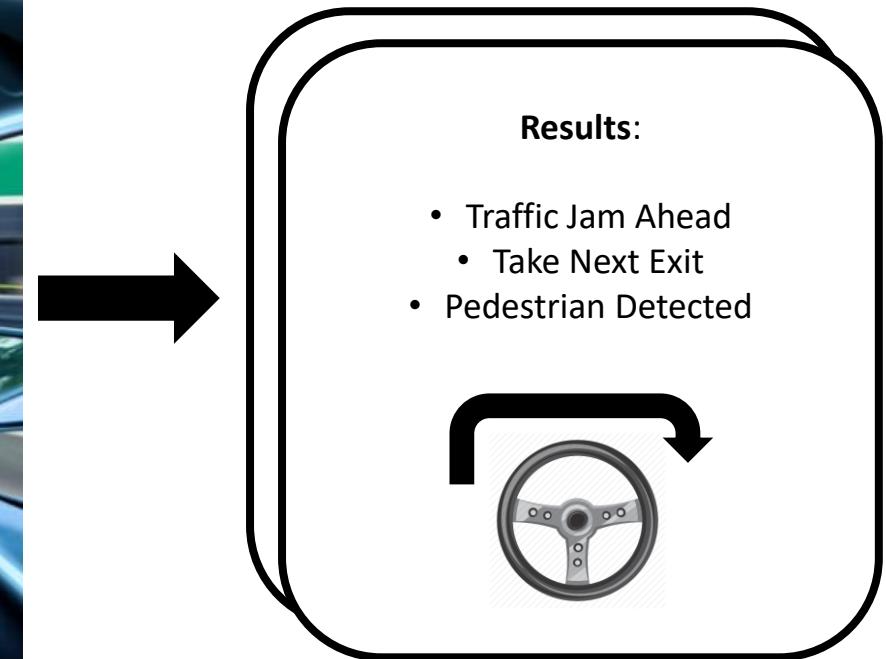
```
1 const fetchNASAPictureOfDay =  
2   return fetch('https://api.nasa.gov/planetary/pictures_of_the_day?  
3     method: 'GET',  
4     headers: {  
5       'Content-Type': 'application/json'  
6     },  
7   })  
8   .then(response => response.json())  
9   .then(json => {  
10     return json;  
11   });  
12 }
```

A blue vertical bar highlights the code between line 8 and 11. At the bottom of the code editor, there's a blue button with the GitHub Copilot logo and the text "Copilot". Above the code editor, there's a progress bar consisting of several small squares.

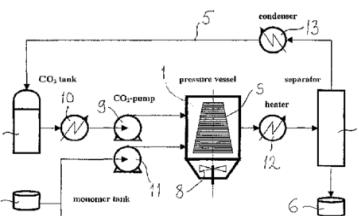
# ...and ML Promise

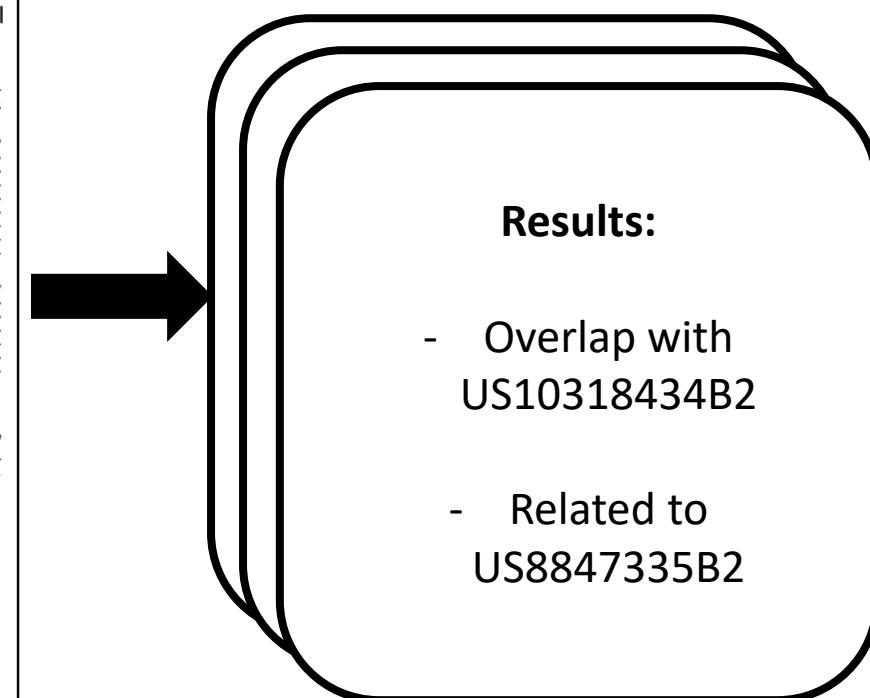


# ML Promise



# ML Promise

<p><b>SUBCONTRACT AGREEMENT</b> (Long Form, With Bond, Rev. April 06, 2001)</p> <p><b>Project:</b> Project Name <b>Subcontract:</b> Subcontract</p> <p><b>General Contractor:</b> TG CONSTRUCTION, INC. 119 Standard Street El Segundo, California 90245</p> <p><b>Subcontractor:</b> Sub Name Address City, State Zip License # Sub License No. Attention: Contact Name</p> <p><b>Site Address:</b> Project Name Address City, State Zip Telephone: Project Phone Fax: Project Fax</p> <p>In consideration of the mutual covenants hereinafter set forth, Contractor and Subcontractor agree as follows:</p> <p><b>I. Definitions</b></p> <ul style="list-style-type: none"> <li>A. Owner: Owner's Name</li> <li>B. Architect: Architect's Name</li> <li>C. General Contract: The agreement between Owner and General Contractor.</li> <li>D. Contract Documents: The drawings and specifications, general conditions and the Architet's and/or the Architect's consultants covering the work.</li> <li>E. Work: All work to be performed by the General Contractor pursuant to the General Contract.</li> <li>F. Site: The real property on which the work is to be constructed as more particularly described in the General Contract.</li> <li>G. Subcontract: This agreement between General Contractor and Subcontractor and TG Construction's General Terms pages 1-6 dated April 06, 2001 attached hereto to the terms hereof.</li> <li>H. Subcontract Documents: This Subcontract and all drawings and specific documents required to be furnished by the Subcontractor.</li> <li>I. Subcontract work: All work required to be performed pursuant to the terms described as:</li> </ul> <p>Project Name in accordance with plans and specifications issued by the Architect and/or the Architect's consultants covering the work.</p> <p>The WORK OF THIS SUBCONTRACT INCLUDES BUT IS NOT LIMITED TO</p> <p><b>A. SCOPE OF WORK:</b> (Contracts Scope Of Work)</p> <p><b>Inclusions:</b> (Contract Inclusions Description)</p> <p>- End of Section -</p> <p><b>B. Exclusions:</b> (Contract Exclusions Description)</p> <p>- End of Section -</p> <p><b>C. SCHEDULE:</b> The Subcontractor shall provide sufficient manpower and equipment upon the General Contractor's Construction Schedule, Exhibit "A". Confirmation of this schedule will be regularly discussed as the work progresses. Other "look ahead" schedules as may be distributed by the General Contractor.</p>		<p align="center"><b>Residential Lease Agreement</b></p> <p>THIS LEASE AGREEMENT (hereinafter referred to as the "Agreement") made this _____ day of _____, 20_____, by and between _____ whose address is _____ and _____ (hereinafter referred to as the "Parties").</p> <p><b>WITNESSETH:</b></p> <p>WHEREAS, Lessor is the fee owner of certain real property being _____, such real property having a street address of _____.</p> <p>WHEREAS, Lessor is desirous of leasing the Premises to Lessee upon the terms and conditions set forth herein.</p> <p>WHEREAS, Lessee is desirous of leasing the Premises from Lessor on the terms and conditions set forth herein.</p> <p>NOW, THEREFORE, for and in consideration of the sum of TEN DOLLARS (\$10.00) contained herein and other good and valuable consideration, the receipt and sufficiency whereof hereby agree as follows:</p> <ol style="list-style-type: none"> <li>1. <b>TERM.</b> Lessor leases to Lessee and Lessee leases from Lessor the appurtenances thereto, for a term of _____ years(s), such term beginning midnight on _____.</li> <li>2. <b>RENT.</b> The total rent for the term hereof is the sum of (\$_____) payable on the _____ day of each month. _____ DOLLARS (\$) for this Agreement, the second installment to be paid on _____.</li> <li>3. <b>USE OF PREMISES.</b> The Premises shall be used and occupied by Lessee as a dwelling, and no part of the Premises shall be used at any time during the term of this Agreement for carrying on any business, profession, or trade of any kind, or for dwelling. Lessee shall not allow any other person, other than Lessee's who are guests of Lessee, to use or occupy the Premises without first obtaining the written consent of Lessor. Lessee shall comply with any and all laws, ordinances, rules and orders of authorities affecting the cleanliness, use, occupancy and preservation of the Premises.</li> <li>4. <b>CONDITION OF PREMISES.</b> Lessee stipulates, represents and warrants that the Premises are in good order, repair, and in a safe, clean condition at the time of this Lease.</li> <li>5. <b>ASSIGNMENT AND SUB-LETTING.</b> Lessee shall not assign this Agreement or any part thereof without the prior written consent of Lessor. Any letter or license shall not be deemed to be a consent to any subsequent sub-letting or license without the prior written consent of Lessor or an assignment or license provided by written agreement between Lessor and Lessee.</li> <li>6. <b>ALTERATIONS AND IMPROVEMENTS.</b> Lessee shall make no alterations or improvements to the Premises or construct any building or make any other improvements on the Premises. Any and all alterations, changes, and/or improvements built, constructed or installed by Lessee shall remain the property of Lessor unless otherwise provided for by written agreement between Lessor and Lessee.</li> <li>7. <b>NON-DELIVERY OF POSSESSION.</b> In the event Lessor cannot commence the term of this Lease, through no fault of Lessor or its agent, but the rental herein provided shall abate until possession is given. Lessor shall remain liable for the rental amount for the period of time remaining on the lease.</li> </ol>
<p align="right"><b>WO 2005/071696 A1</b></p>		<p align="center">(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)</p> <p align="center">(19) World Intellectual Property Organization International Bureau</p> <p align="center">PCT</p> <p align="center">(43) International Publication Date 4 August 2005 (04.08.2005)</p> <p align="center">(10) International Publication Number WO 2005/071696 A1</p> <p align="center">   </p> <p>(51) International Patent Classification? H01B 1/12.</p> <p>(54) Designated States (index otherwise indicated, for every kind of national protection available): AF, AG, AI, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CI, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, IU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NL, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SH, SG, SK, SI, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.</p> <p>(71) Applicant (for all designated States except US): VALMET TEKNILLINEN TUTKIMUSKESKUS [FI/FI]; Vuorimiehenkatu 5, FI-02150 Espoo (FI).</p> <p>(72) Inventors; and</p> <p>(75) Inventors/Applicants (for US only): PELTO, Jani [FI/FI]; Kullantie 11 A 2, FI-33960 Pirkkala (FI); AALTO, Samu [FI/FI]; Kemiintie 1 C 17, FI-33720 Tampere (FI); LAITINEN, Antero [FI/FI]; Kunnaskyläntie 3 B 45, FI-02330 Espoo (FI).</p> <p>(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GIL, GM, KE, LS, MW, MZ, NA, SD, SL, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CL, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LU, MC, NL, PL, PT, RO, SI, SI, SK, TR), OAPI (BJ, CI, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).</p> <p align="center">[Continued on next page]</p> <p align="center">(54) Title: PROCESS FOR DEPOSITION OF CONDUCTIVE POLYMER COATINGS IN SUPERCRITICAL CARBON DIOXIDE</p> <p align="center">  </p> <p align="center">(57) Abstract: A method for forming an electrically conductive polymeric surface on a solid polymeric substrate (S) comprises the following successive steps: 1) treatment of the solid polymeric substrate (S) in a pressure reactor (1) with a first supercritical or liquid carbon dioxide phase containing a monomer to cause the monomer to enter the structure of the polymeric substrate, 2) removal of the first supercritical or liquid carbon dioxide phase from the reactor (1), together with possible residues of the monomer, 3) feeding of a second supercritical or liquid carbon dioxide phase containing an oxidative agent into the reactor (1) into contact with the substrate (S) that has remained in the reactor, and 4) performing an in-situ oxidative polymerization of the monomer in the polymeric substrate with the help of the oxidative agent to form an electrically conductive polymeric surface on the polymeric substrate (S), 5) removal of the second supercritical or liquid carbon dioxide phase from the reactor.</p>



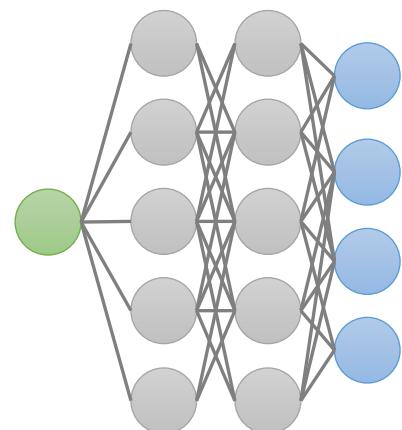
# Supervised Machine Learning

- Today's supervised ML pipeline components:

0.3500	0.8687	0.1690	0.5797	0.9037	0
0.1966	0.0844	0.6491	0.4389	0.8909	0
0.2511	0.3998	0.7317	0.1111	0.3342	0
0.6160	0.2599	0.6477	0.2581	0.6987	0
0.4733	0.8001	0.4509	0.4087	0.1978	0
0.3517	0.4314	0.5470	0.5949	0.0305	0
0.8308	0.9106	0.2963	0.2622	0.7441	0
0.5853	0.1818	0.7447	0.6028	0.5000	0
0.5497	0.2638	0.1890	0.7112	0.4799	0
0.9172	0.1455	0.6868	0.2217	0.9047	0
0.2858	0.1361	0.1835	0.1174	0.6099	0
0.7572	0.8693	0.3685	0.2967	0.6177	0
0.7537	0.5797	0.6256	0.3188	0.8594	0
0.3804	0.5499	0.7802	0.4242	0.8055	0
0.5678	0.1450	0.0811	0.5079	0.5767	0
0.0759	0.0850	0.9294	0.0855	0.1829	0
0.0540	0.6221	0.7757	0.2625	0.2399	0
0.5308	0.3510	0.4868	0.8010	0.8965	0
0.7792	0.5132	0.4359	0.0292	0.0287	0
0.9340	0.4018	0.4468	0.5289	0.4899	0
0.1299	0.0760	0.3063	0.7303	0.1679	0
0.5688	0.2399	0.5085	0.4886	0.9787	0
0.4694	0.1233	0.5108	0.5785	0.7127	0
0.0119	0.1839	0.8176	0.2373	0.5005	0
0.3371	0.2400	0.7948	0.4588	0.4711	0
0.1622	0.4173	0.6443	0.9631	0.0596	0
0.7943	0.0497	0.3786	0.5468	0.6820	0
0.3112	0.9027	0.8116	0.5211	0.0424	0
0.5285	0.9448	0.5328	0.2316	0.0714	0
0.1656	0.4909	0.3507	0.4889	0.5216	0
0.6020	0.4893	0.9390	0.6241	0.0967	0
0.2630	0.3377	0.8759	0.6791	0.8181	0
0.6541	0.9001	0.5502	0.3955	0.8175	0
0.6892	0.3692	0.6225	0.3674	0.7224	0
0.7482	0.1112	0.5870	0.9880	0.1499	0
0.4505	0.7803	0.2077	0.0377	0.6596	0
0.0838	0.3897	0.3012	0.8852	0.5186	0
0.2290	0.2417	0.4709	0.9133	0.9730	0
0.9133	0.4039	0.2305	0.7962	0.6490	0
0.1524	0.0965	0.8443	0.0987	0.9003	0
0.8258	0.1320	0.1948	0.2619	0.4538	0

