Learning Representations for Images with Hierarchical Labels

Ankit Dhall
October 2, 2019

Supervised by: Prof. Andreas Krause
Anastasia Makarova, Octavian-Eugen Ganea, Dario Pavllo

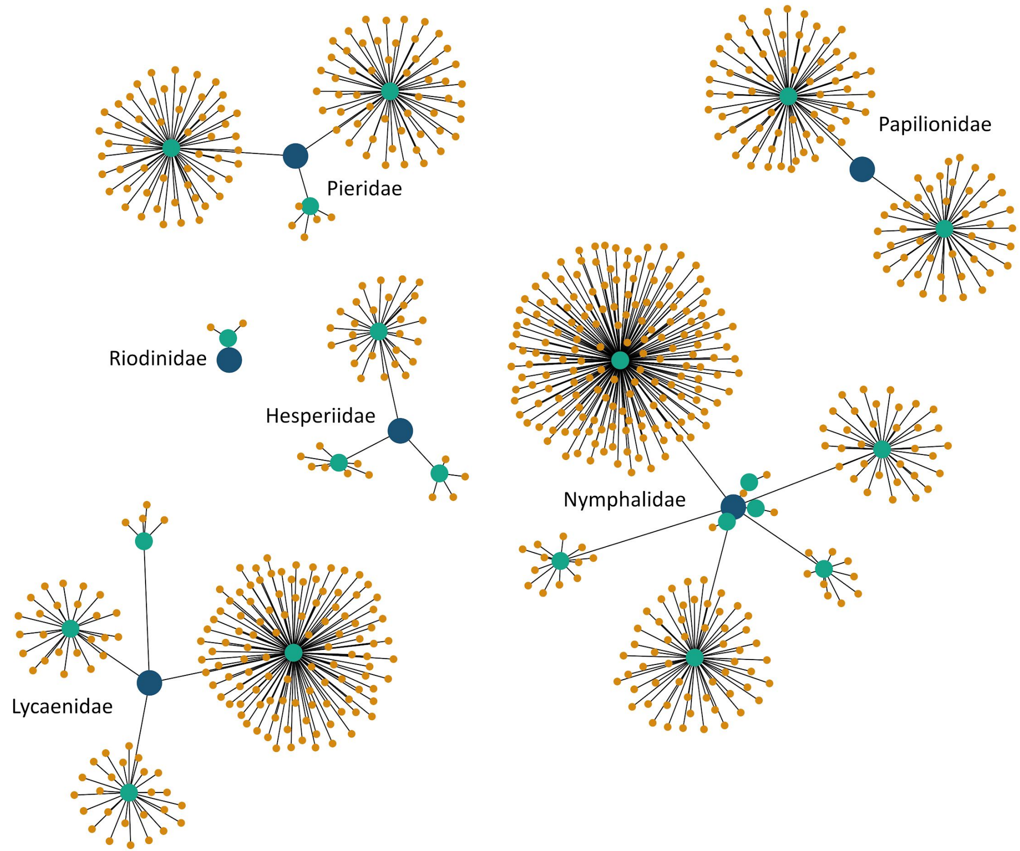
ETH Entomological Collection (ETHEC) Dataset

- 47,978 butterfly images with a 4-level label-hierarchy
- 6 *family* -> 21 *sub-family* -> 135 *genus* -> 561 *species*



family	Papilionidae	Nymphalidae	Nymphalidae
sub-family	Papilioninae	Limenitidinae	Nymphalinae
genus	Papilio	Neptis	Nymphalis
species	Papilio machaon	Neptis rivularis	Nymphalis polychloros

ETH Entomological Collection (ETHEC) Dataset



ETH Entomological Collection (ETHEC) Dataset

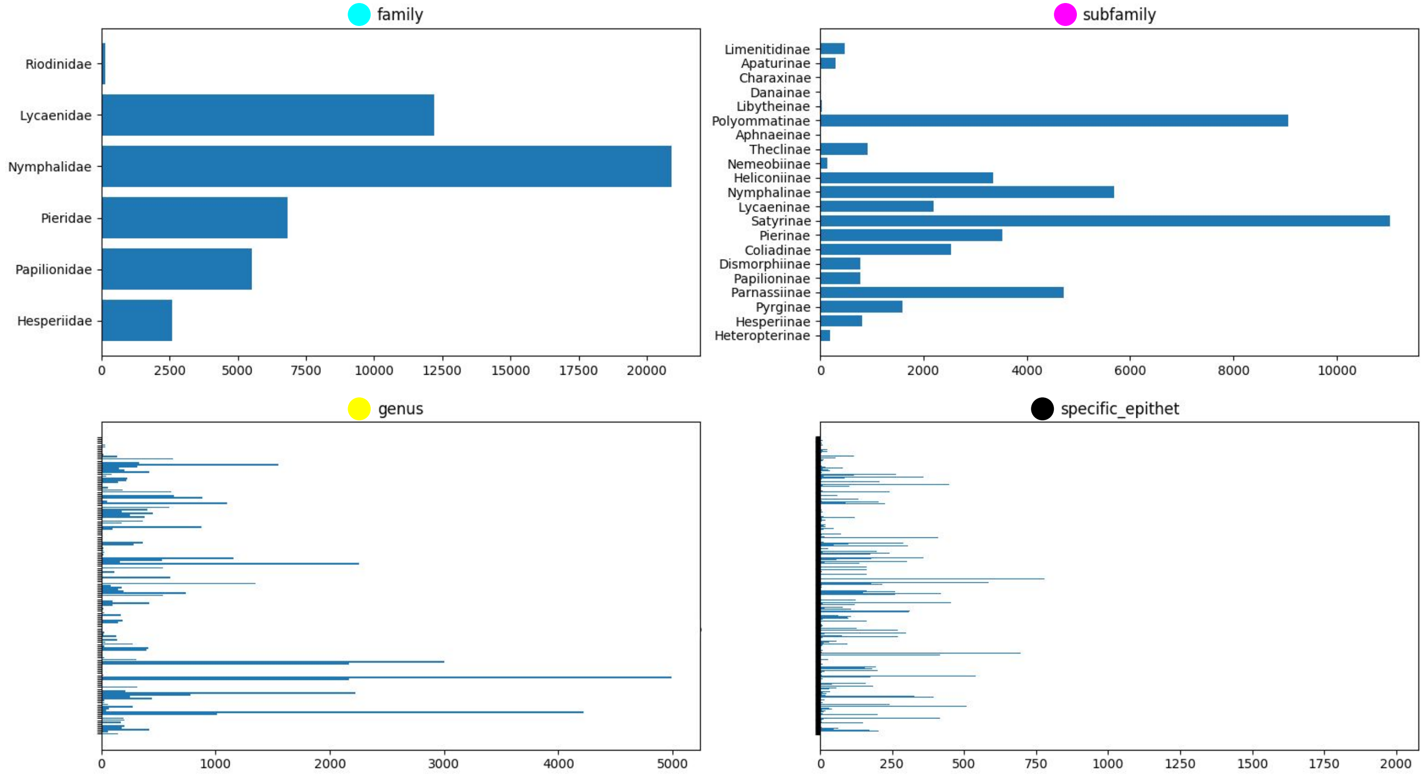
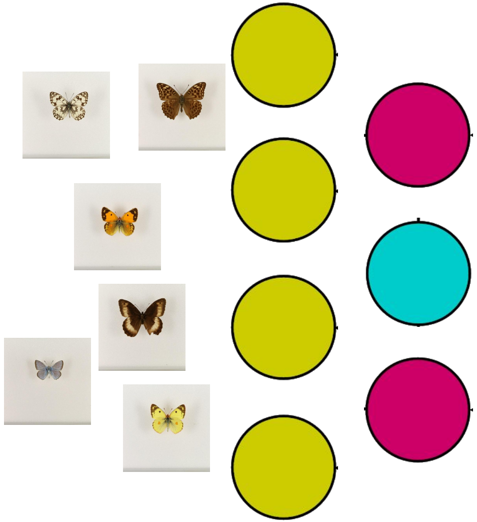


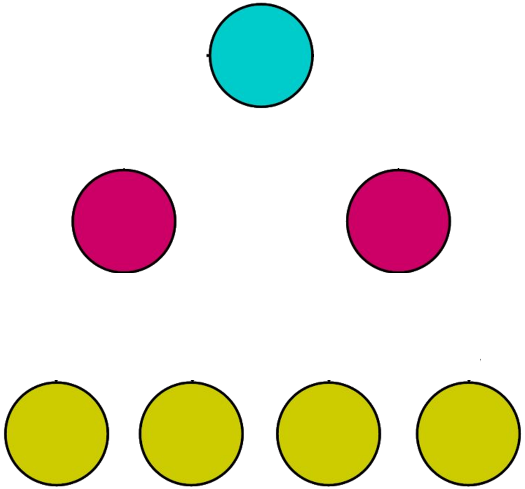
Image distribution for each label across the 4 levels.
y-axis: class label, x-axis: image frequency

Motivation

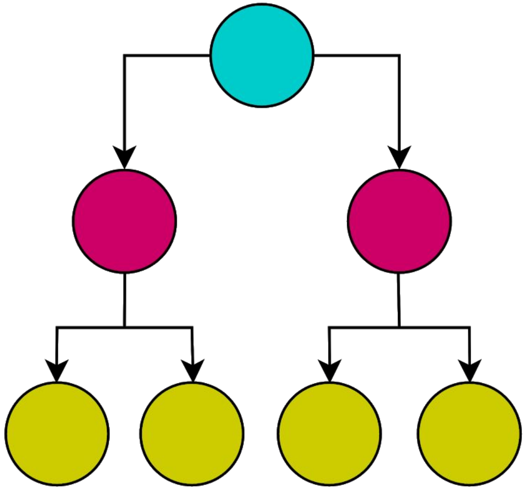
- Leveraging both label-label and label-image information for classification