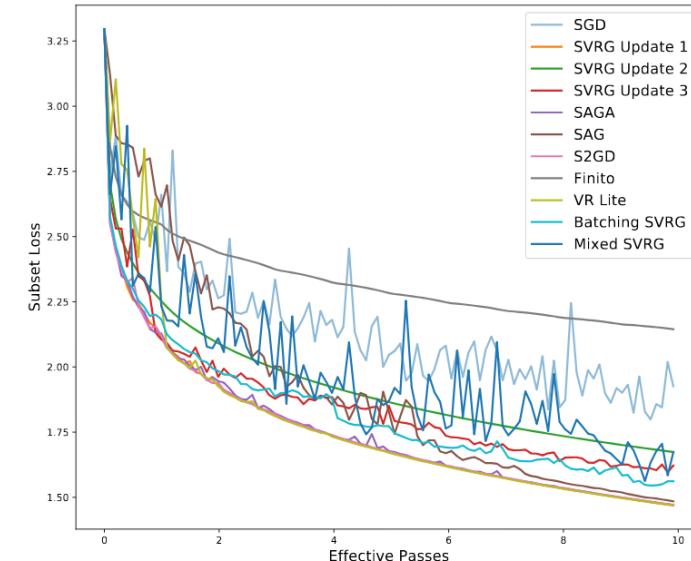
Data      Labels

$$-\frac{1}{N} \sum_{n=1}^N \left[ y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right]$$



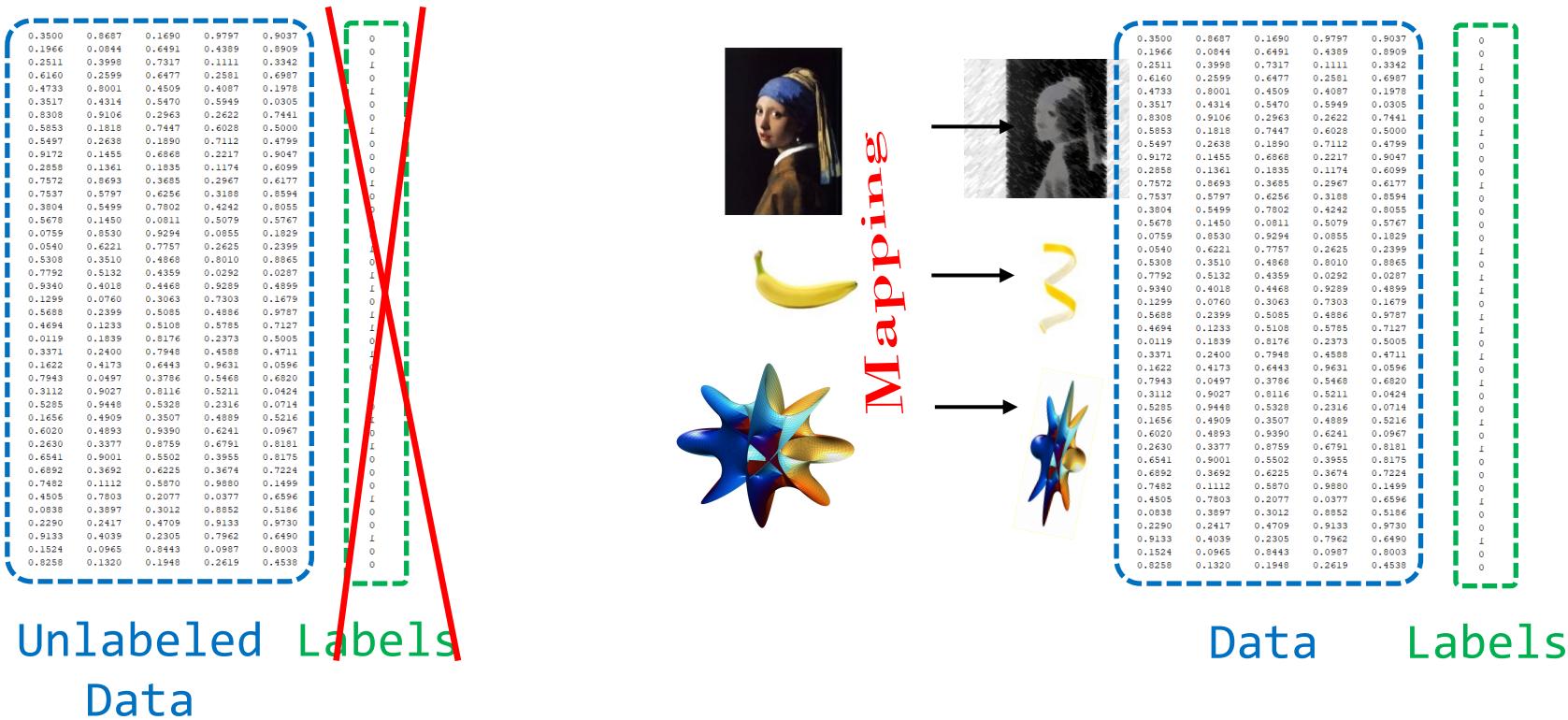
Model

Loss



Training

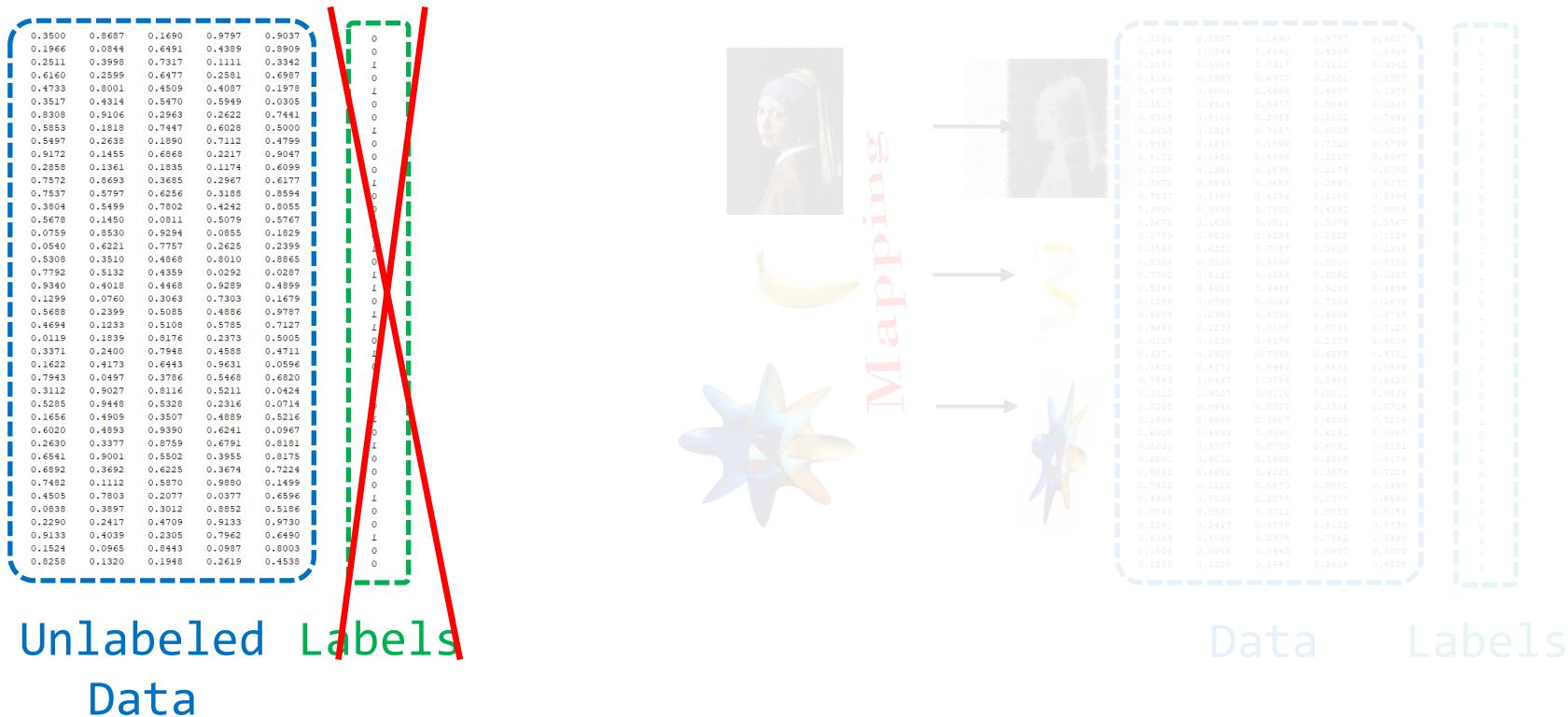
# Data Bottlenecks



**Bottleneck 1: Getting Labels**

**Bottleneck 2: Distortion**

# Data Bottlenecks



**Bottleneck 1: Getting Labels**

**Bottleneck 2: Distortion**

# The Need for Labels...

Modern supervised models need **lots** of labeled data



# The Need for Labels...

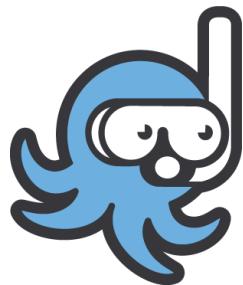
Modern supervised ML models need **lots** of labeled data  
Tons of unlabeled data, but labeling is

- Expensive,
- Static,
- Slow.



# Weak Supervision To the Rescue

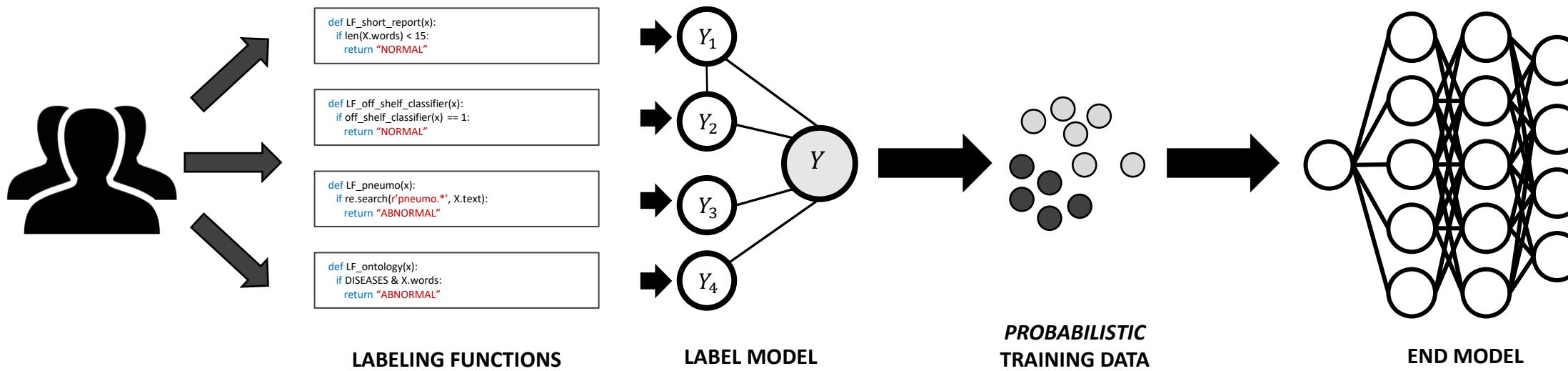
- **Weak supervision:** reduce the label bottleneck
  - Some of the users:



# Snorkel



# The Snorkel / Weak Supervision Pipeline

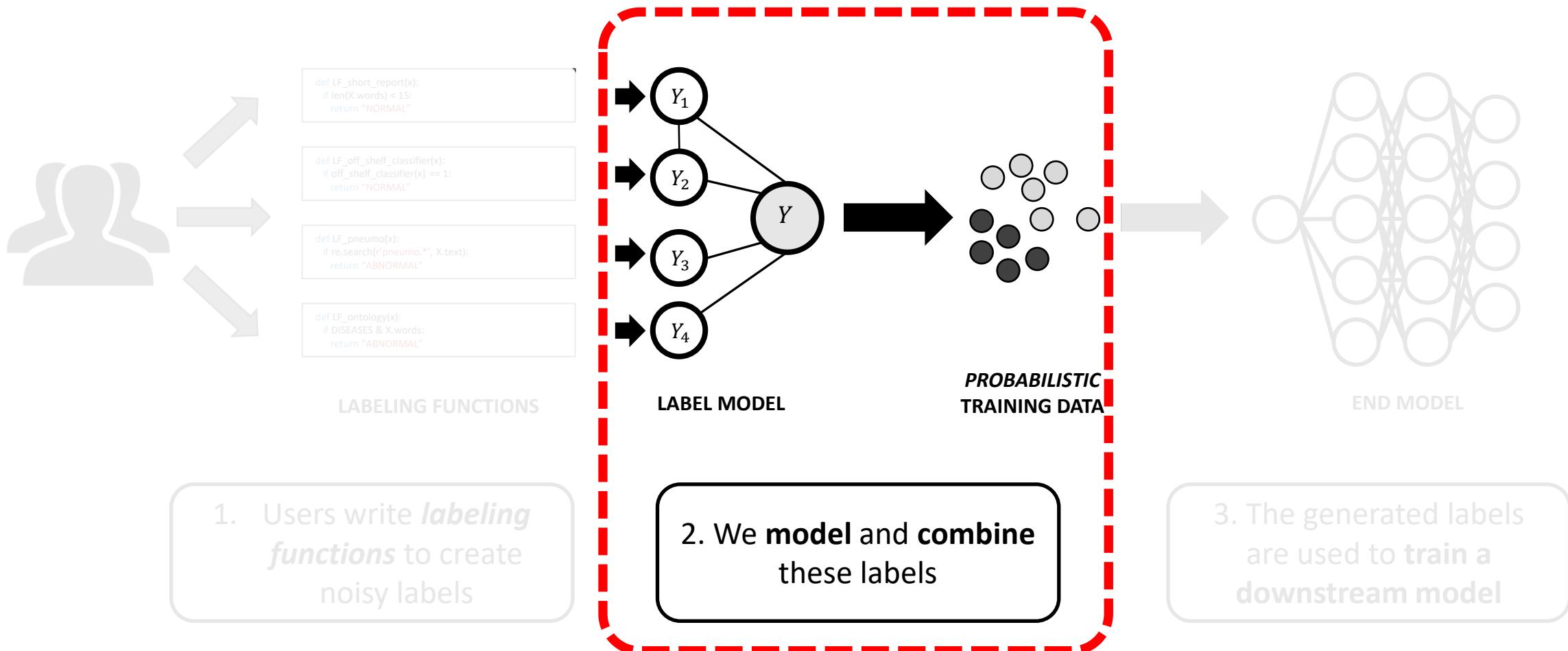


1. Users write ***labeling functions*** to create noisy labels

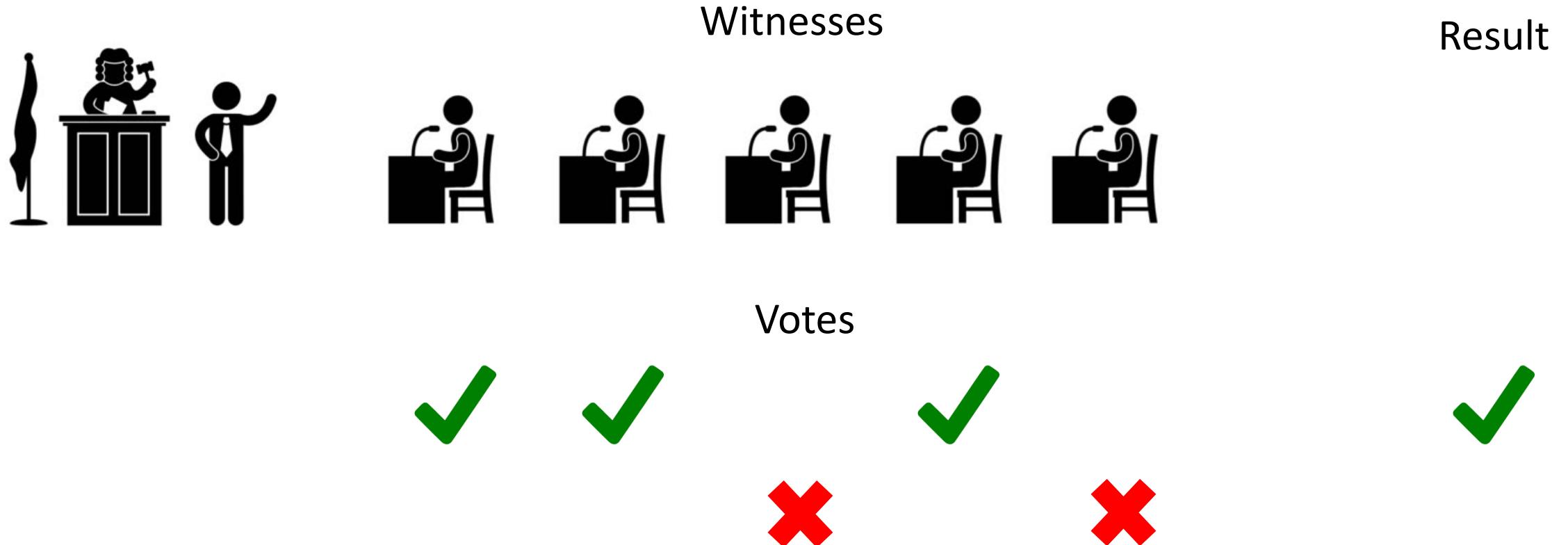
2. We **model** and **combine** these labels

3. The generated labels are used to **train a downstream model**

# The Snorkel / Weak Supervision Pipeline

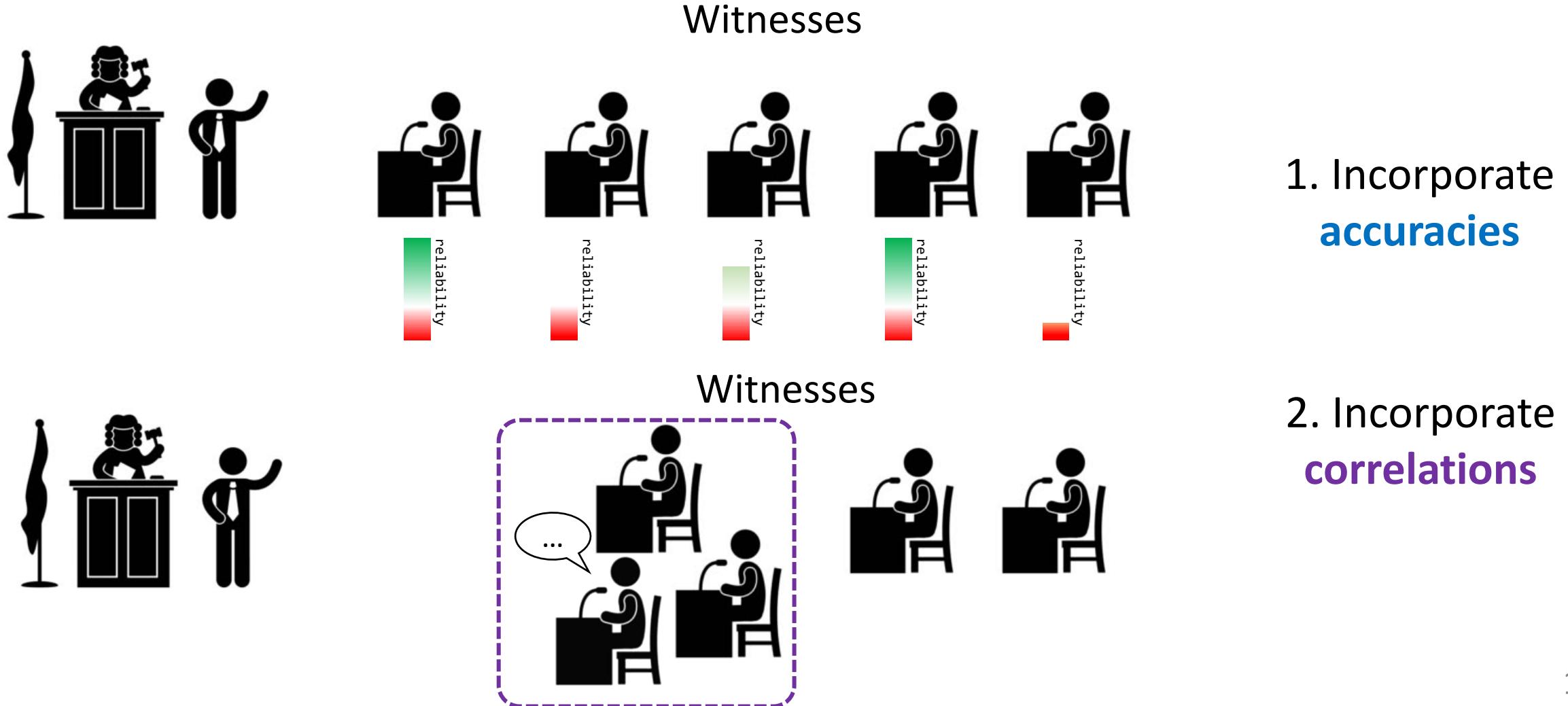


# Intuition

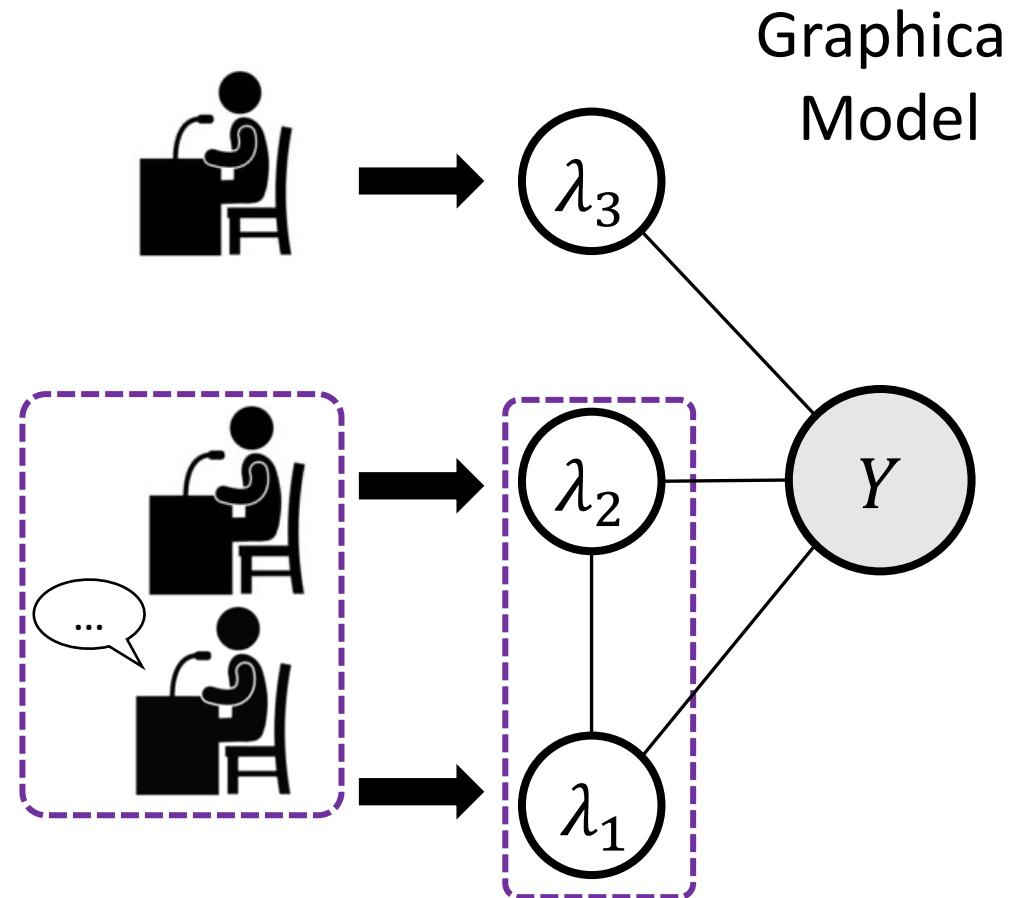


Naïve approach: **majority vote**

# Improving on Majority Vote



# Label Model



Parameters

1. **Accuracies**
2. **Correlations**

$$\mathbf{E}[\lambda_i Y]$$

$$\mathbf{E}[\lambda_i \lambda_j]$$

If we knew parameters... could do inference  $P(Y|\lambda_1, \lambda_2, \dots, \lambda_m)$

**Our goal:** learn parameters,  
**without observing  $Y$**

# How Does WS Work?

Look for latent relationships with **observables**

$$\mathbb{E}[\lambda_1 \lambda_2] = \mathbb{E}[\lambda_1 Y] \mathbb{E}[\lambda_2 Y]$$


**Observable:** Rate of  
agreement/disagreement

**Accuracy Parameters:** Want  
to estimate these

# How Does WS Work?

Exploit latent relationships with **observables**

$$\left\{ \begin{array}{l} \mathbb{E}[\lambda_1 \lambda_2] = \mathbb{E}[\lambda_1 Y] \mathbb{E}[\lambda_2 Y] \\ \mathbb{E}[\lambda_1 \lambda_3] = \mathbb{E}[\lambda_1 Y] \mathbb{E}[\lambda_3 Y] \\ \mathbb{E}[\lambda_2 \lambda_3] = \mathbb{E}[\lambda_2 Y] \mathbb{E}[\lambda_3 Y] \end{array} \right.$$



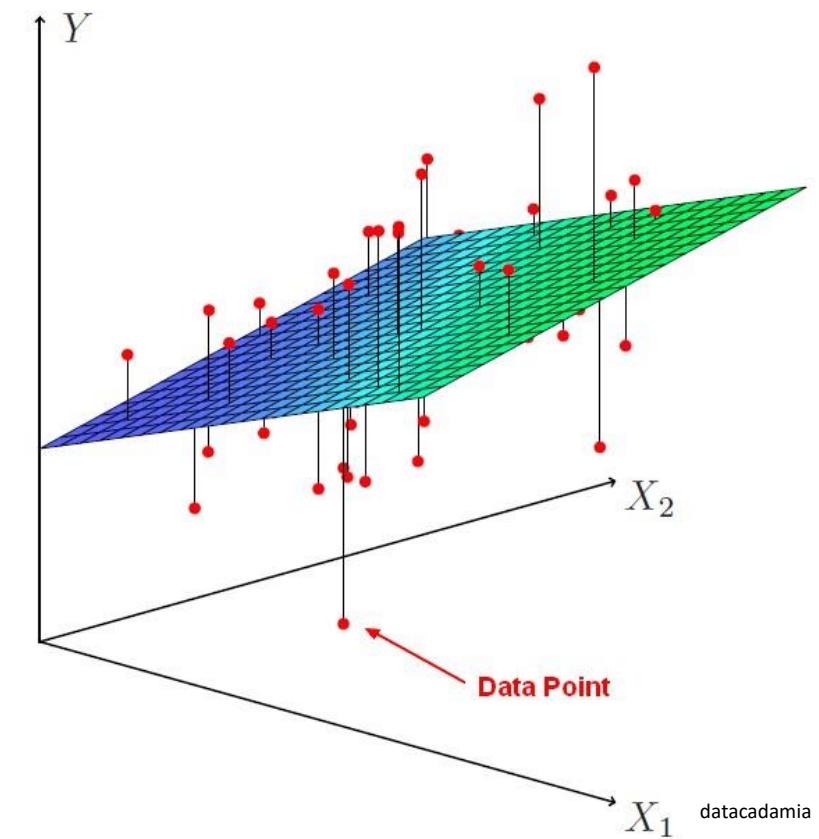
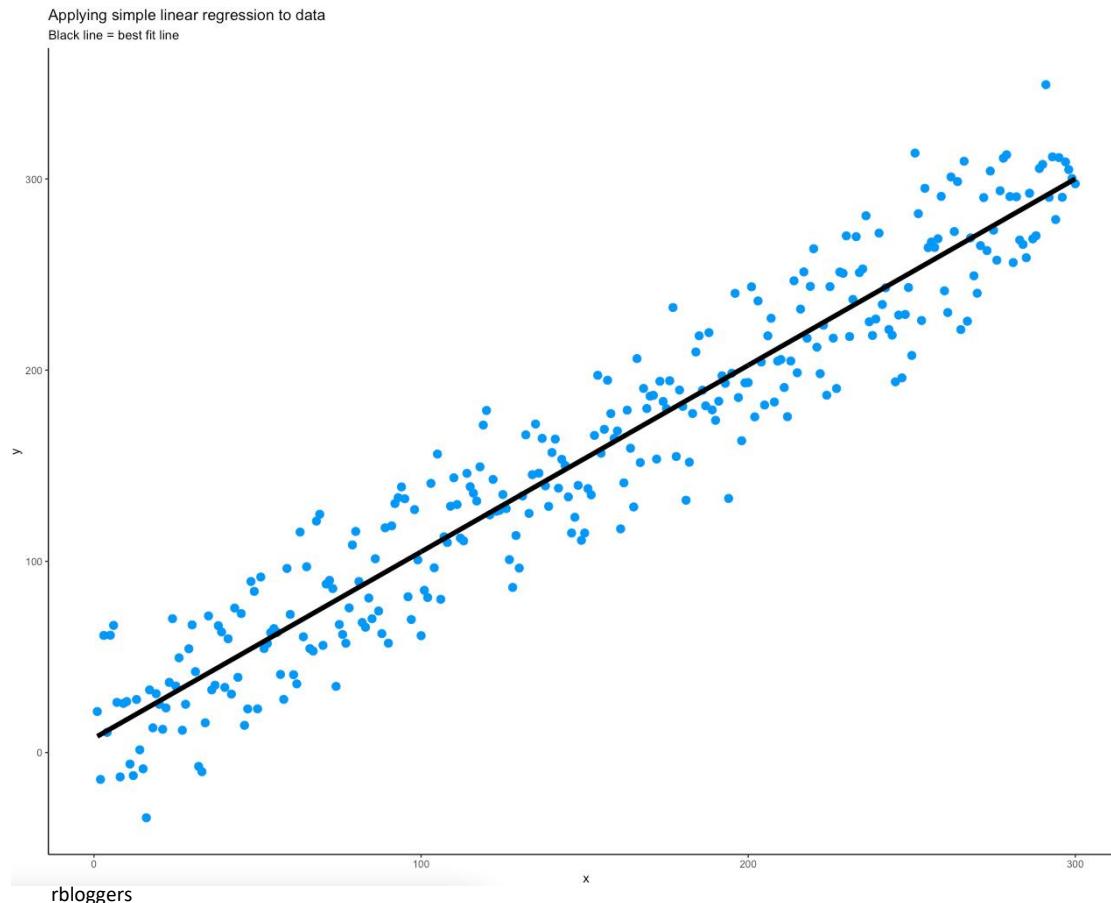
System in three accuracies, lhs are three pairwise rates

Multiply first two equations, divide by third

$$|\mathbb{E}[\lambda_1 Y]| = \sqrt{\frac{\mathbb{E}[\lambda_1 \lambda_2] \mathbb{E}[\lambda_1 \lambda_3]}{\mathbb{E}[\lambda_2 \lambda_3]}}$$