Label-image information



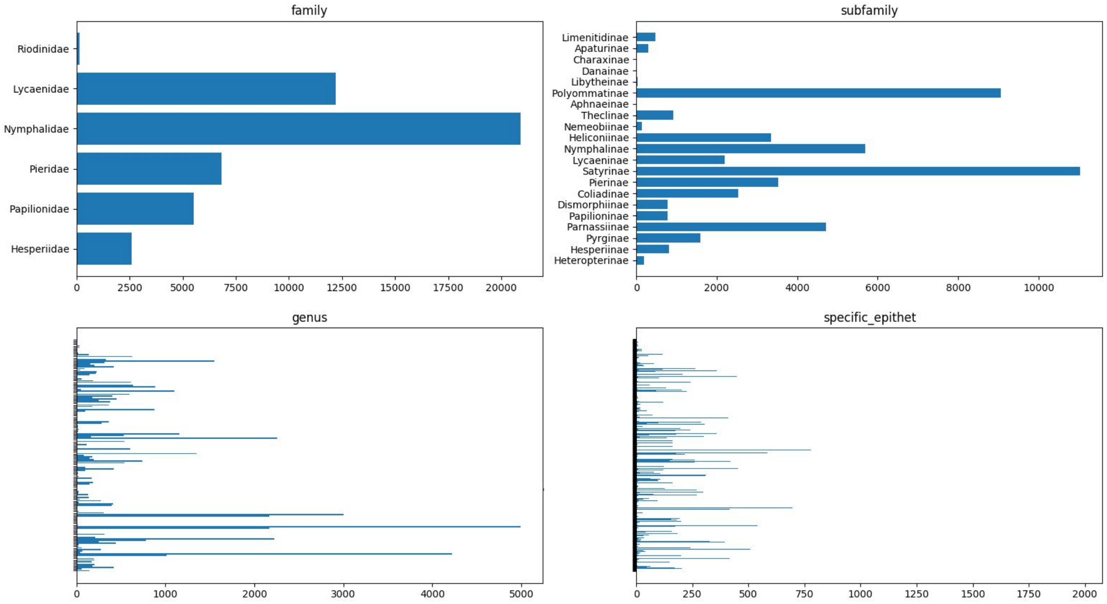
Information about levels in hierarchy

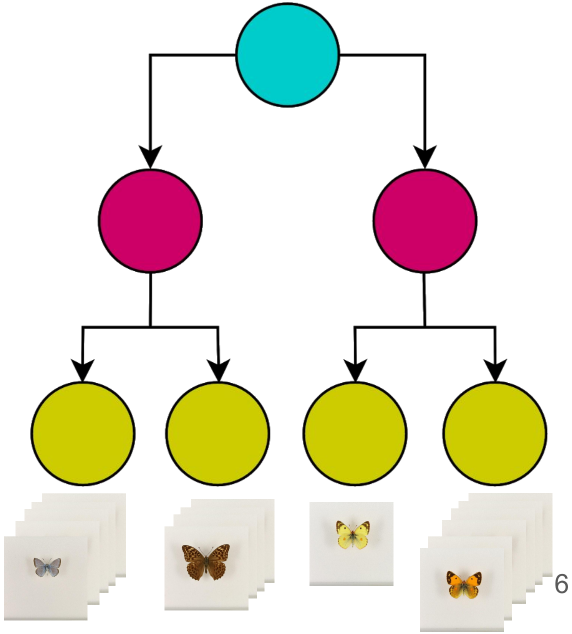


Subtree and edge relations

Motivation

- Leveraging both label-label and label-image information for classification
- Sharing information between images from unbalanced data





y-axis: class label, x-axis: image frequency

Motivation

- Leveraging both label-label and label-image information for classification
- Sharing information between images from unbalanced data
- Jointly infer visual cues (from images) and semantics (from label-hierarchy)

(a) orange

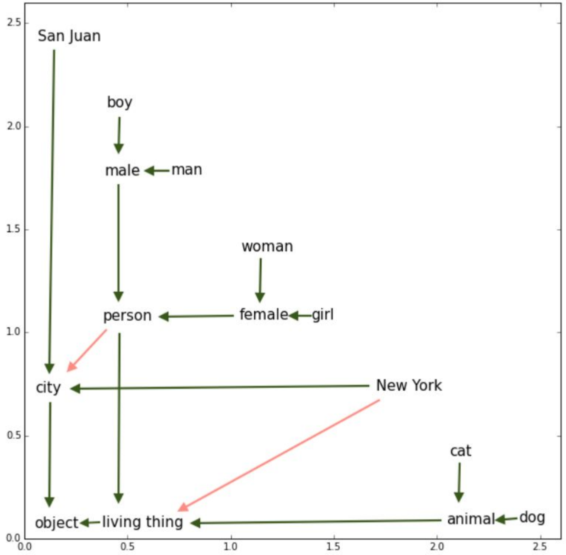
(b) clock

(c) clock

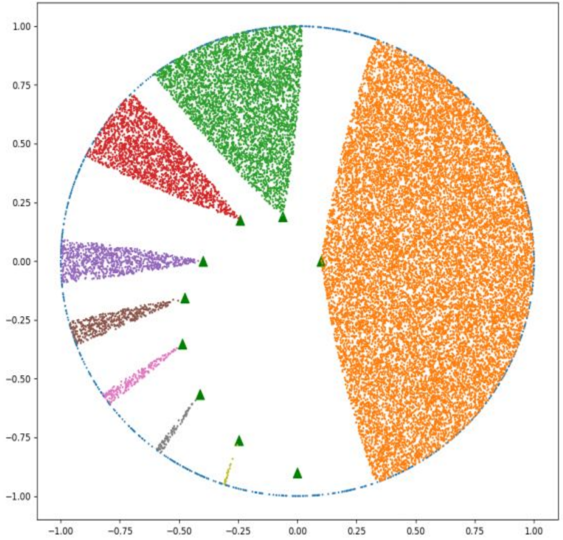
(d) clock

Related Work

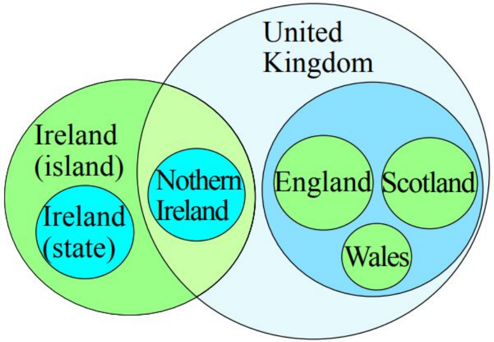
- Embedding-based models for **language** (Euclidean + non-Euclidean)



Order-embeddings



Hyperbolic entailment cones



Hyperbolic Disk Embeddings

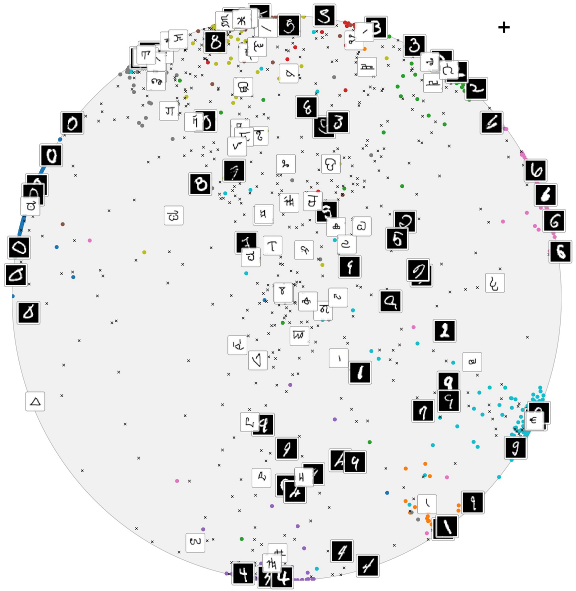
* *Order-Embeddings; I Vendrov, R Kiros, S Fidler, R Urtasun*

** *Hyperbolic Entailment Cones; OE Ganea, G Bécigneul, T Hofmann*

\+ *Hyperbolic Disk Embeddings for Directed Acyclic Graphs; R Suzuki, R Takahama, S Onoda*

Related Work

- Embedding-based models for **language** (Euclidean + non-Euclidean)
- Embedding-based models for **images**
  - Image-captioning and retrieval*
  - Zero-shot learning**
  - Hyperbolic image embeddings +



* *VSE++: Improving Visual-Semantic Embeddings with Hard Negatives; F Faghri, et al.*
** *DeViSE: A Deep Visual-Semantic Embedding Model; A Frome, et al.*
+ *Hyperbolic Image Embeddings, V Khrulkov, et al. (image source)*

Related Work

- Embedding-based models for **language** (Euclidean + non-Euclidean)
- Embedding-based models for **images**
- Convolutional Neural Networks based models (modified CNN architectures)
  - Attention-based models*
  - Predict labels for each level with a separate neural-network +

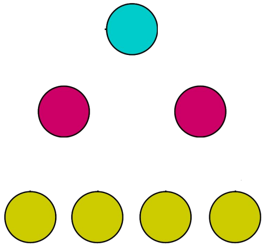
** See Better Before Looking Closer; T Hu, et al.*
+ Fine-Grained Representation Learning and Recognition by Exploiting Hierarchical Semantic Embedding, T Chen, et al.

Methods: Injecting Label-hierarchy into CNN Classifiers

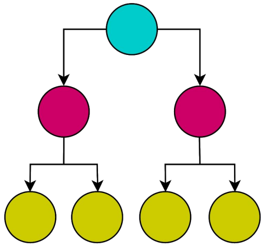
Injecting Label-hierarchy into CNN Classifiers

- Hierarchy-agnostic classifier
- Per-level classifier
- Masked Per-level classifier
- Marginalization
- Hierarchical softmax

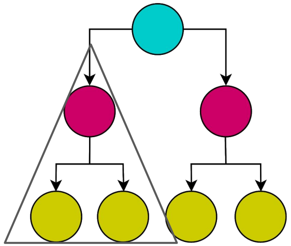
These methods provide hierarchical information at different levels of abstraction



Information about levels in hierarchy



Edge relations



Subtree relations

Experimental Setup

- Input: 224 x 224 RGB image
- Output: predicted logits for each level $\mathcal{F}(\mathcal{I}) = x = \{x_1, x_2, x_3, x_4\} \quad x_i \in \mathbb{R}^{N_i},$
- Ground-truth: 4 x labels (*family*, *subfamily, genus*, *species)* $y = \{y_1, y_2, y_3, y_4\}$

$$y_1 \in [0, N_{\text{family}} - 1], y_2 \in [0, N_{\text{subfamily}} - 1], y_3 \in [0, N_{\text{genus}} - 1], y_4 \in [0, N_{\text{species}} - 1]$$

Loss computation:

$$\mathcal{L}(x, y) = \sum_{i=1}^{L=4} \mathcal{L}_i(x_i, y_i)$$

Cross-entropy for classifying each level

Metrics:

- Precision, recall and F1-score for **each** label
- Micro and Macro averaged **global scores**

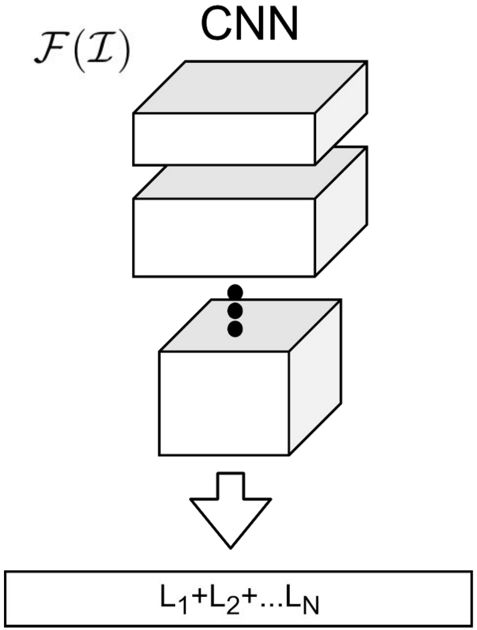
example:

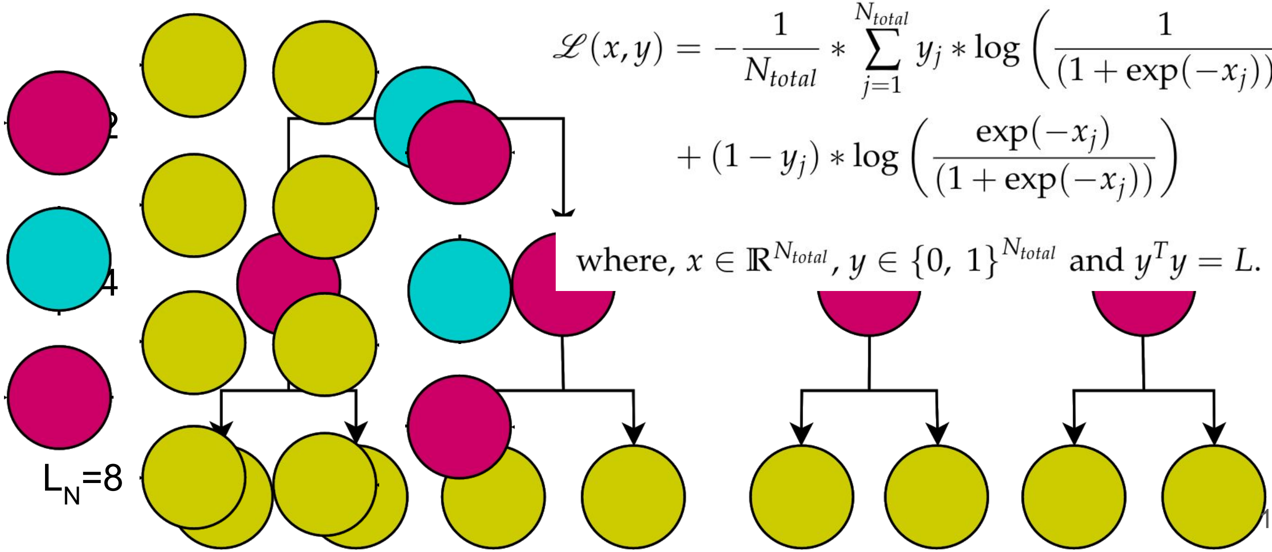
$$\text{M-Precision} = \frac{1}{N} \sum_{j=1}^{N} \text{Precision}(\text{label}_j)$$

$$\text{m-Precision} = \frac{\sum_{j=1}^{N} \text{TP}(\text{label}_j)}{\sum_{j=1}^{N} \text{TP}(\text{label}_j) + \sum_{j=1}^{N} \text{FP}(\text{label}_j)}$$

Hierarchy-agnostic Classifier

- Indifferent to the presence of label-hierarchy
- Multi-label classifier: can predict as many label as it likes



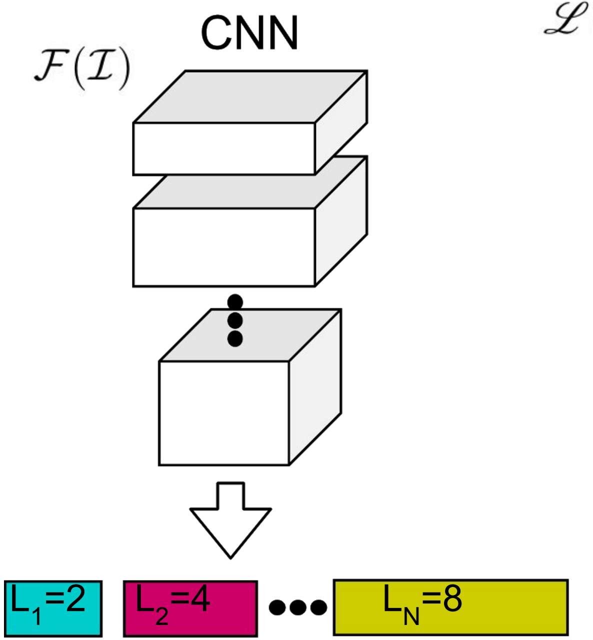


$$\mathcal{L}(x,y) = -\frac{1}{N_{total}} * \sum_{j=1}^{N_{total}} y_j * \log\left(\frac{1}{(1+\exp(-x_j))}\right) + (1-y_j) * \log\left(\frac{\exp(-x_j)}{(1+\exp(-x_j))}\right)$$

where, $x \in \mathbb{R}^{N_{total}}$, $y \in \{0,\ 1\}^{N_{total}}$ and $y^T y = L$.

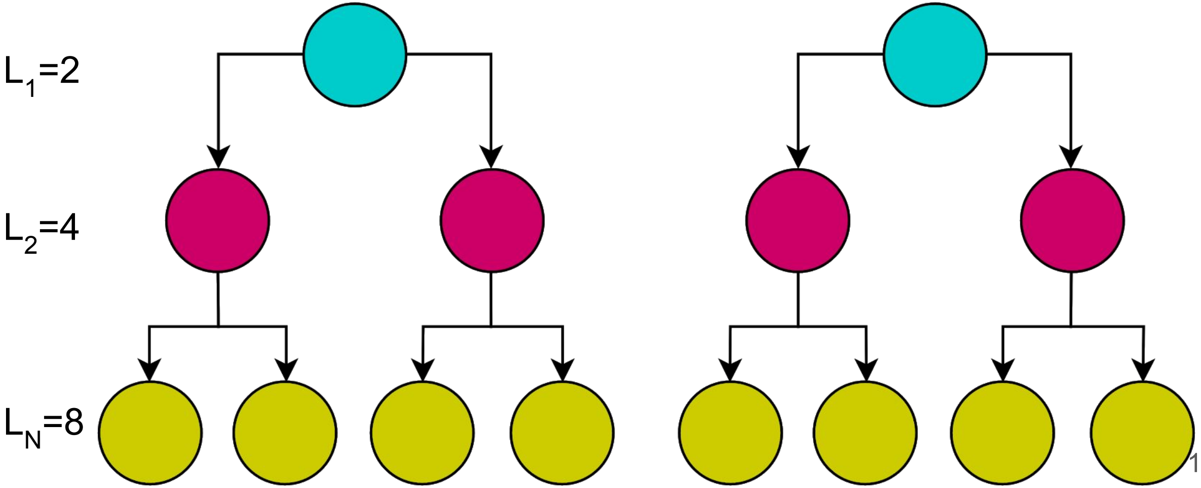
Per-level Classifier

- Exploits: number of levels in the label-hierarchy



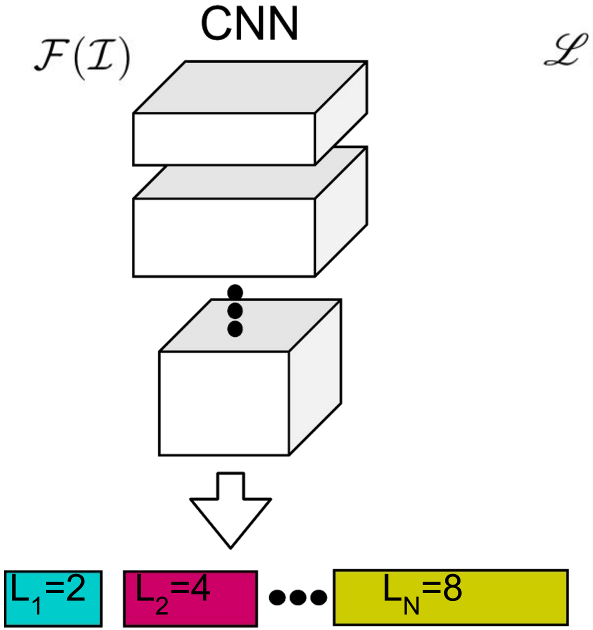
$$\mathscr{L}(x,y) = \sum_{i=1}^{L} \mathscr{L}_i(x_i, y_i) \qquad \mathscr{L}_i(x_i, y_i) = -\log\left(\frac{\exp(x_i[y_i])}{\sum_{j=1}^{L_i} \exp(x_i[j])}\right)$$

where, y_i is the true label for the *i-th* level. $x_i \in \mathbb{R}^{L_i}$, $y \in \mathbb{I}_{+}^{L}$



Masked Per-level Classifier

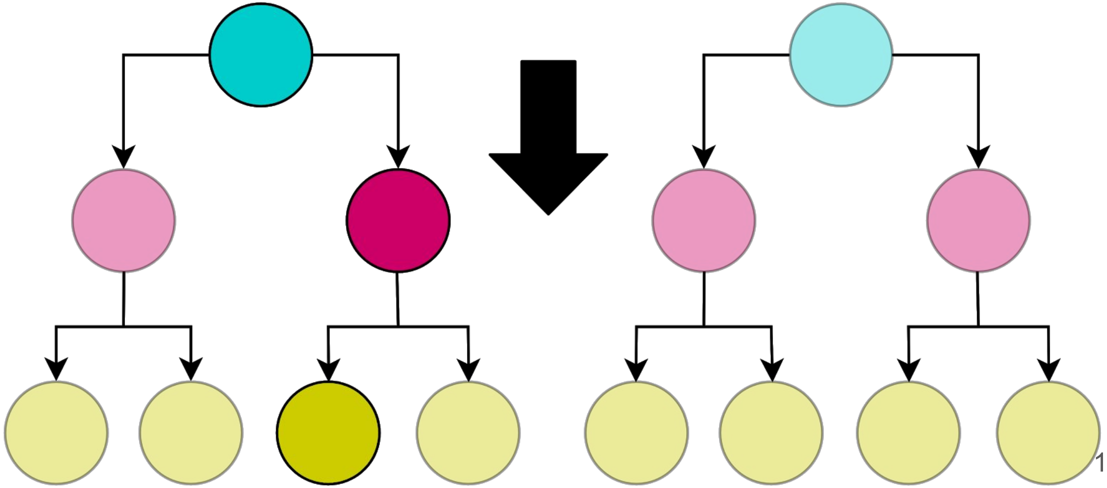
- Exploits: sub-tree relation + number of levels in label-hierarchy
- Use CNN prediction to mask implausible nodes down the hierarchy



$$\mathscr{L}(x,y) = \sum_{i=1}^{L} \mathscr{L}_i(x_i, y_i)$$

$$\mathscr{L}_i(x_i, y_i) = -\log\left(\frac{\exp(x_i[y_i])}{\sum_{j\in C}\exp(x_i[j])}\right)$$

$$C = \text{childrenOf}(v_{i-1}^{y_{i-1}})$$



Marginalization

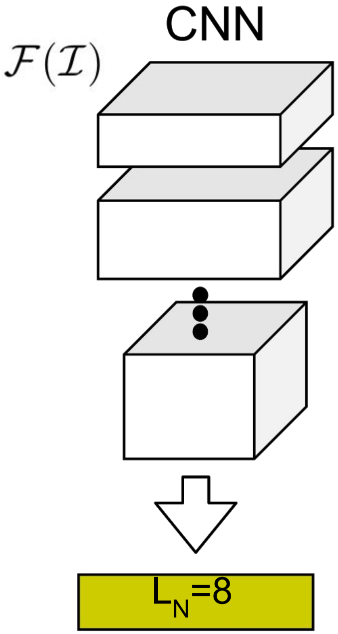
- Exploits: parent-child relationship
- Upper levels by summing over children.