# But ML is Far More Diverse...

- Labels can be **real-valued**

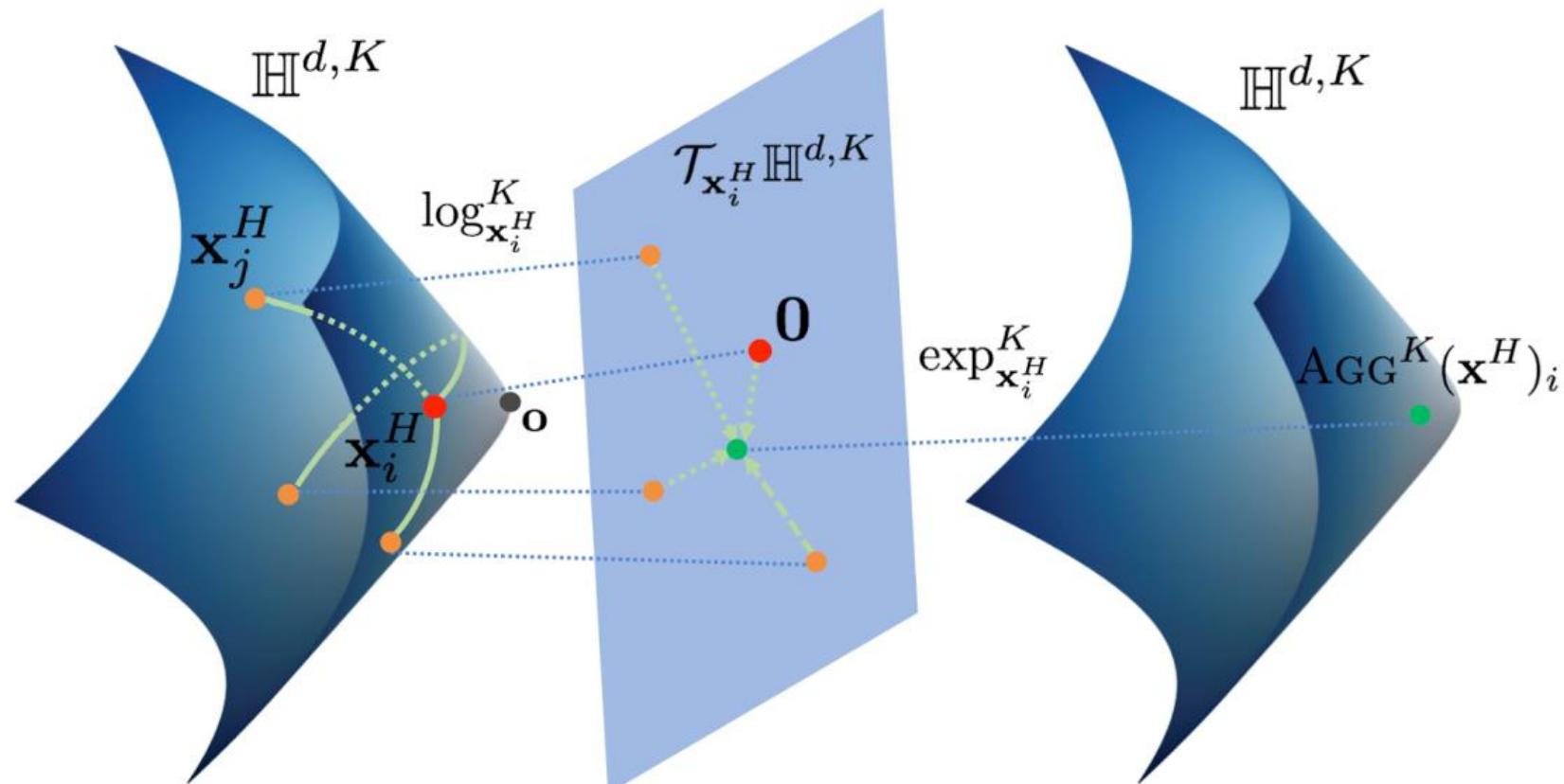


# Labels Can Be: Rankings



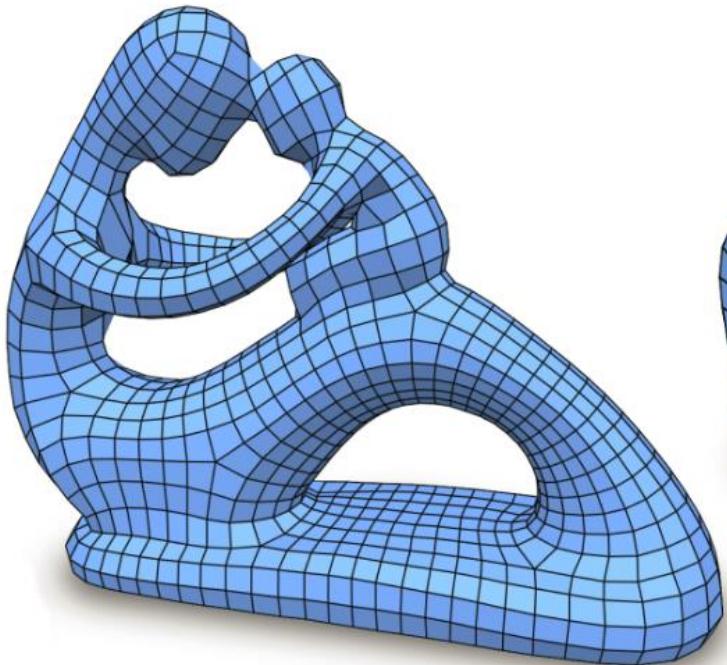
Casey Newell

# Labels Can Be: Hyperbolic Space Points

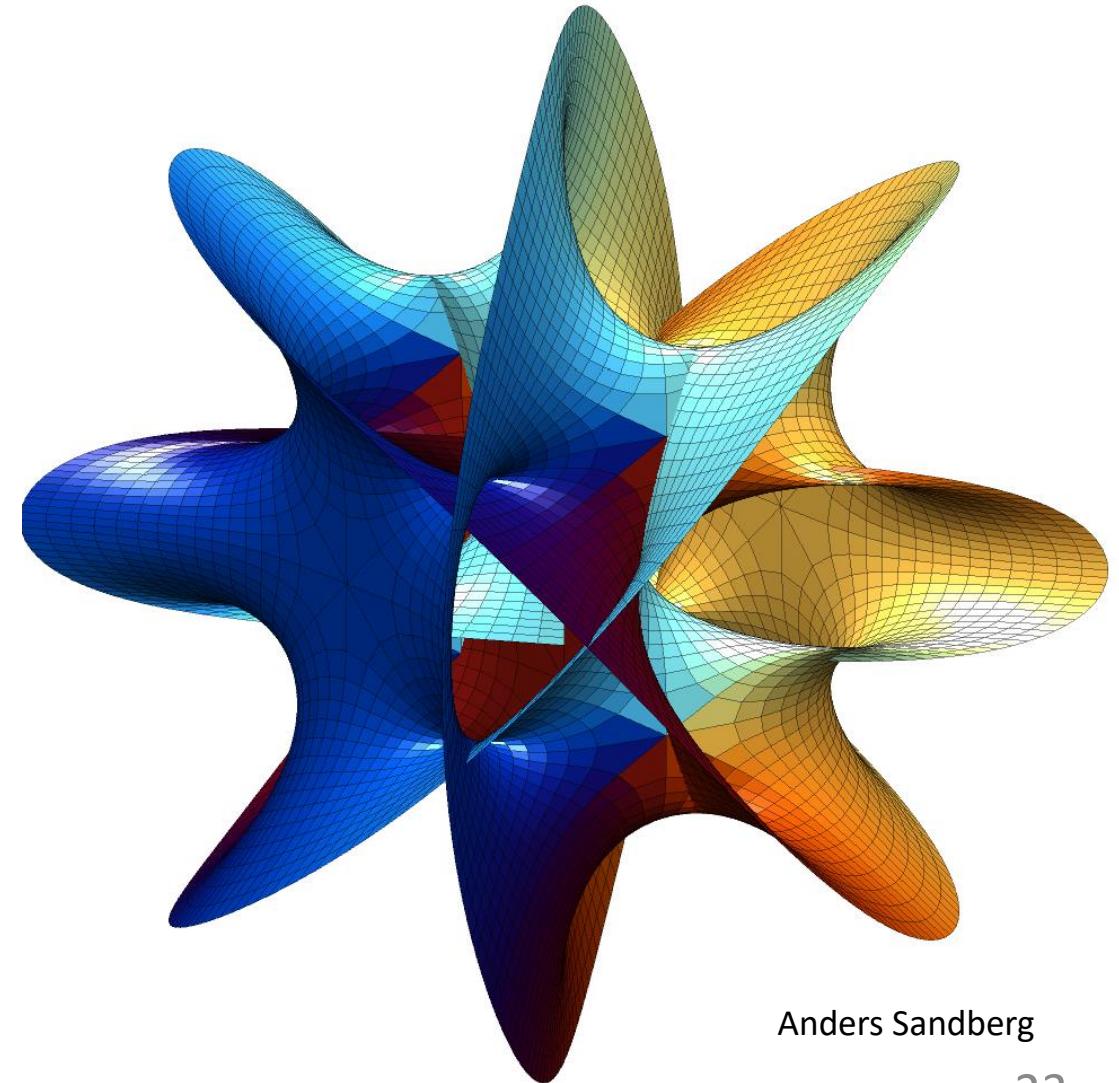
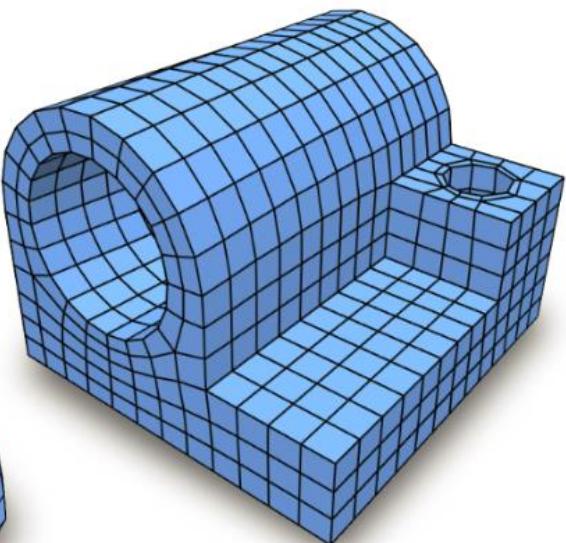


Hyperbolic Graph Convolution Networks, NeurIPS '19

# Labels Can Be: Manifold Points

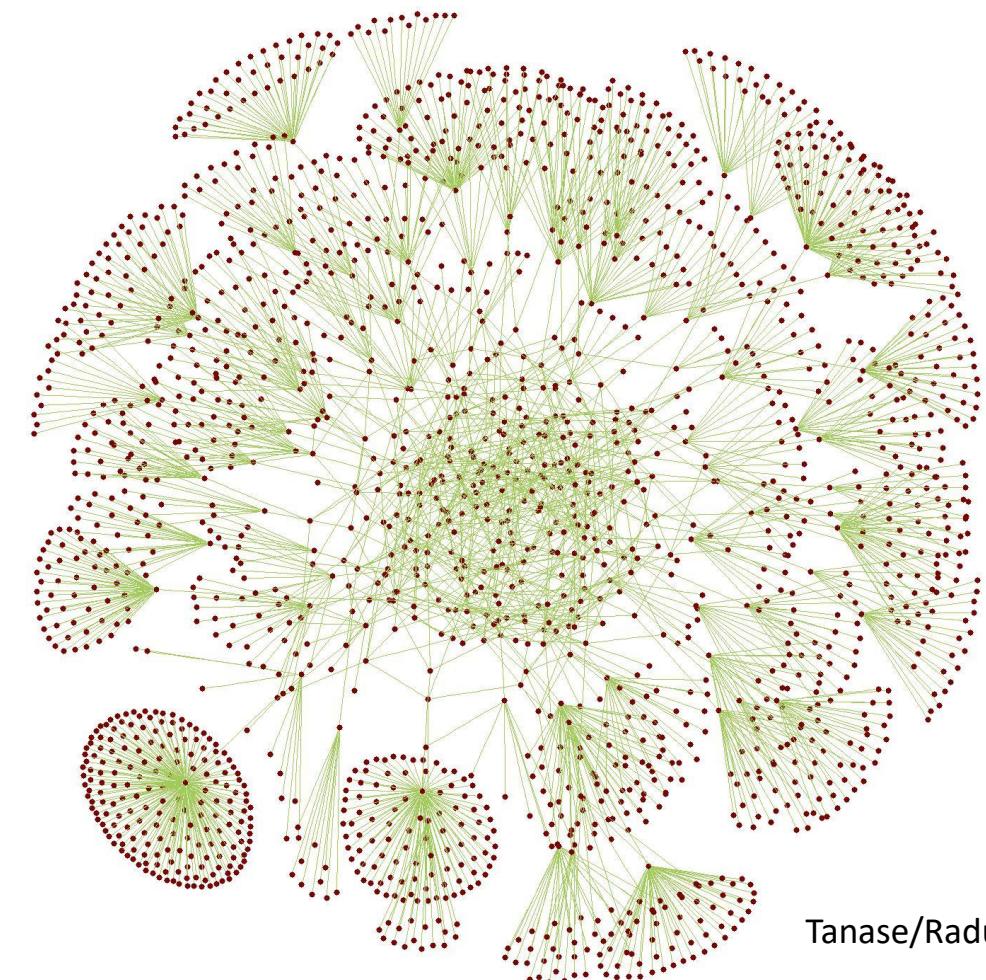
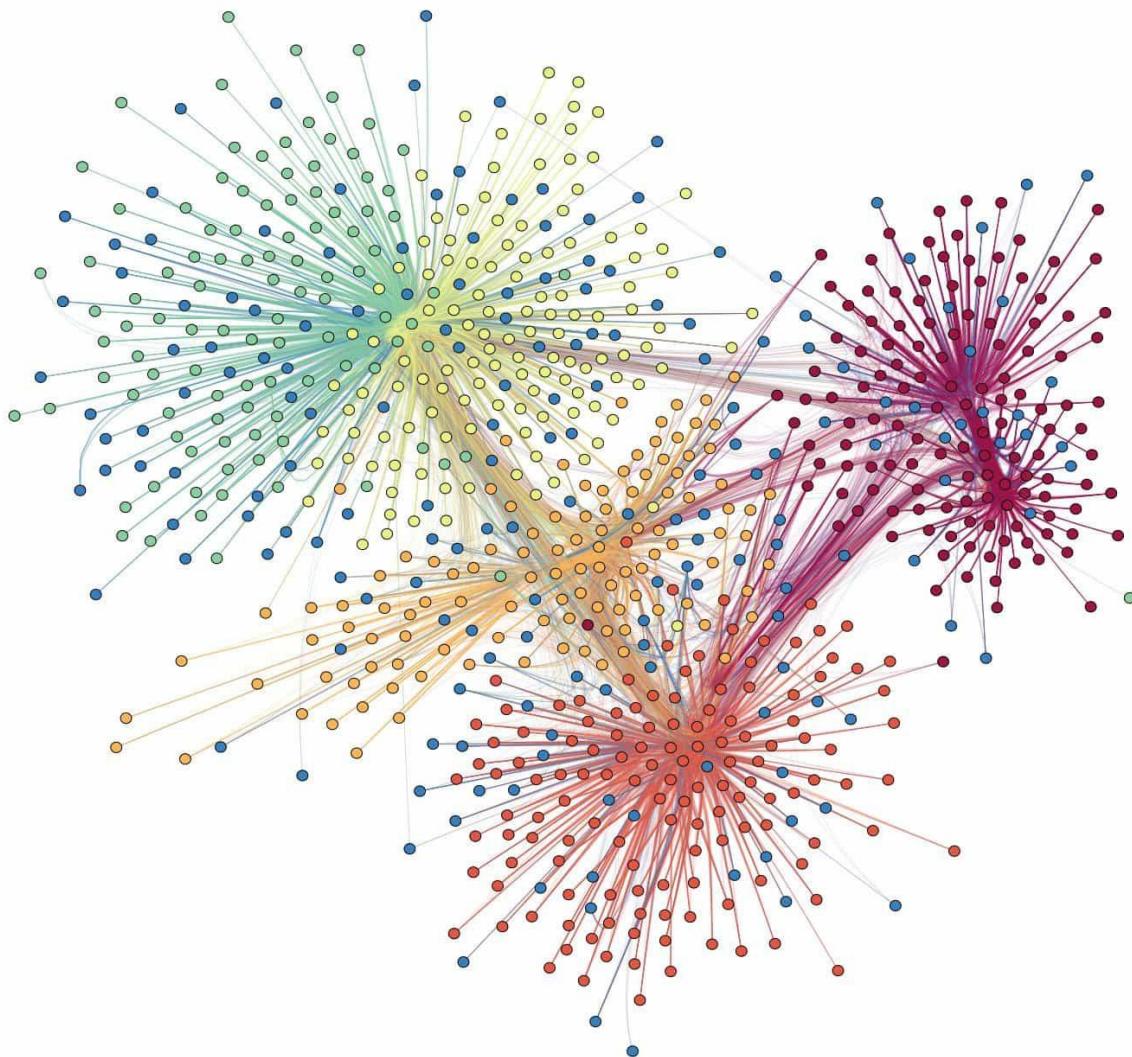