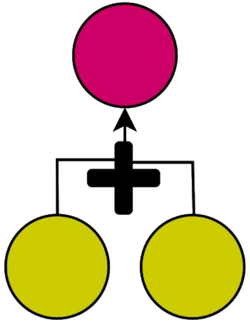
$$\mathscr{L}(x,y) = \sum_{i=1}^{L} \mathscr{L}_i(x_i, y_i) = -\sum_{i=1}^{L} \log\left(p_i[y_i]\right)$$

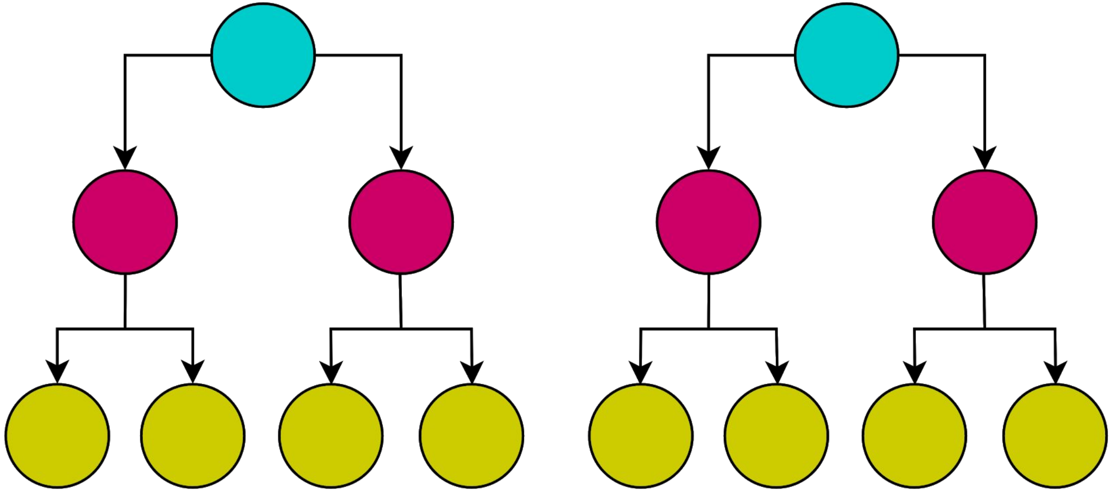
$$p_L[j] = P(v_L^j|\mathcal{I}) = \left(\frac{\exp(x_j)}{\sum_{k=1}^{N_L} \exp(x_k)}\right)$$

$$p_i[j] = P(v_i^j|\mathcal{I}) = \sum_{c \in \text{childrenOf}(v_i^j)} P(c|\mathcal{I}), \forall i \in 1, 2, ..., (L-1)$$

where, v_i^j is the *j-th* vertex (node) in the *i-th* level





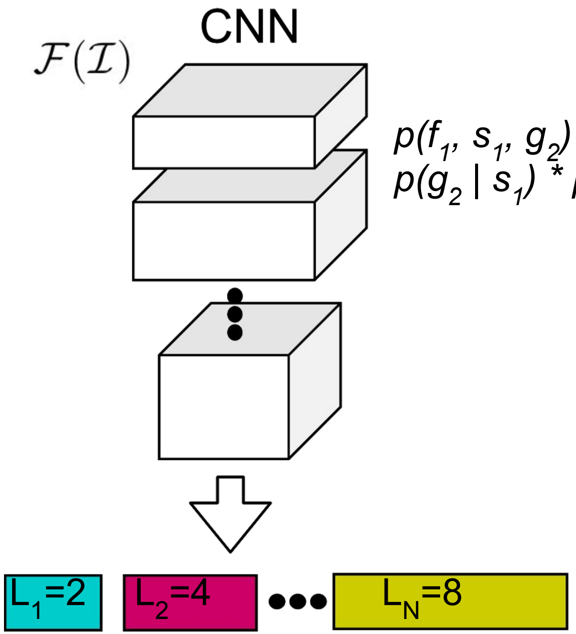


Hierarchical Softmax

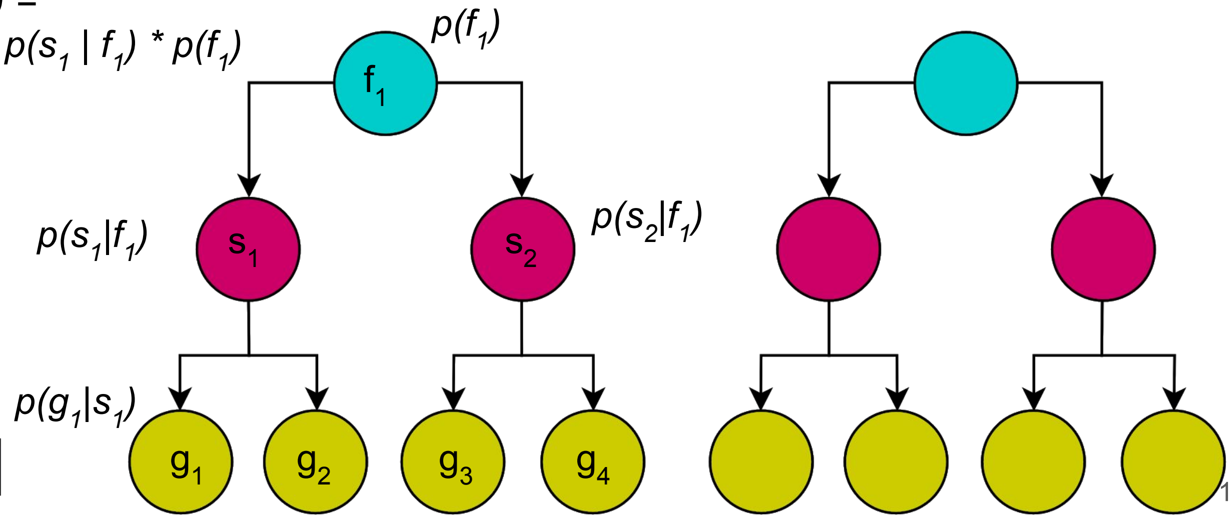
- Exploits: sub-tree relation + number of levels in label-hierarchy
- CNN predicts *p(child$_i$|parent)*

$$\mathcal{L}(x,y) = -\log\left(p(v_1^{y_1}, v_2^{y_2}, ..., v_{(L-1)}^{y_{L-1}}, v_L^{y_L})\right)$$

$$p(v_1^{j_1}, v_2^{j_2}, ..., v_{(L-1)}^{j_{(L-1)}}, v_L^{j_L}) = p(v_1^{j_1})p(v_2^{j_2}|v_1^{j_1})...p(v_L^{j_L}|v_{(L-1)}^{j_{(L-1)}})$$



$p(f_1, s_1, g_2) =$
$p(g_2 \mid s_1) * p(s_1 \mid f_1) * p(f_1)$



Experiments: Injecting Label-hierarchy into CNN Classifiers

Experiments

	micro-F1
Hierarchy-agnostic classifier	0.8147
Per-level classifier	0.9084
Masked Per-level classifier	0.9173
Marginalization	**0.9223**
Hierarchical softmax	0.9180

Model performance on *test* set for image classification on the ETHEC dataset.

m-F1	m-F1 L_1	m-F1 L_2	m-F1 L_3	m-F1 L_4
	Per-level micro-F1			
0.9223	0.9887	0.9758	0.9273	0.7972

Level-wise micro-F1 for the best performing baseline (Marginalization model).

Methods: Order-preserving Embeddings

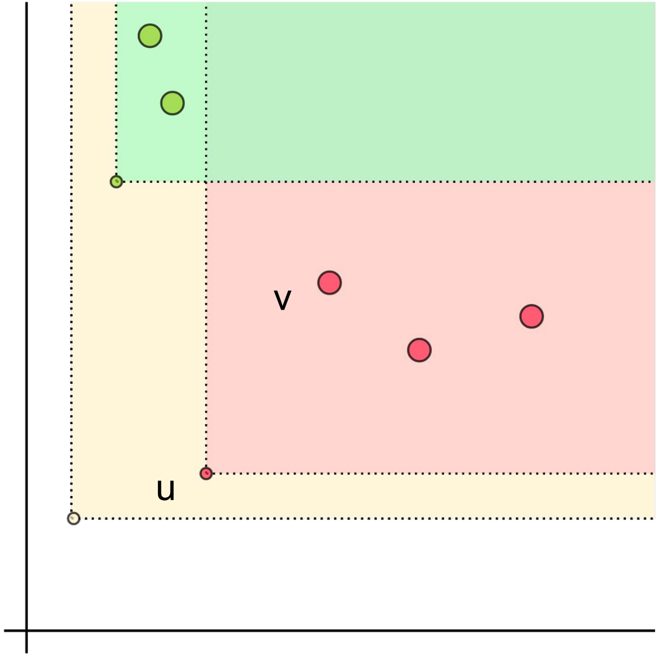
Label-hierarchy only

Label-hierarchy with Images

Learning Joint-Embeddings For Image Classification

Order-preserving Embeddings

- Order-Embeddings
- Euclidean Cones
- Hyperbolic Cones



Order Embeddings* and Entailment Cones**

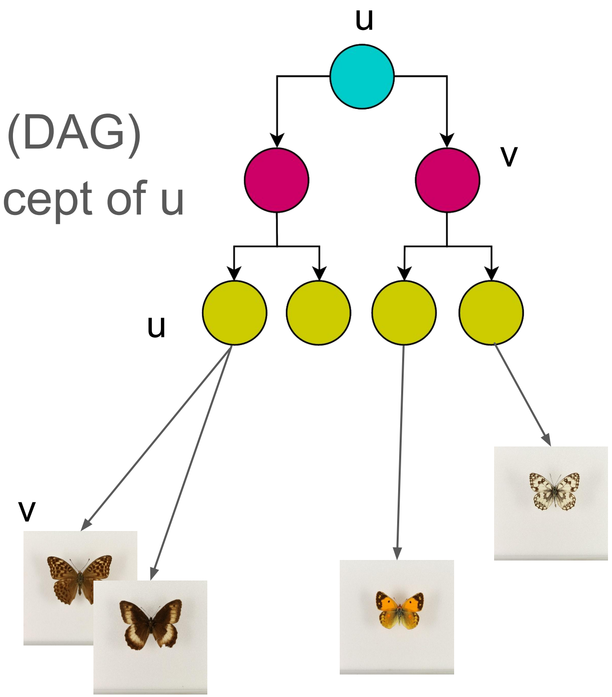
(1) For embedding label-hierarchy only:

- Treat the label-hierarchy as a directed acyclic graph (DAG)
- A directed edge (u, v) symbolizes that v is a sub-concept of u

(2) For embedding images and labels jointly:

- Connect the image to the label associated with it from the last level in the label-hierarchy

Use the joint-embeddings for image classification



Images form the <u>leaves</u> as upper nodes are more abstract

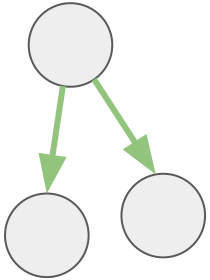
* *Order-Embeddings*; I Vendrov, R Kiros, S Fidler, R Urtasun

** *Hyperbolic Entailment Cones*; OE Ganea, G Bécigneul, T Hofmann

Experimental Setup

- Input: +ve and -ve edges from the DAG
- Output: if given pair of concepts *(u, v)* have a directed edge in the DAG; classify (u, v) as +ve or -ve

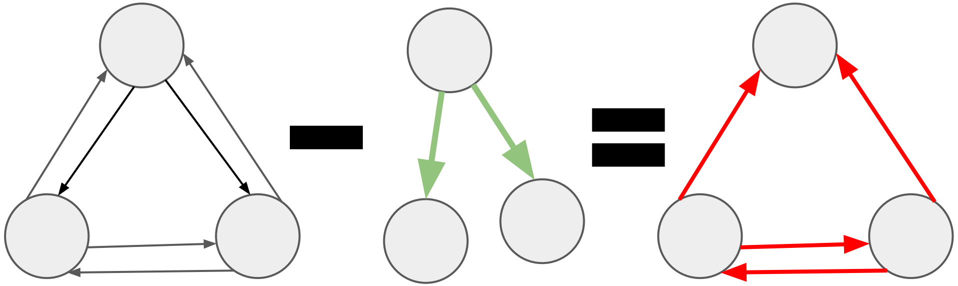
+ve edges



Loss:

$$\mathcal{L}(P,N) = \sum_{(u,v)\in P} E(u,v) + \sum_{(u',v')\in N} \max(0, \gamma - E(u',v'))$$

E is an energy function. *P* and *N* are +ve and -ve edges
-ve concepts should be separated by a margin



Fully-connected digraph w/o self-loops

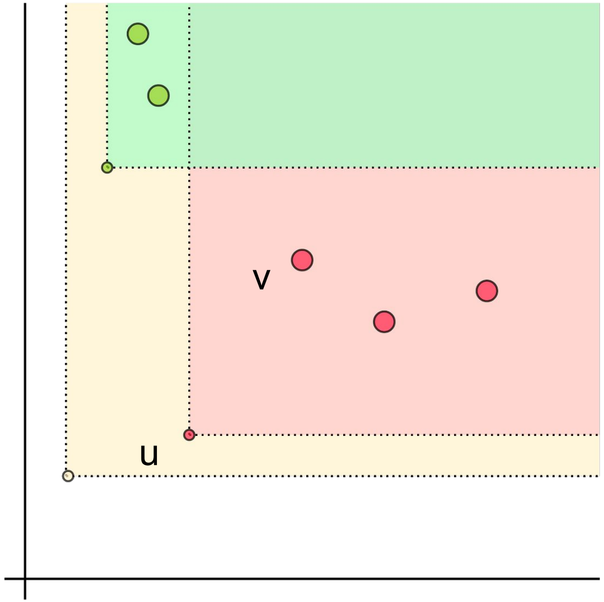
Delete +ve edges

-ve edges

Metrics:

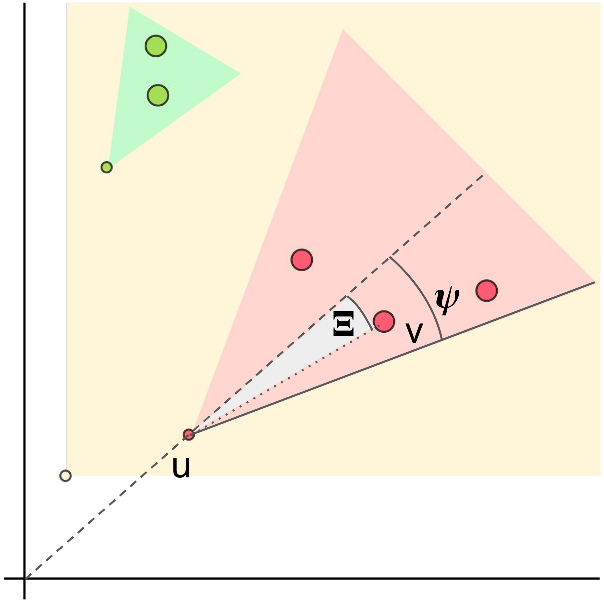
- True positive rate (TPR) and True negative rate (TNR)
- full-F1 score: F1 score on **all** +ve and -ve edges in the DAG => check reconstruction capability

Order Embeddings and Entailment Cones



For a given pair of concepts, (u, v), if u entails v then u falls within the quadrant that originates at u.

$$E(u, v) := \|\max(0, v - u)\|^2$$



For a given pair of concepts, (u, v), if u entails v then u falls within the cone that originates at u.

$$E(u, v) := \max(0, \Xi(u, v) - \psi(u))$$

Performance: Order-preserving Embeddings

Label-hierarchy only

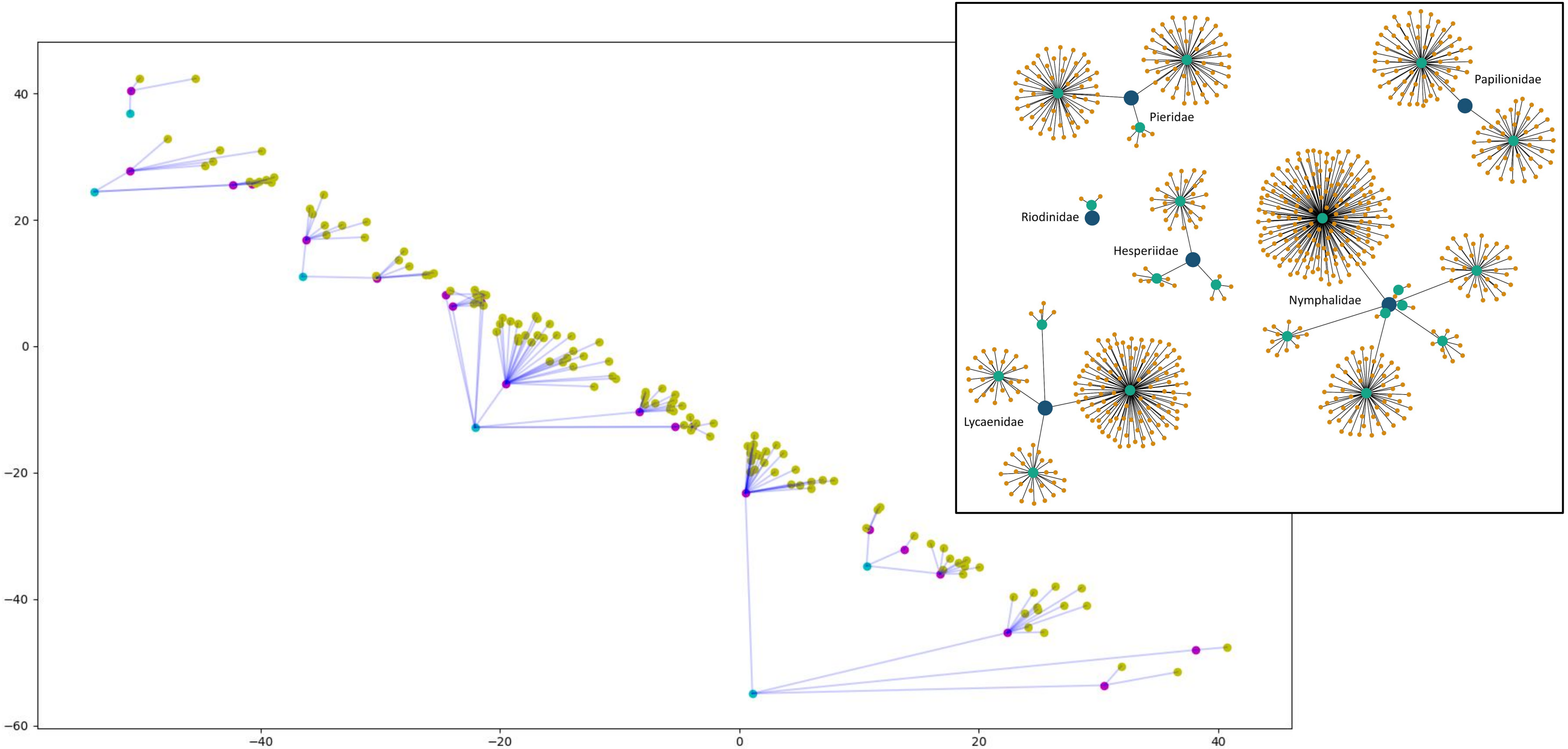
Label-hierarchy with Images

Embedding Labels | Order Embeddings

family

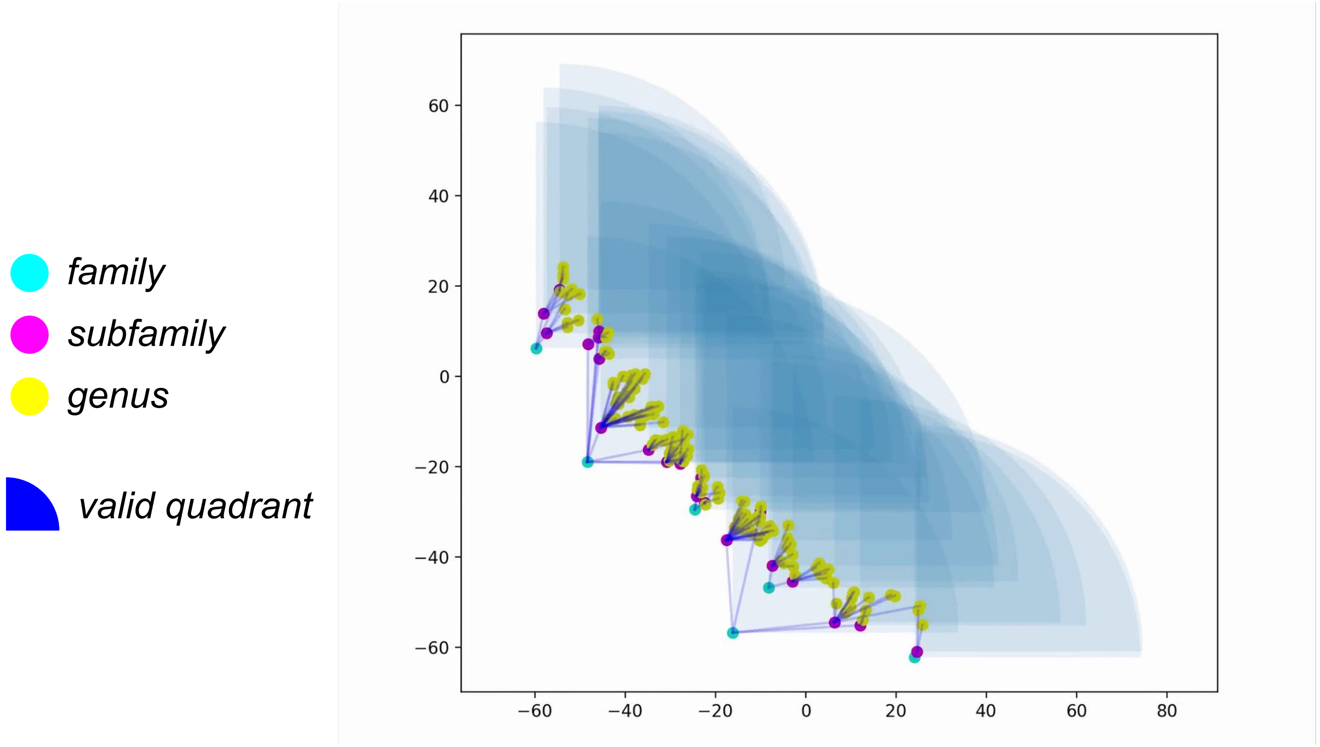
subfamily

genus



2-dimensional Order Embeddings for the ETHEC dataset in R^2 embedding space (for labels only).