Marcel Campen



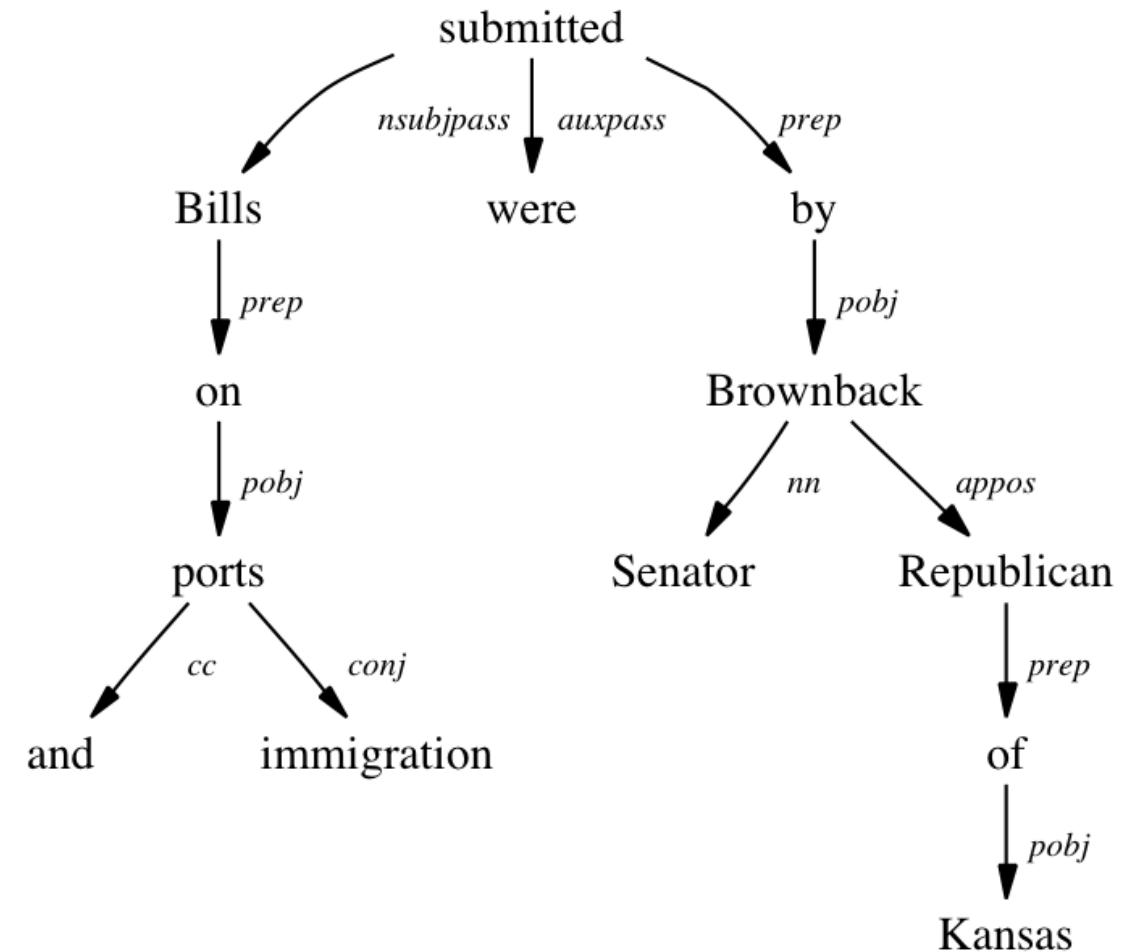
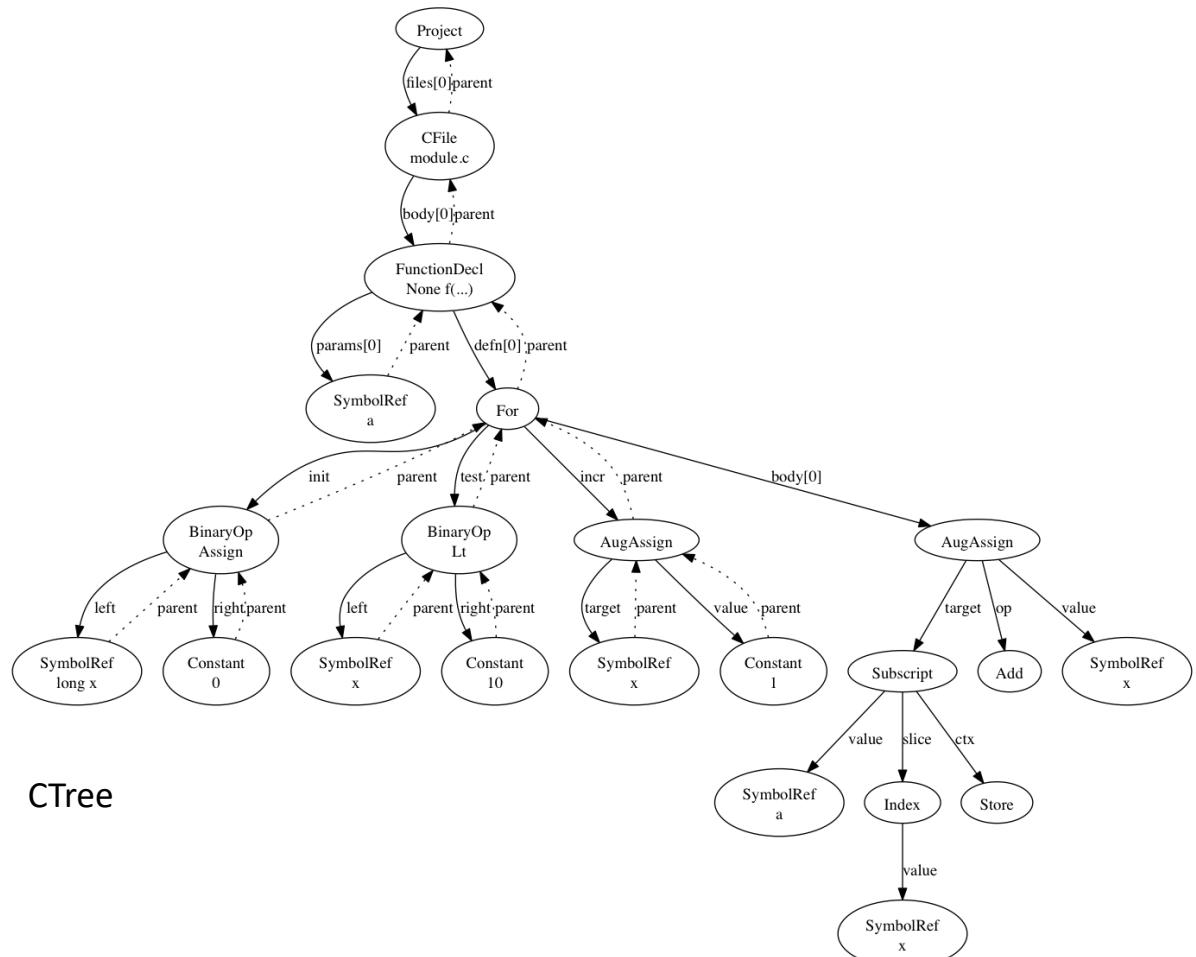
Anders Sandberg

# Labels Can Be: Graphs



Tanase/Radu

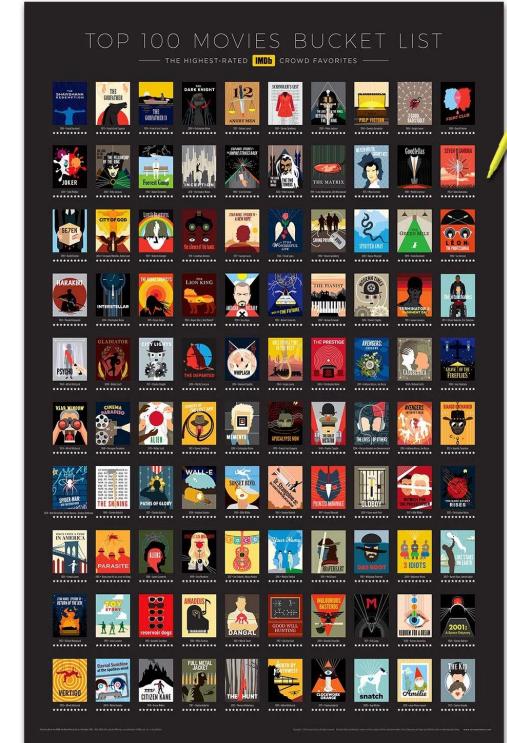
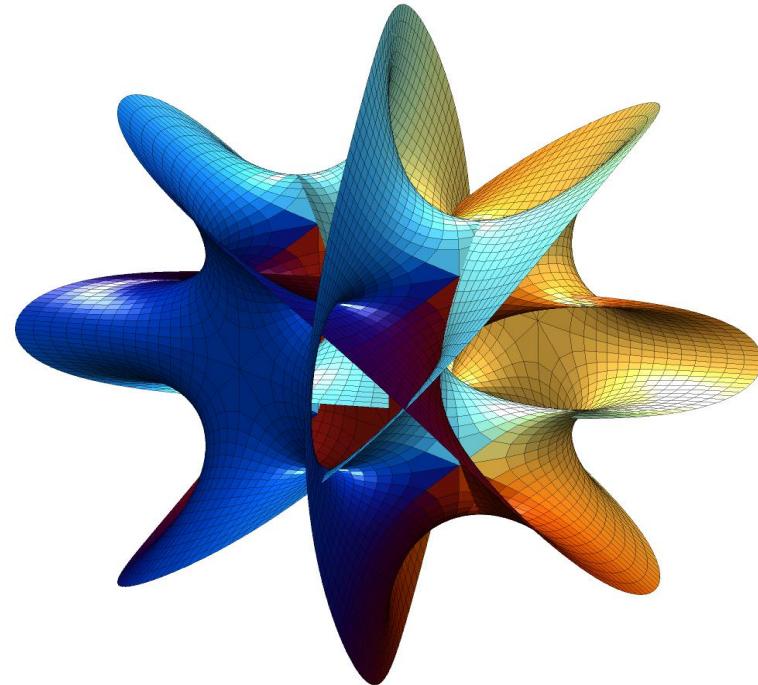
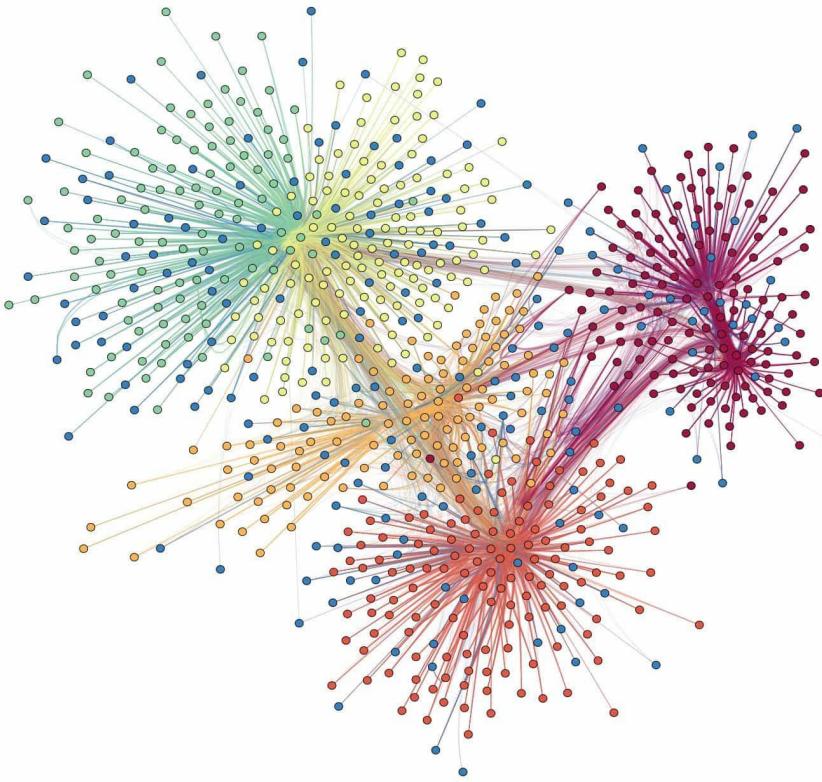
# Labels Can Be: Trees



# How Do We Do This for Diverse Y's?

# What does this multiplication even mean?

$$\mathbb{E}[\lambda_1 Y]$$



# Another Way of Encoding Accuracy

Common to our labels: a way to measure **distance**

- Ranked lists: various ways to measure closeness
- Manifolds: often equipped with a distance
- Graphs: edges in common
- Space with a distance: **metric space**

$$\frac{1}{Z} \exp \left( \underbrace{\theta_i \lambda_i Y}_{\text{Binary accuracy term}} \right) \rightarrow \frac{1}{Z} \exp \left( \theta_i d \left( \underbrace{\lambda_i, Y}_{\text{General accuracy term}} \right) \right)$$

# Distances Generalize Majority Vote

Before modeling accuracies... what's a **majority vote**?

- All weak outputs might be different!  $(1, 2, 3, 4, 5)$   
 $(2, 1, 3, 4, 5)$
- Need to use distance... the natural choice:  $(3, 2, 1, 4, 5)$   
 $(3, 2, 4, 1, 5)$

$$\hat{Y} = \arg \min_y \sum_{i=1}^m d(y, \lambda_i)$$

# How Do We Find Accuracies?

Need more relationships between latent terms and observables...

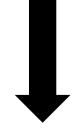
$$\mathbb{E}[\lambda_1 \lambda_2] = \mathbb{E}[\lambda_1 Y] \mathbb{E}[\lambda_2 Y]$$



**Observable**: Rate of agreement



**Accuracy Parameters**



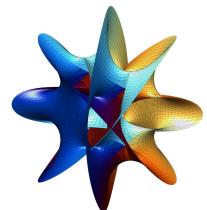
$$\mathbb{E}[d(\lambda_1, \lambda_2)] = \mathbb{E}[d(\lambda_1, Y)] + \mathbb{E}[d(\lambda_2, Y)]$$

Needs strong assumptions... alternatives exist

# Some of Our Work



- **Snorkel MeTaL**: Training complex models with multi-task weak supervision, RHDSPR, [AAAI '19](#)
- Multi-Resolution Weak Supervision for Sequential Data, [SVSFFKRXFPR](#), [NeurIPS '19](#)
- **FlyingSquid**: Fast and three-rious: Speeding up weak supervision with triplet methods, FCSHFR, [ICML '20](#)
- Comparing the value of labeled and unlabeled data in method-of-moments latent variable estimation”, CCMSR, [AISTATS '21](#)
- Universalizing Weak Supervision, SLVRS, [ICLR '22](#)
- **Liger**: “Shoring Up the Foundations: Fusing Model Embeddings and Weak Supervision”, CFAZSFR, '22.