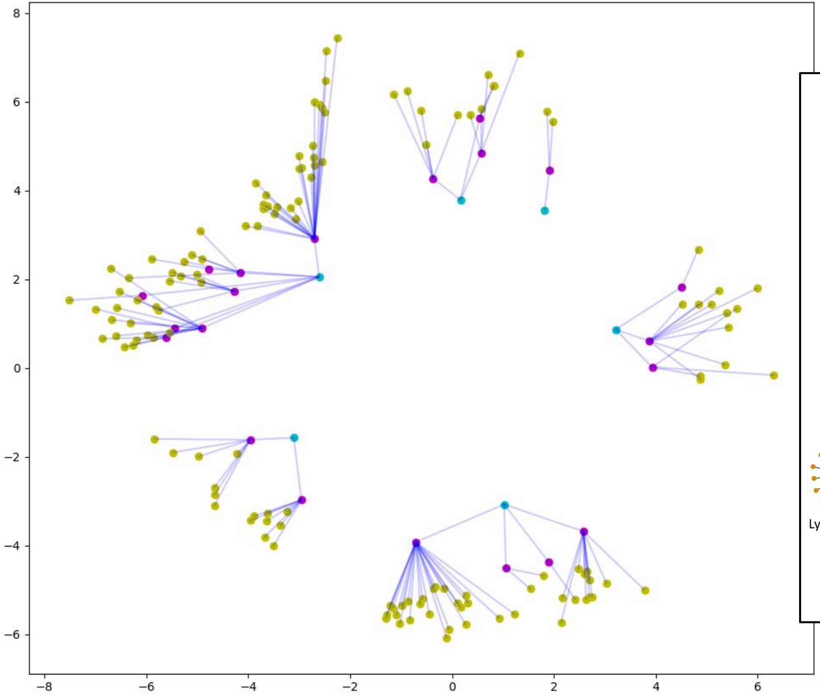
Embedding Labels | Order Embeddings

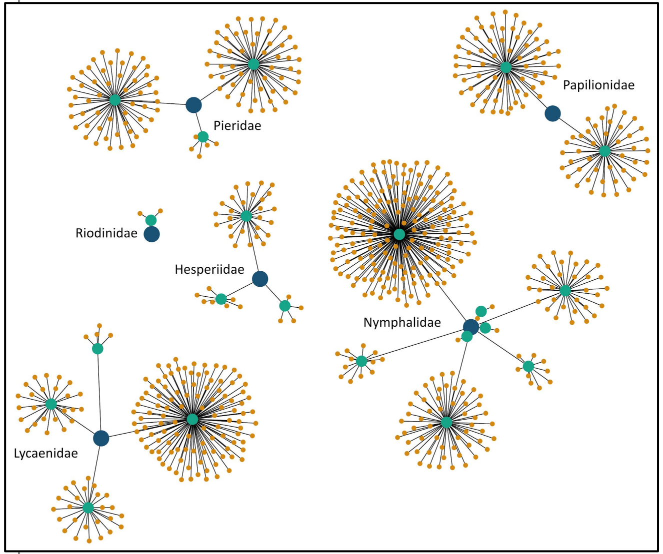


Evolution of 2-dimensional Order Embeddings for ETHEC dataset (for labels only) over time.
The metrics above are computed by classifying (distinguishing between) all positive and negative relations in the hierarchy.

Embedding Labels | Euclidean Cones

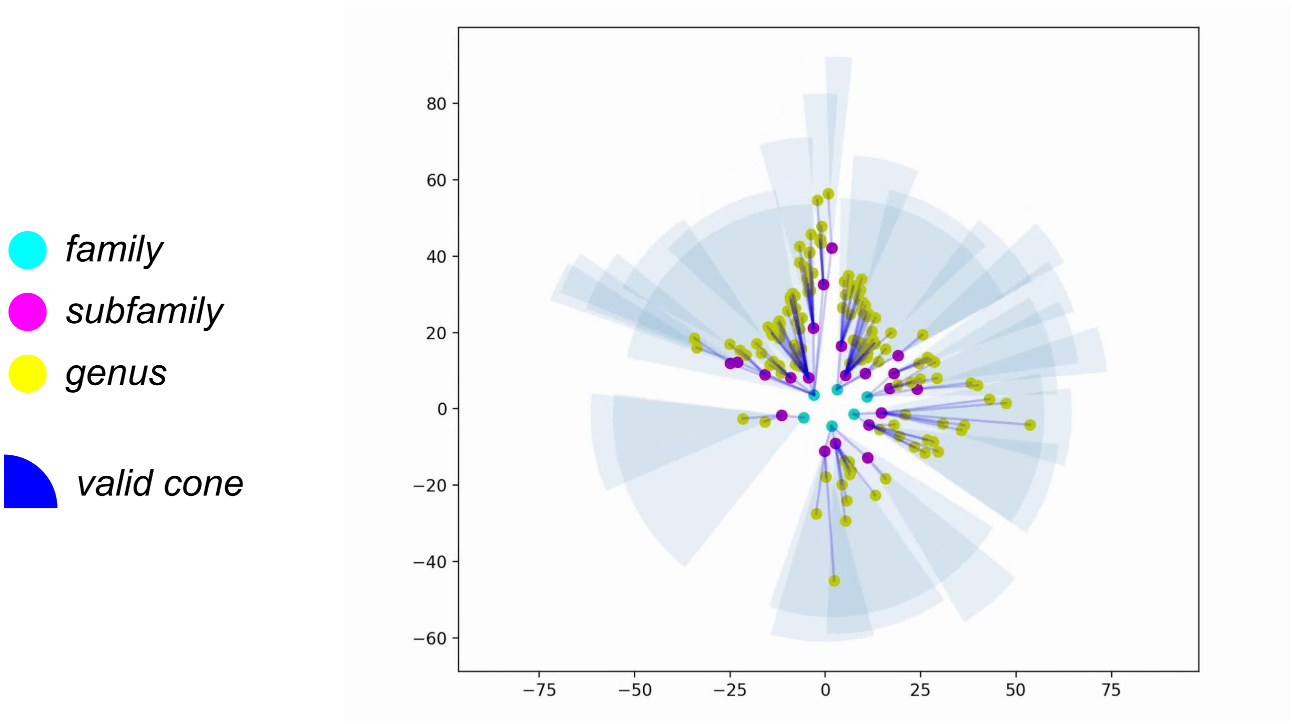
family
subfamily
genus





2-dimensional Euclidean cones for the ETHEC dataset in R^2 embedding space (for labels only).

Embedding Labels | Euclidean Cones



Evolution of 2-dimensional Euclidean Cones for the ETHEC dataset (for labels only) over time.
The metrics above are computed by classifying (distinguishing between) all positive and negative relations in the hierarchy.

Hyperbolic Cones

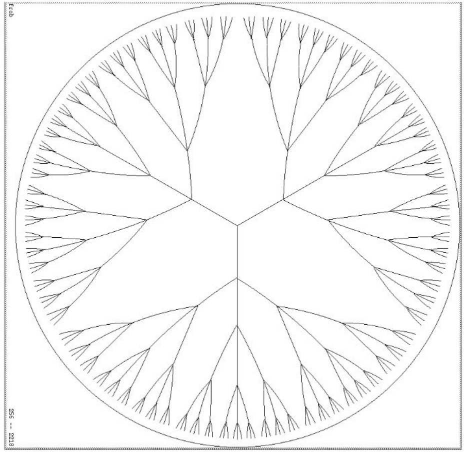
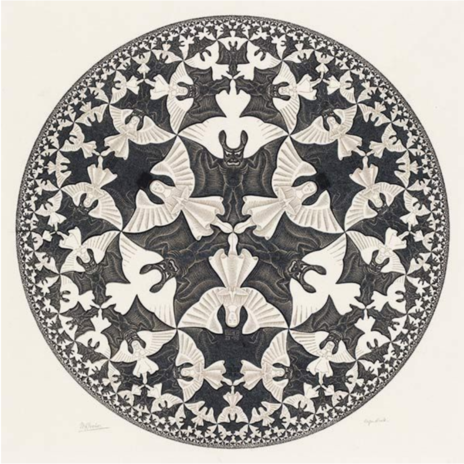


Image source: http://prior.sigchi.org



Volume of d-dimensional ball

Euclidean: $V_d^{\mathbb{E}}(r) \propto r^d$

Hyperbolic: $V_d^{\mathbb{H}}(r) \propto e^r$

Nodes in a tree with height *h* and branching factor *b*

$$\text{num\_nodes}_b(h) \propto b^h$$

- Move away from model parameters that assumes Euclidean geometry
- Embeddings live in hyperbolic space and exploit hyperbolic geometry
- Embed tree structure in Hyperbolic space with low-distortion*

Center image: M Escher

* Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane; R Sarkar

Optimization in Hyperbolic Space [+]

Gradient descent with Euclidean gradient in Euclidean space,

$$u \leftarrow u - \eta \, \nabla_u \mathcal{L}$$

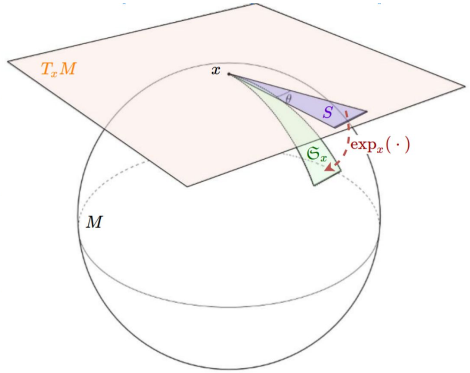
Riemannian Gradient for parameters living in non-Euclidean space,

$$\nabla_u^R \mathcal{L} = (1/\lambda_u)^2 \nabla_u \mathcal{L} \qquad\qquad \lambda_u = 2/(1 - ||u||^2)$$

Riemannian Gradient Descent using exponential map,

$$u \leftarrow \exp_u(\eta \, \nabla_u^R \mathcal{L})$$

$$\exp_x(v) \, : \, T_x\mathbb{D}^n \rightarrow \mathbb{D}^n$$



+ *Hyperbolic Entailment Cones*; OE Ganea, G Bécigneul, T Hofmann (image source)

Performance | Embedding labels only

	d=2	d=100	d=1000
	TPR/ TNR/ (full-F1)	TPR/ TNR/ (full-F1)	TPR/ TNR/ (full-F1)
OE	0.2309 / 0.9708 / (0.1372)	0.4686 / 0.9880 / (0.3894)	0.3788 / 0.9878 / (0.3489)
EC	0.3617 / 0.9975 / (0.3573)	0.4802 / 0.9985 / (0.4151)	0.5790 / 0.9973 / (0.4091)
HC	0.4443 / 0.9907 / (0.2296)	0.9336 / 0.9986 / (0.8060)	**0.9721** / **0.9986** / **(0.8257)**

d=number of dimensions of embedding space

OE: Order-embeddings, EC: Euclidean cones, HC: Hyperbolic cones

- True positive rate and true negative rate on all +ve and -ve edges from DAG
- DAG represents label-hierarchy in the ETHEC dataset
- Also report F1 score on classifying **all** edges
- 723 +ve edges; 521,289 -ve edges

Experiments: Order-preserving Embeddings

Label-hierarchy only

Label-hierarchy with Images

Jointly Embedding Images and Label-hierarchy

Image-embedding

Learnable matrix

Frozen CNN

$$f_i(i) = W * \mathrm{CNN}(i) \in \mathbb{R}^{\mathrm{d}}$$

Label-embedding

$$g_l(l) \in \mathbb{R}^{\mathrm{d}}$$

To classify a given image *i*,

$$\arg\min_l E(g_l(l), f_i(i)), \forall l \in \text{labels}$$

Return label with least-violating energy E

Label-embedding

Image-embedding

labels from a single level

Loss:

$$\mathcal{L}(P,N) = \sum_{(u,v)\in P} E(u,v) + \sum_{(u',v')\in N} \max(0, \gamma - E(u',v'))$$

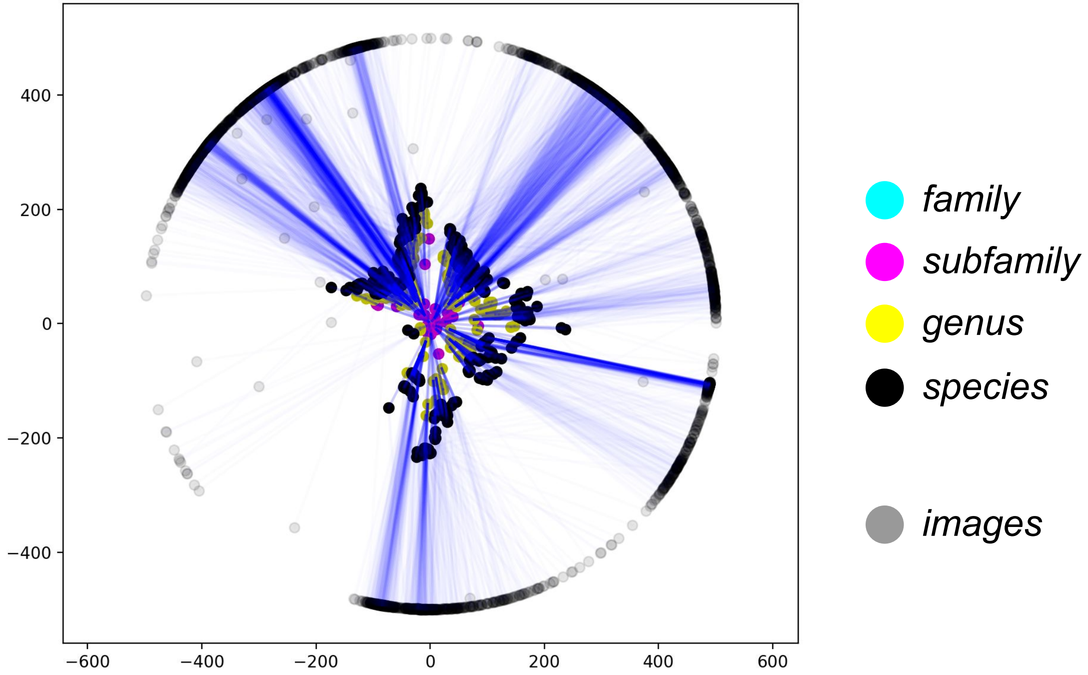
Perform same optimization as before,

(u, v) := $(g_l(l), f_i(i))$

Optimize using Adam to learn *W* and label embeddings, $f_l(l)$

Extremely challenging to optimize!
- Highly non-convex non-Euclidean landscape
- 2 different types of objects: images & labels
- Riemannian optimizer is accurate but weak!
- Hard to manage Adam & RSGD together
- Adam with approximation works best

Jointly Embedding Images and Label-hierarchy



Visualization of labels and images in joint 2D embedding space using Euclidean Cones.
The nodes on the periphery are images.

Jointly Embedding Images and Label-hierarchy

	classify test set images			graph reconstruction		
Model	**m-F1**	hit@3	hit@5	TPR	TNR	full-F1
Euclidean Cones						
d=10	0.7795	0.8893	0.9204	0.8045	0.9982	0.7040
d=100	0.8350	0.9018	0.9425	**0.9630**	**0.9986**	**0.8210**
d=1000	0.8013	0.8971	0.9278	0.8146	0.9981	0.7073
Hyperbolic Cones						
d=100	**0.8404**	**0.9200**	**0.9386**	0.6418	0.9978	0.5756
d=1000	0.8045	0.9023	0.9281	0.5233	0.9973	0.4832

Classification performance directly comparable with the CNN-based image classifiers!

Performance Summary

Models that use label-hierarchy information outperform the hierarchy-agnostic model.

		Per-level micro-F1			
Model	m-F1	m-F1 L_1	m-F1 L_2	m-F1 L_3	m-F1 L_4
CNN-based methods					
Hierarchy-agnostic (baseline)	0.8147	0.9417	0.9446	0.8311	0.4578
Per-level classifier	0.9084	0.9766	0.9661	0.9204	0.7704
Marginalization classifier	**0.9223**	**0.9887**	**0.9758**	**0.9273**	**0.7972**
Masked Per-level classifier	0.9173	0.9828	0.9701	0.9233	0.7930
Hierarchical-softmax	0.9180	0.9879	0.9731	0.9253	0.7855
Order-preserving (joint) embedding models					
Euclidean cones d=100	0.8350	0.9728	0.9370	0.8336	0.5967
Hyperbolic cones d=100*	0.7627	0.9695	0.9205	0.7523	0.4246
Hyperbolic cones d=100	**0.8404**	**0.9800**	**0.9439**	**0.8477**	**0.5977**

Labels initialized w/ pre-trained *label-only* embeddings

** Randomly initialized*

Contributions

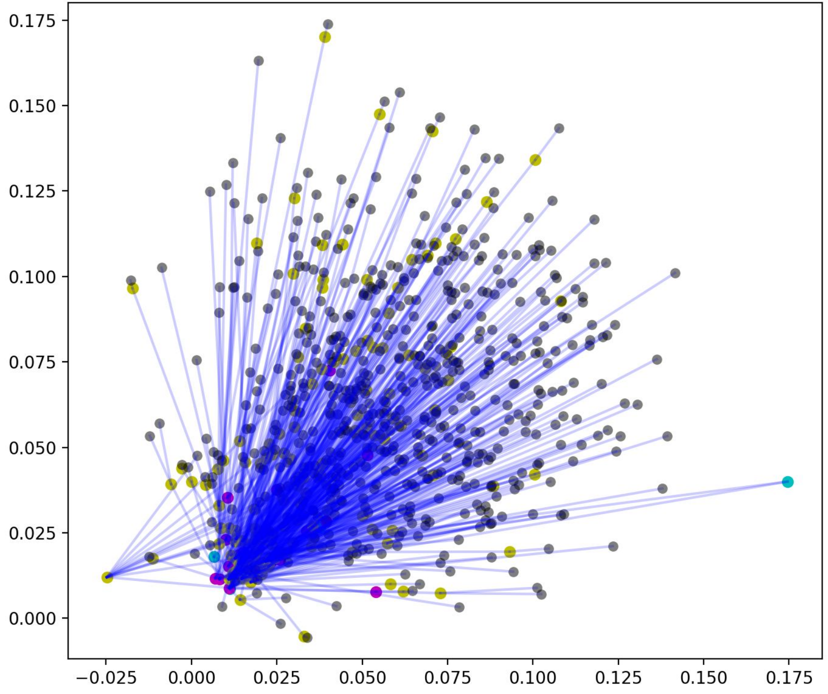
- Compared methods that exploit label-hierarchy knowledge
- Provide a reasonable model that can be used by Entomological collections
- Order-preserving embeddings show promise for computer vision

Future Directions

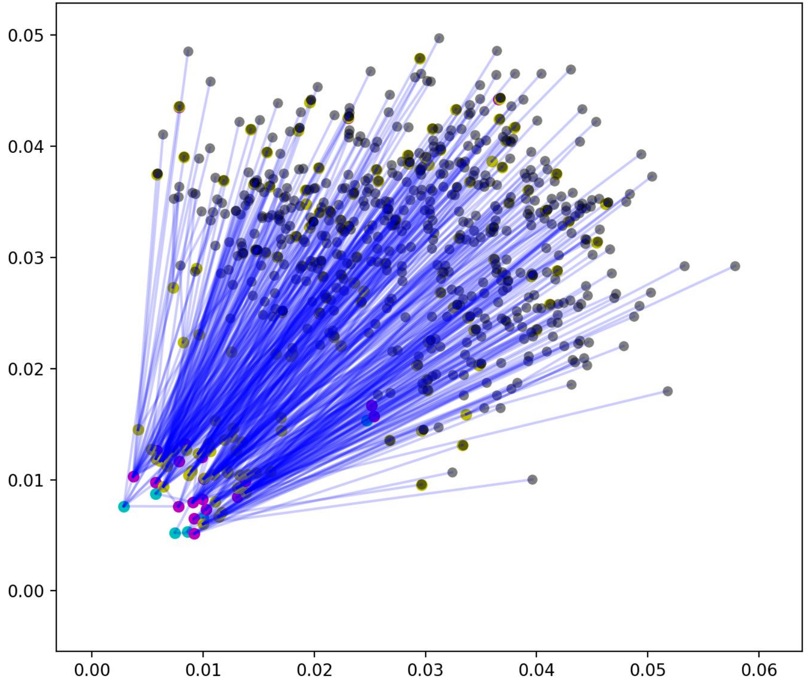
- Validate performance with other datasets with hierarchical labels
- Submit work to a conference

- Applications: Visual-Question Answering, Scene-graph generation = joint modeling of semantics and visual cues
- Label accuracy vs. label specificity: predict more generic if unsure about a more specific label (*eg: mammal* instead of *dog*)
- Model complexity to map images to embedding space

Thank you for your attention!