# Data Bottlenecks

0.3500	0.8687	0.1690	0.9797	0.9037	0
0.1966	0.0844	0.6491	0.4389	0.8909	0
0.2511	0.3998	0.7317	0.1111	0.3342	1
0.6160	0.2599	0.6477	0.2581	0.6987	0
0.4733	0.8001	0.4509	0.4087	0.1978	1
0.3517	0.4314	0.5470	0.5949	0.0305	0
0.8308	0.9106	0.2963	0.2622	0.7441	0
0.5853	0.1818	0.7447	0.6028	0.5000	1
0.5497	0.2638	0.1890	0.7112	0.4799	0
0.9172	0.1455	0.6868	0.2217	0.9047	0
0.2858	0.1361	0.1835	0.1174	0.6099	0
0.7572	0.8693	0.3685	0.2967	0.6177	1
0.7537	0.5797	0.6256	0.3188	0.8594	0
0.3804	0.5499	0.7802	0.4242	0.8055	0
0.5678	0.1450	0.0811	0.5079	0.5767	0
0.0759	0.8530	0.9294	0.0855	0.1829	0
0.0540	0.6221	0.7757	0.2625	0.2399	1
0.5308	0.3510	0.4868	0.8010	0.8865	0
0.7792	0.5132	0.4359	0.0292	0.0287	1
0.9340	0.4018	0.4468	0.9289	0.4899	0
0.1299	0.0760	0.3063	0.7303	0.1679	0
0.5688	0.2399	0.5085	0.4886	0.9787	1
0.4694	0.1233	0.5108	0.5785	0.7127	1
0.0119	0.1839	0.8176	0.2373	0.5005	0
0.3371	0.2400	0.7948	0.4588	0.4711	1
0.1622	0.4173	0.6443	0.9631	0.0596	0
0.7943	0.0497	0.3786	0.5468	0.6820	0
0.3112	0.9027	0.8116	0.5211	0.0424	1
0.5285	0.9448	0.5328	0.2316	0.0714	0
0.1656	0.4909	0.3507	0.4889	0.5216	1
0.6020	0.4993	0.9390	0.6241	0.0967	0
0.2630	0.3377	0.8759	0.6791	0.8181	1
0.6541	0.9001	0.5502	0.3955	0.8175	0
0.6892	0.3692	0.6225	0.3674	0.7224	0
0.7482	0.1112	0.5870	0.9880	0.1499	0
0.4505	0.7903	0.2077	0.0377	0.6596	1
0.0838	0.3897	0.3012	0.8852	0.5186	0
0.2290	0.2417	0.4709	0.9133	0.9730	0
0.9133	0.4039	0.2305	0.7962	0.6490	1
0.1524	0.0965	0.8443	0.0987	0.8003	0
0.8258	0.1320	0.1948	0.2619	0.4538	0

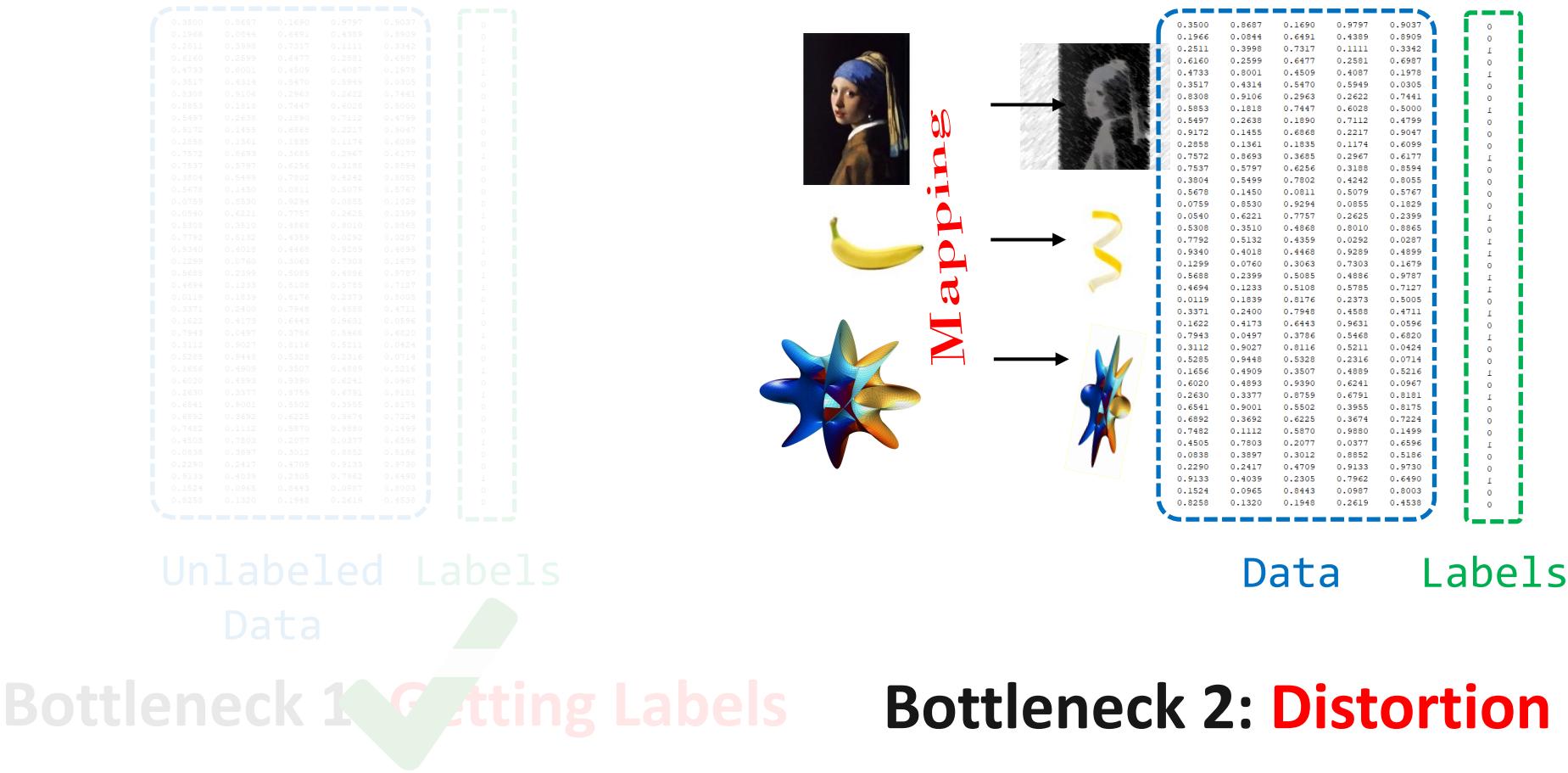
Unlabeled Labels  
Data

Bottleneck 1: Getting Labels

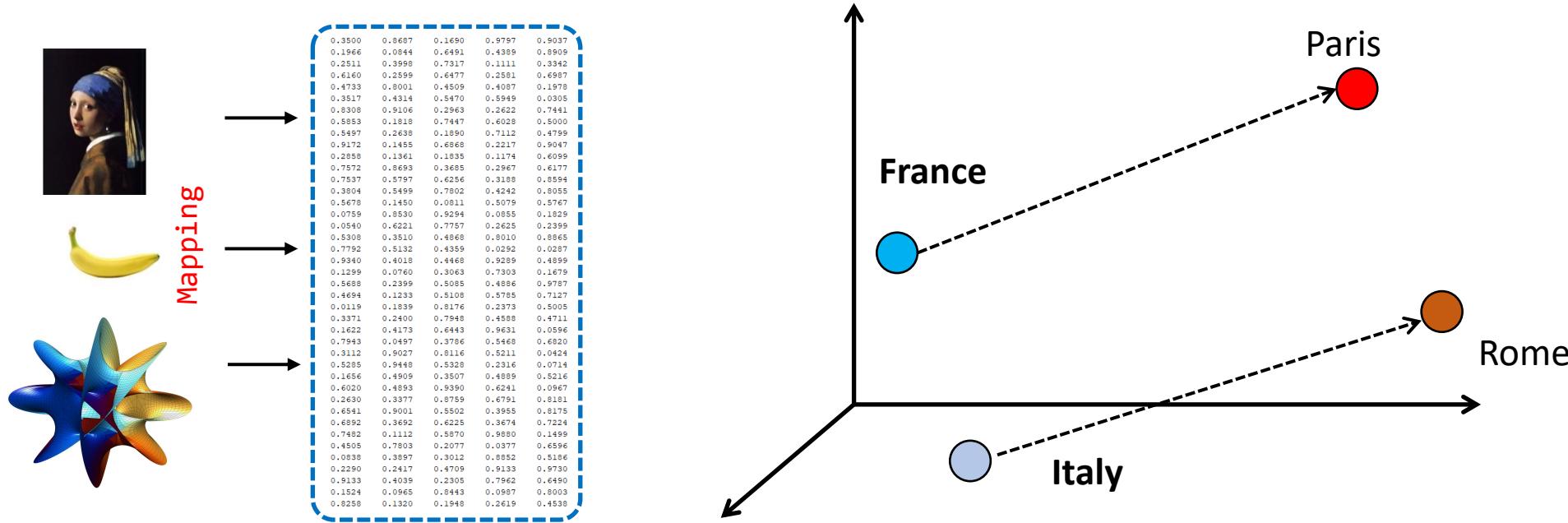


Bottleneck 2: Distortion

# Data Bottlenecks



# Embeddings



Continuous representations that  
**preserve structure & relationships**

# Preserving Relationships

Encode relationships into a **graph**

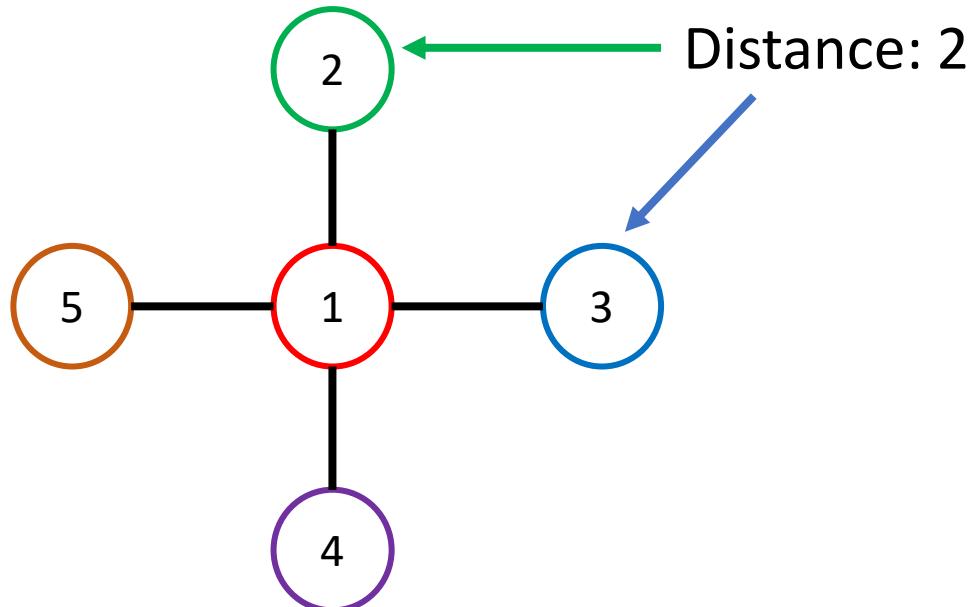
- **Hierarchical relationships:** trees
  - Ex: Artists -> Albums -> Songs
- Embed into **Euclidean space?**
  - Results in **distortion**



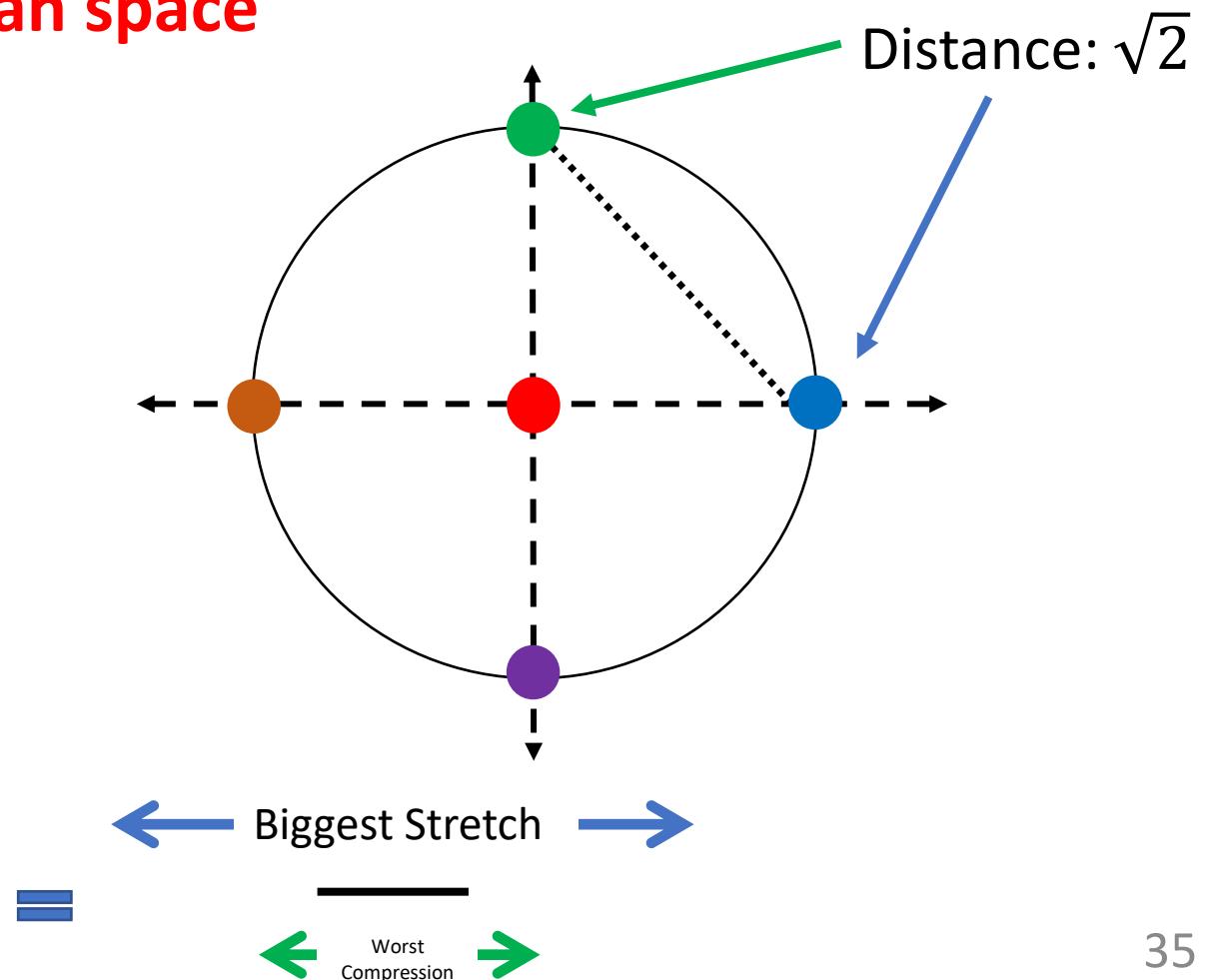
→ **Non-Euclidean Embeddings!**

# Choice of Embedding Space Matters!

- Q: Do trees embed well in **Euclidean space**

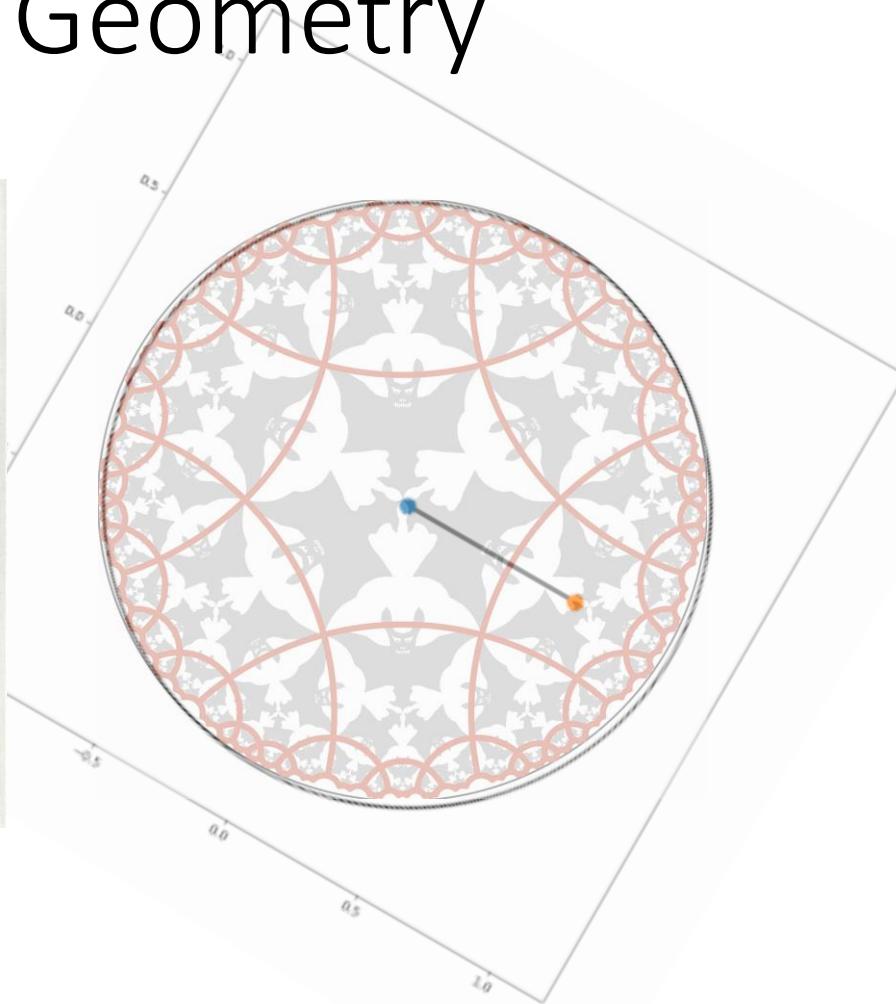
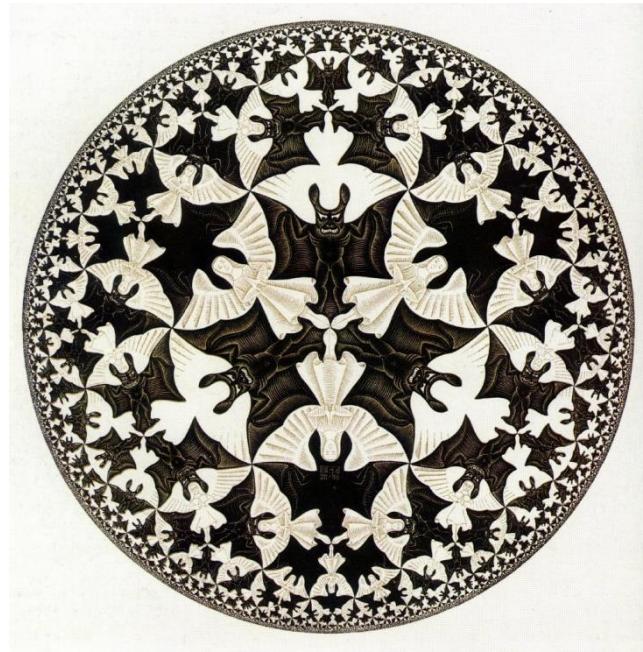


**Distortion!**



**Euclidean space distorts hierarchical relationships.**

# Hyperbolic Geometry



What are these spaces? How do we use them in ML?

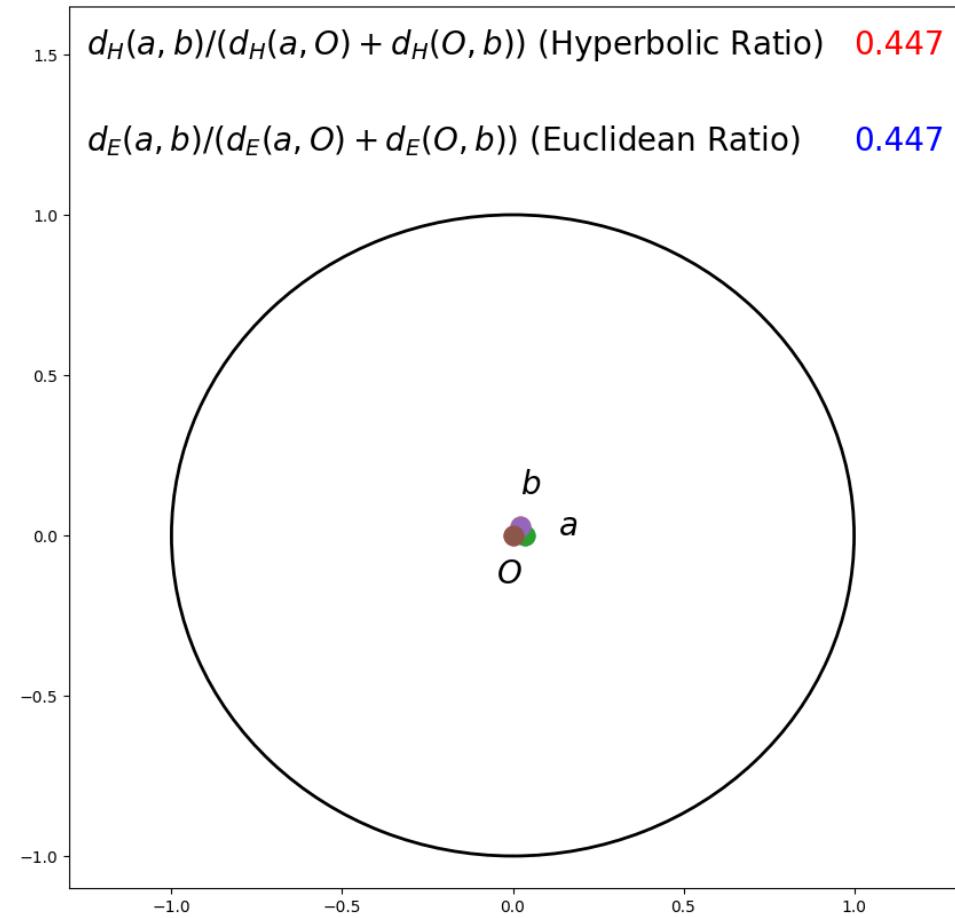
# Models, Distances, and Trees

- Poincaré **model** of hyperbolic space

- Connection to **tree distance**:
  - Hyperbolic distance:

$$d_H(x, y) = \text{acosh} \left( 1 + 2 \frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)} \right)$$

- Hyperbolics naturally represent trees!



# Embedding Trees: New Constructions

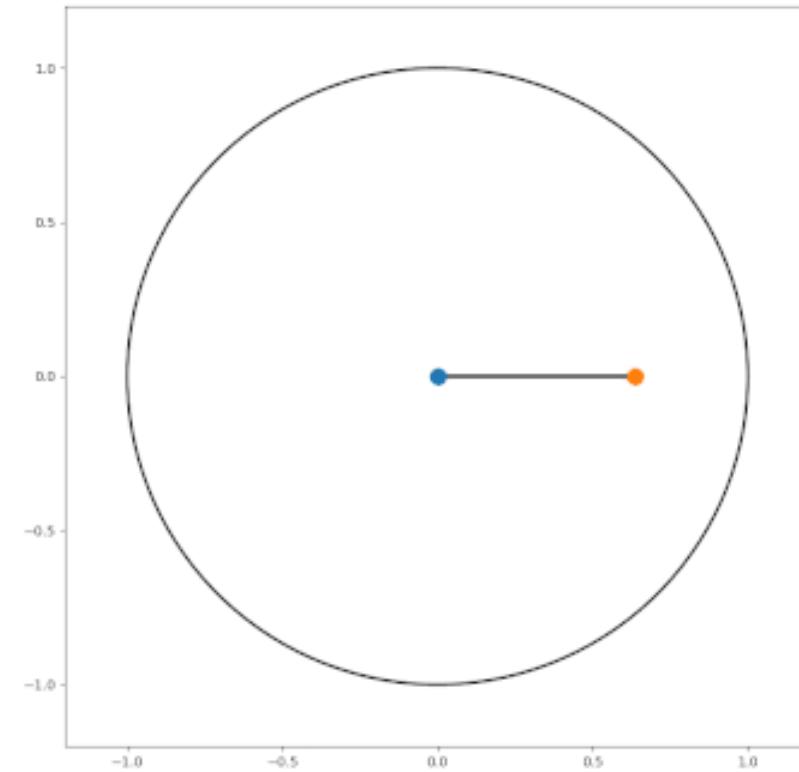
- Powerful tool for **embedding trees**

- **Arbitrarily low distortion!**



- At each node: place children into disjoint **subcones**

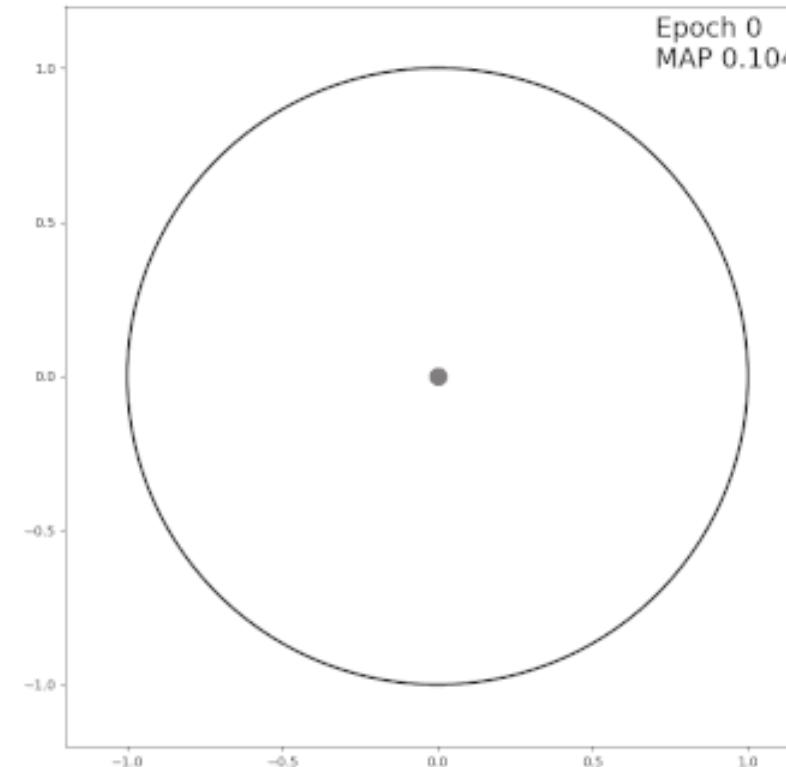
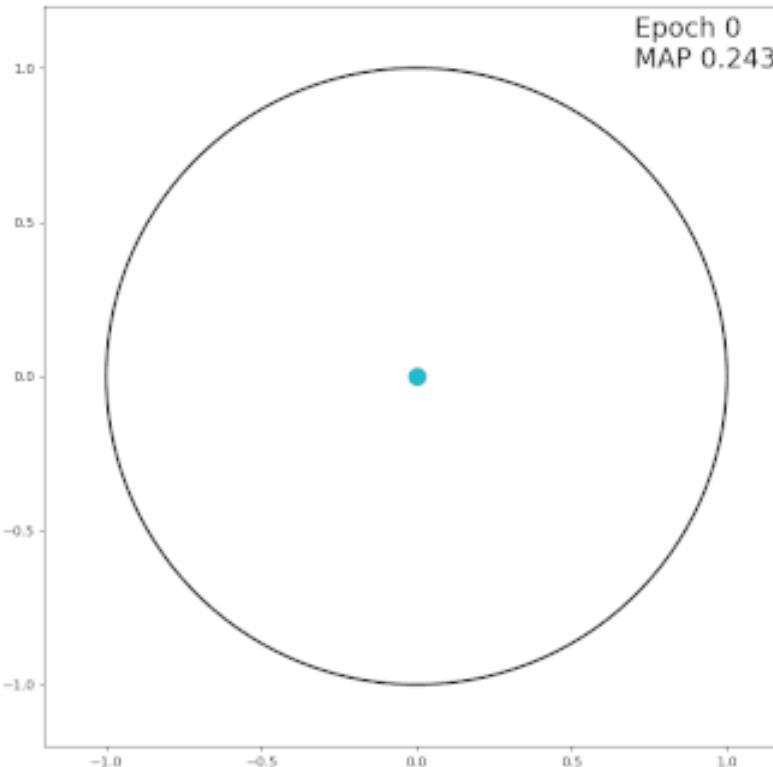
- Single global scaling factor.



Non-Euclidean embeddings: **scales matter!**

# Optimization Model

- Examples: 20 node **cycle** and **ternary tree**



**Loss:**

$$\sum_{1 \leq i < j \leq n} \left| \left( \frac{d_H(p_i, p_j)}{d_G(x_i, x_j)} \right)^2 - 1 \right|$$

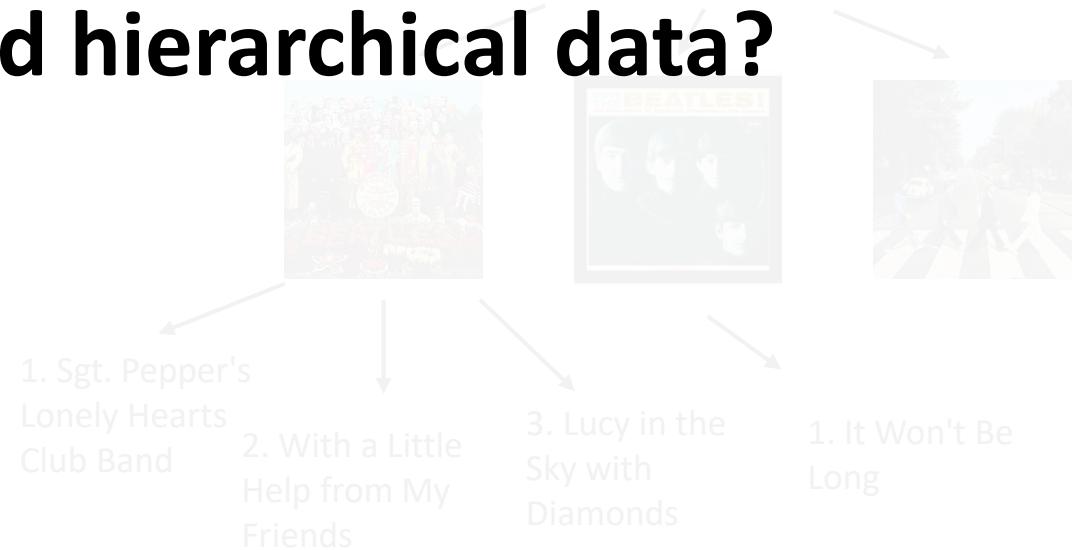
**Optimizer:**  
**Riemannian**  
**SGD**

- Hierarchical relationships: trees
  - Ex: Artists -> Albums -> Songs

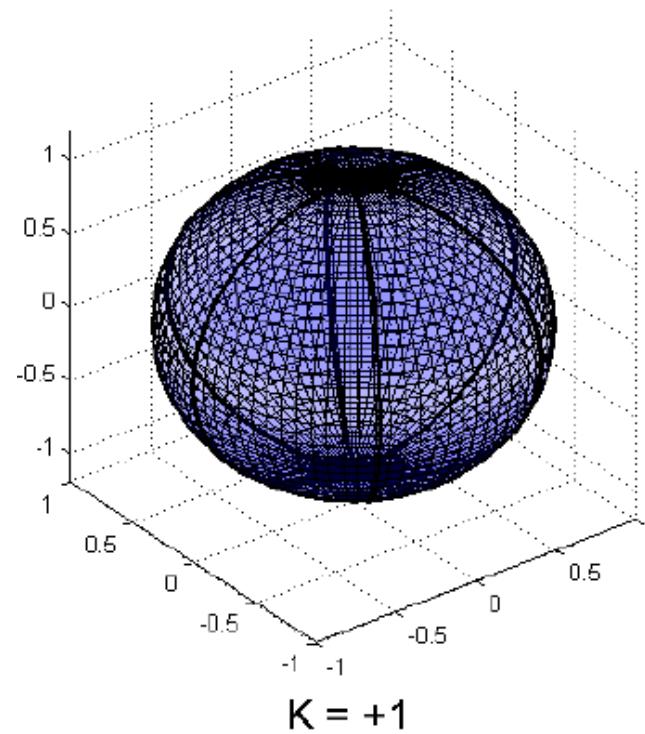


## How do we go beyond hierarchical data?

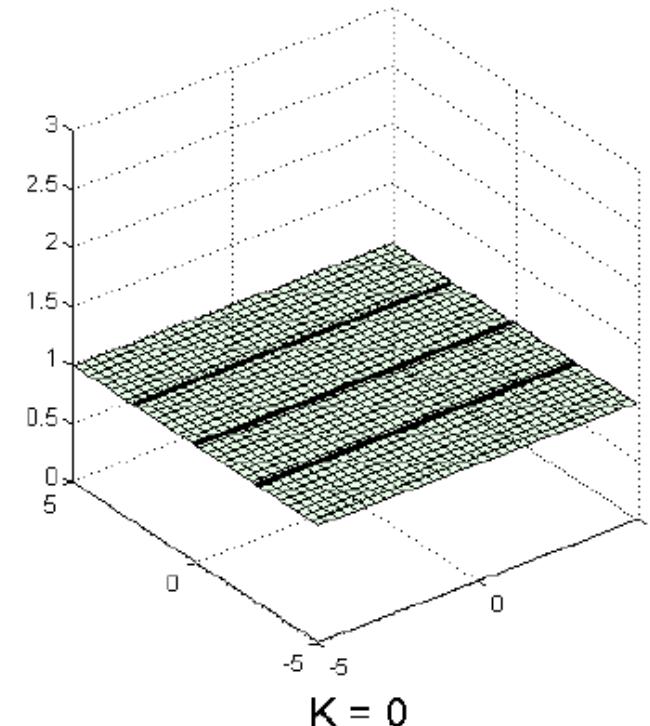
- Embed into hyperbolic space!
  - Low **distortion**
  - + **Guarantees**



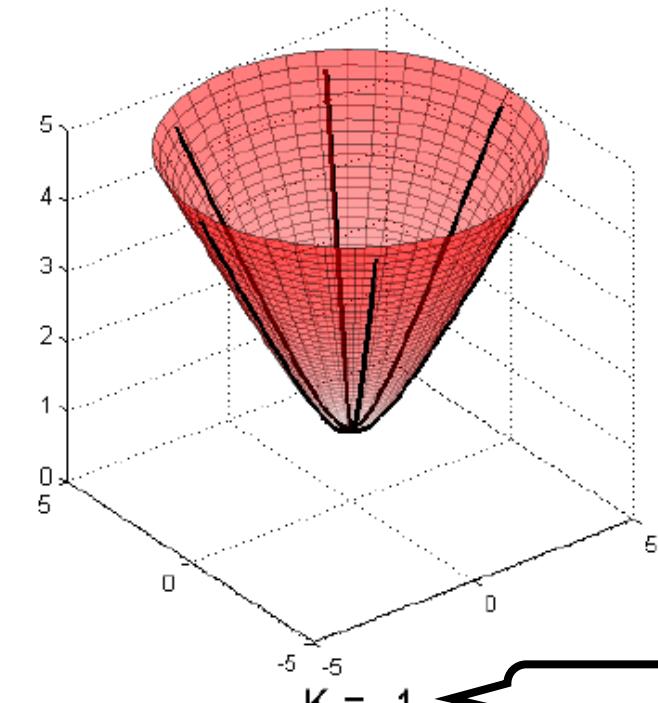
# Model Spaces



Spherical



Euclidean



Hyperbolic

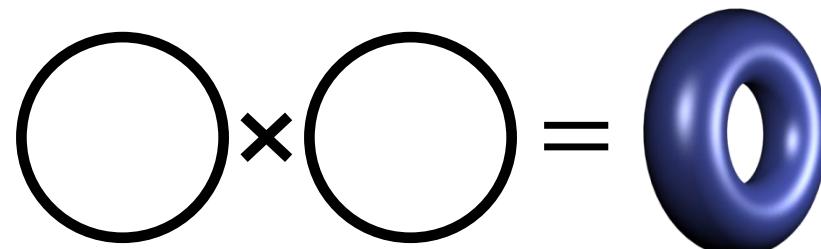
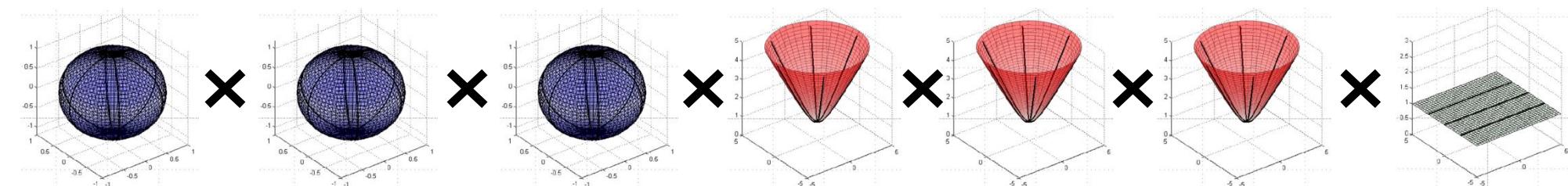
**Problem:** How do we combine these?

## 2. How to embed?

# Simple Answer: Take Products

Product manifold

$$\mathcal{P} = \mathbb{S}^{s_1} \times \mathbb{S}^{s_2} \times \cdots \times \mathbb{S}^{s_m} \times \mathbb{H}^{h_1} \times \mathbb{H}^{h_2} \times \cdots \times \mathbb{H}^{h_n} \times \mathbb{E}^e,$$

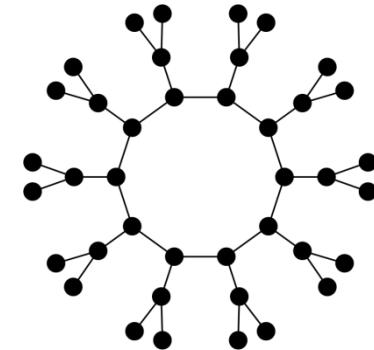
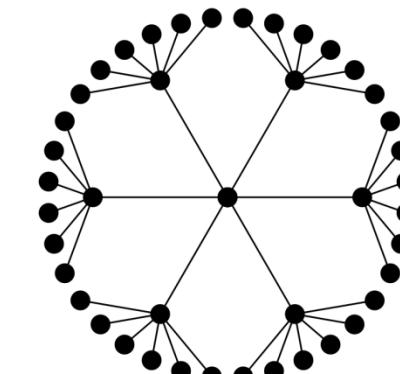
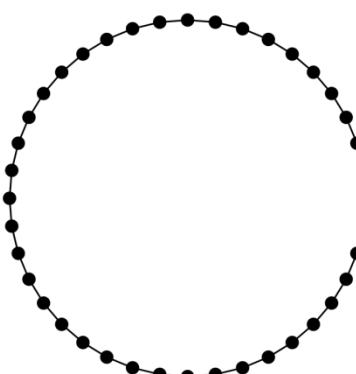


- Distances decompose:
- Easy optimization

$$d_{\mathcal{P}}^2(x, y) = \sum_{i=1}^k d_i^2(x_i, y_i)$$

## 2. How to embed?

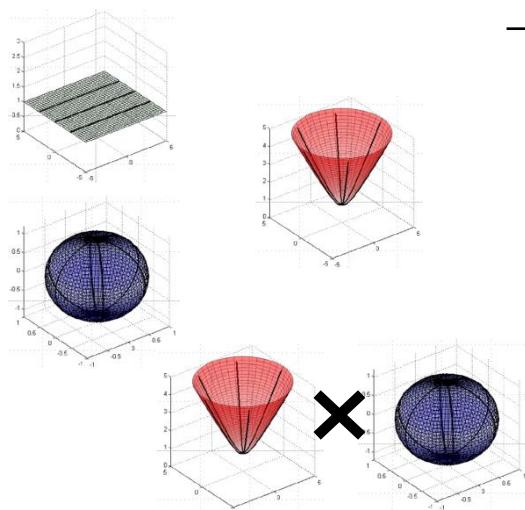
## Results



**Cycle**  
( $|V| = 40, |E| = 40$ )

**Tree**  
( $|V| = 40, |E| = 39$ )

**Ring of Trees**  
( $|V| = 40, |E| = 40$ )



$(\mathbf{E}^3)^1$

0.106

0.148

0.099

$(\mathbf{H}^3)^1$

0.164

**0.032**

0.080

$(\mathbf{S}^3)^1$

**0.001**

0.161

0.111

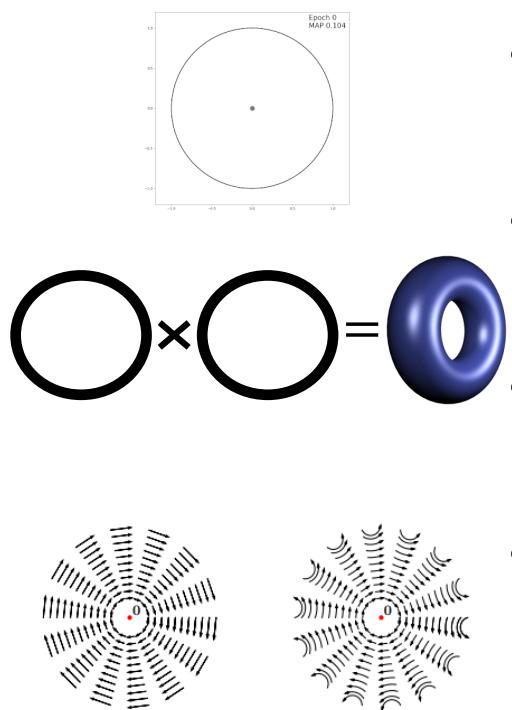
$(\mathbf{H}^3)^1 \times (\mathbf{S}^3)^1$

0.111

0.054

**0.062**

# Some of Our Work



- Representation tradeoffs for hyperbolic embeddings, RHDSPR, **ICML'18**
- Learning mixed-curvature representations in product spaces, GSGR, **ICLR '19**
- Hyperbolic graph convolutional neural networks, CYRL, **NeurIPS '19**
- Low-dimensional knowledge graph embeddings via hyperbolic rotations, CWSR, **NeurIPS GRL '19**
- Low-Dimensional Hyperbolic Knowledge Graph Embeddings”, CWJSRR, **ACL '20**

# Data Bottlenecks

0.3500	0.8687	0.1690	0.9797	0.9037	0
0.1966	0.0844	0.6491	0.4389	0.8909	0
0.2511	0.3998	0.7317	0.1111	0.3342	1
0.6160	0.2599	0.6477	0.2581	0.6987	0
0.4733	0.8001	0.4509	0.4087	0.1978	1
0.3517	0.4314	0.5470	0.5949	0.0305	0
0.8308	0.9106	0.2963	0.2622	0.7441	0
0.5853	0.1818	0.7447	0.6028	0.5000	1
0.5497	0.2638	0.1890	0.7112	0.4799	0
0.9172	0.1455	0.6868	0.2217	0.9047	0
0.2858	0.1361	0.1835	0.1174	0.6099	0
0.7572	0.8693	0.3685	0.2967	0.6177	1
0.7537	0.5797	0.6256	0.3188	0.8594	0
0.3804	0.5499	0.7802	0.4242	0.8055	0
0.5678	0.1450	0.0811	0.5079	0.5767	0
0.0759	0.8530	0.9294	0.0855	0.1829	0
0.0540	0.6221	0.7757	0.2625	0.2399	1
0.5308	0.3510	0.4868	0.8010	0.8865	0
0.7792	0.5132	0.4359	0.0292	0.0287	1
0.9340	0.4018	0.4468	0.9289	0.4899	0
0.1299	0.0760	0.3063	0.7303	0.1679	0
0.5688	0.2399	0.5085	0.4886	0.9787	1
0.4694	0.1233	0.5108	0.5785	0.7127	1
0.0119	0.1839	0.8176	0.2373	0.5005	0
0.3371	0.2400	0.7948	0.4588	0.4711	1
0.1622	0.4173	0.6443	0.9631	0.0596	0
0.7943	0.0497	0.3786	0.5468	0.6820	1
0.3112	0.9027	0.8116	0.5211	0.0424	0
0.5285	0.9448	0.5328	0.2316	0.0714	0
0.1656	0.4909	0.3507	0.4809	0.5216	1
0.6020	0.4893	0.9390	0.6241	0.0967	0
0.2630	0.3377	0.8759	0.6791	0.8181	1
0.6541	0.9001	0.5502	0.3955	0.8175	0
0.6892	0.3692	0.6225	0.3674	0.7224	0
0.7482	0.1112	0.5870	0.9880	0.1499	0
0.4505	0.7803	0.2077	0.0377	0.6596	1
0.0838	0.3897	0.3012	0.8852	0.5186	0
0.2290	0.2417	0.4709	0.9133	0.9730	0
0.9133	0.4039	0.2305	0.7962	0.6490	1
0.1524	0.0965	0.8443	0.0987	0.8003	0
0.8258	0.1320	0.1948	0.2619	0.4538	0

Unlabeled  
Labels  
Data

Bottleneck 1: Getting Labels



Mappings

0.3500	0.8687	0.1690	0.9797	0.9037	0
0.1966	0.0844	0.6491	0.4389	0.8909	0
0.2511	0.3998	0.7317	0.1111	0.3342	1
0.6160	0.2599	0.6477	0.2581	0.6987	0
0.4733	0.8001	0.4509	0.4087	0.1978	1
0.3517	0.4314	0.5470	0.5949	0.0305	0
0.8308	0.9106	0.2963	0.2622	0.7441	0
0.5853	0.1818	0.7447	0.6028	0.5000	1
0.5497	0.2638	0.1890	0.7112	0.4799	0
0.9172	0.1455	0.6868	0.2217	0.9047	0
0.2858	0.1361	0.1835	0.1174	0.6099	0
0.7572	0.8693	0.3685	0.2967	0.6177	1
0.7537	0.5797	0.6256	0.3188	0.8594	0
0.3804	0.5499	0.7802	0.4242	0.8055	0
0.5678	0.1450	0.0811	0.5079	0.5767	0
0.0759	0.8530	0.9294	0.0855	0.1829	0
0.0540	0.6221	0.7757	0.2625	0.2399	1
0.5308	0.3510	0.4868	0.8010	0.8865	0
0.7792	0.5132	0.4359	0.0292	0.0287	1
0.9340	0.4018	0.4468	0.9289	0.4899	0
0.1299	0.0760	0.3063	0.7303	0.1679	0
0.5688	0.2399	0.5085	0.4886	0.9787	1
0.4694	0.1233	0.5108	0.5785	0.7127	1
0.0119	0.1839	0.8176	0.2373	0.5005	0
0.3371	0.2400	0.7948	0.4588	0.4711	1
0.1622	0.4173	0.6443	0.9631	0.0596	0
0.7943	0.0497	0.3786	0.5468	0.6820	1
0.3112	0.9027	0.8116	0.5211	0.0424	0
0.5285	0.9448	0.5328	0.2316	0.0714	0
0.1656	0.4909	0.3507	0.4809	0.5216	1
0.6020	0.4893	0.9390	0.6241	0.0967	0
0.2630	0.3377	0.8759	0.6791	0.8181	1
0.6541	0.9001	0.5502	0.3955	0.8175	0
0.6892	0.3692	0.6225	0.3674	0.7224	0
0.7482	0.1112	0.5870	0.9880	0.1499	0
0.4505	0.7803	0.2077	0.0377	0.6596	1
0.0838	0.3897	0.3012	0.8852	0.5186	0
0.2290	0.2417	0.4709	0.9133	0.9730	0
0.9133	0.4039	0.2305	0.7962	0.6490	1
0.1524	0.0965	0.8443	0.0987	0.8003	0
0.8258	0.1320	0.1948	0.2619	0.4538	0

Data Labels

Bottleneck 2: Distortion

# Thank you!

## Joint Work With:

Nicholas Roberts, Changho Shin, Winfred Li, Harit Vishwakarma, Dyah Adila, Aws Albarghouthi, Ben Boecking, Chris Ré, Chris De Sa, Alex Ratner, Albert Gu, Paroma Varma, Jared Dunnmon, Ines Chami, Beliz Gunel, Dan Fu, Mayee Chen

<https://pages.cs.wisc.edu/~fredsala/>

fredsala@cs.wisc.edu