100-dimensional Hyperbolic cones projected in 2D.



1000-dimensional Hyperbolic cones projected in 2D.

family *subfamily* *genus* *species*

Additional material

ETH Entomological Collection (ETHEC)

- 2,000,000+ specimens; one of the largest insect collections in Europe
- New specimens need to be digitized and organized taxonomically
- Classification requires specialists and is expensive

ETH Entomological Collection (ETHEC) Dataset

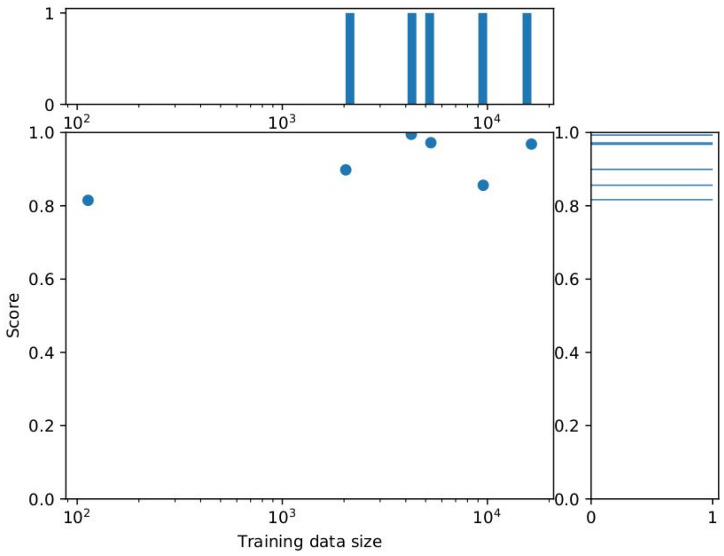
- Dataset with images and their corresponding hierarchical labels
- 47,978 butterfly images with a 4-level label-hierarchy
- 6 *family* -> 21 *sub-family* -> 135 *genus* -> 561 *species*
- Unbalanced tree & non-uniform image distribution among labels
- Each image has an associated label from each level in the hierarchy

Dataset has been made publicly available here:
https://www.research-collection.ethz.ch/handle/20.500.11850/365379

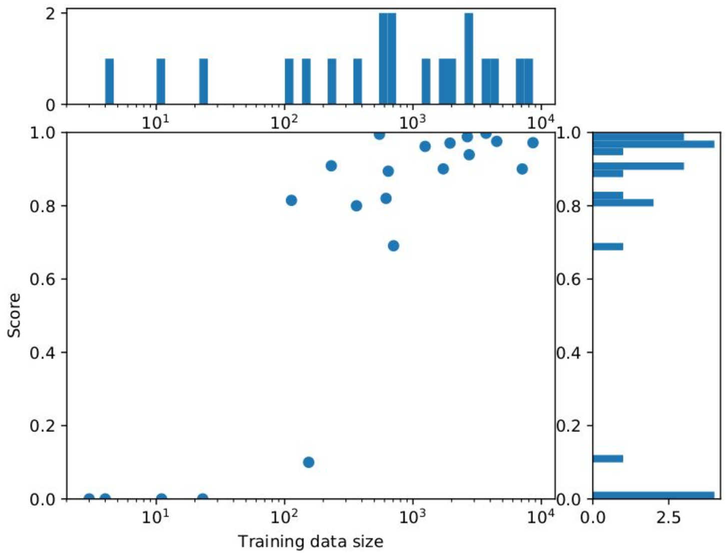
Hierarchy-agnostic model

- Per-class decision boundary vs. One-fits-all decision boundary
- Loss-reweighting and data resampling

Hierarchy-agnostic model | *family, subfamily*



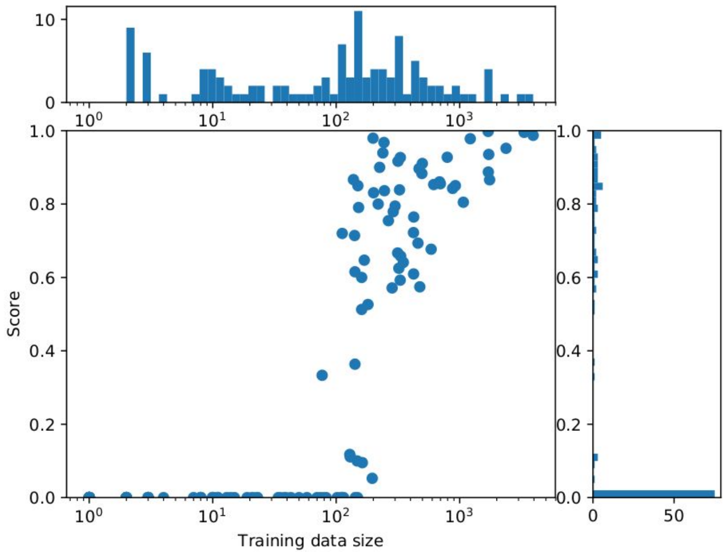
(a) *family*



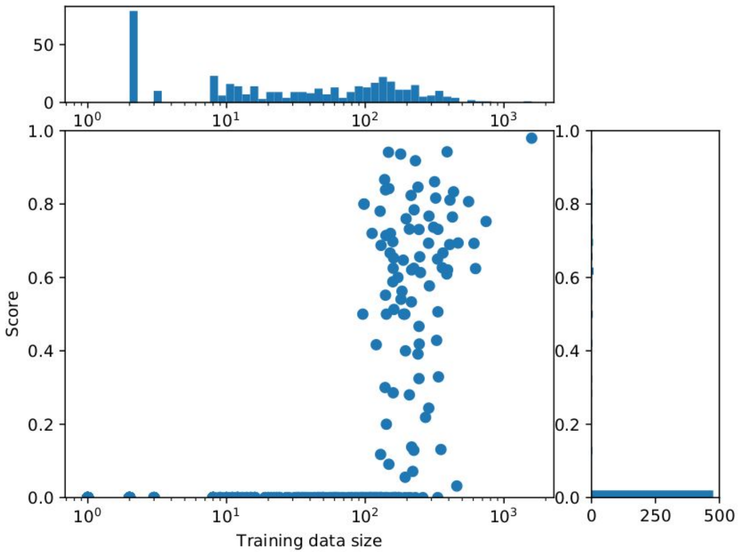
(b) *subfamily*

Each point represents a label from the particular level in the hierarchy. In addition, the distribution of the F1-score and training data size across labels. (x-axis: Training data size; y-axis: F1-score)

Hierarchy-agnostic model | *family, subfamily*



(c) *genus*



(d) *genus + specific epithet*

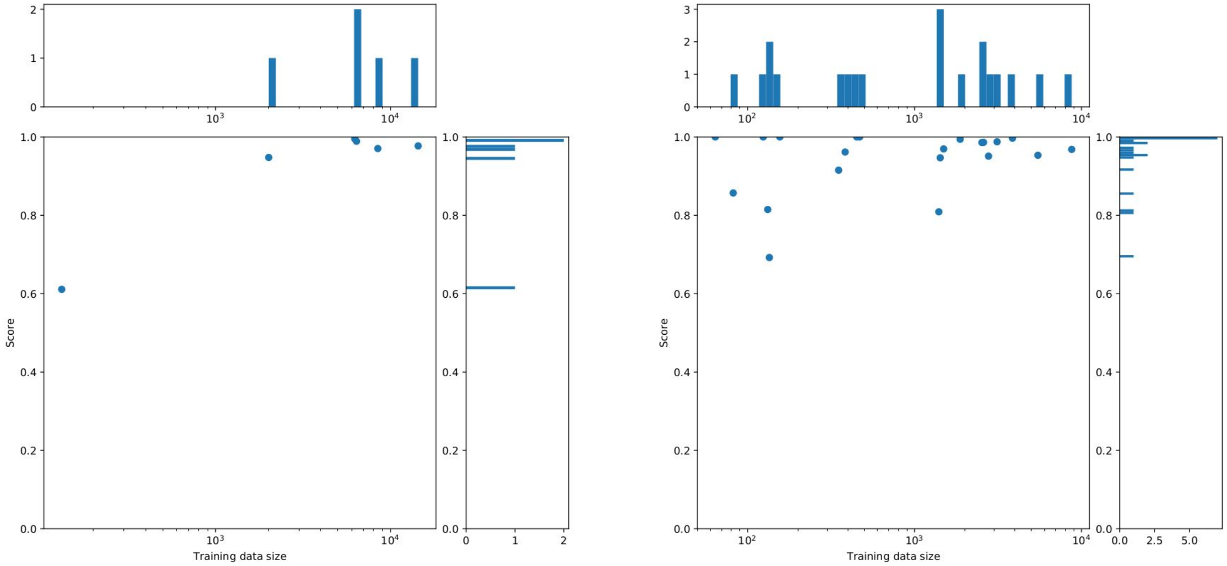
Each point represents a label from the particular level in the hierarchy. In addition, the distribution of the F1-score and training data size across labels. (x-axis: Training data size; y-axis: F1-score)

Hierarchy-agnostic model

cw	rs	m-P	m-R	m-F1	M-P	M-R	M-F1	(min, max), $\mu \pm \sigma$
ResNet-50 - Per-class decision boundary								
✗	✗	0.0355	0.7232	0.0677	0.3066	0.4053	0.2195	(3, 351), 81.42 $\pm$ 69.51
✗	✓	0.7159	0.7543	0.7346	**0.4402**	0.4362	**0.3718**	(0, 13), 4.21 $\pm$ 2.07
✓	✗	0.0077	**0.8702**	0.0153	0.0120	**0.8397**	0.0183	(84, 718), 451.14 $\pm$ 136.69
✓	✓	0.0081	0.7519	0.0161	0.0105	0.5909	0.0165	(33, 714), 369.96 $\pm$ 120.55
ResNet-50 - One-fits-all decision boundary								
✗	✗	0.9324	0.7235	**0.8147**	0.1913	0.1462	0.1568	(0, 7), 3.10 $\pm$ 1.16
✗	✓	**0.9500**	0.6564	0.7763	0.1078	0.0947	0.0959	(0, 5), 2.76 $\pm$ 0.60
✓	✗	0.2488	0.2960	0.2704	0.0021	0.0067	0.0030	(4, 9), 4.76 $\pm$ 0.76
✓	✓	0.1966	0.3800	0.2591	0.0027	0.0110	0.0037	(4, 10), 7.73 $\pm$ 0.61

Level	N_i	m-P	m-R	m-F1	M-P	M-R	M-F1
ResNet-50 (OFADB) with resampler (cw: ✗, rs: ✗)							
family	6	0.9861	0.9012	0.9417	0.9718	0.8801	0.9173
subfamily	21	0.9860	0.9065	0.9446	0.7941	0.6548	0.6968
genus	135	0.9290	0.7518	0.8311	0.3918	0.2961	0.3212
genus + specific epithet	561	0.7249	0.3345	0.4578	0.1121	0.0832	0.0888

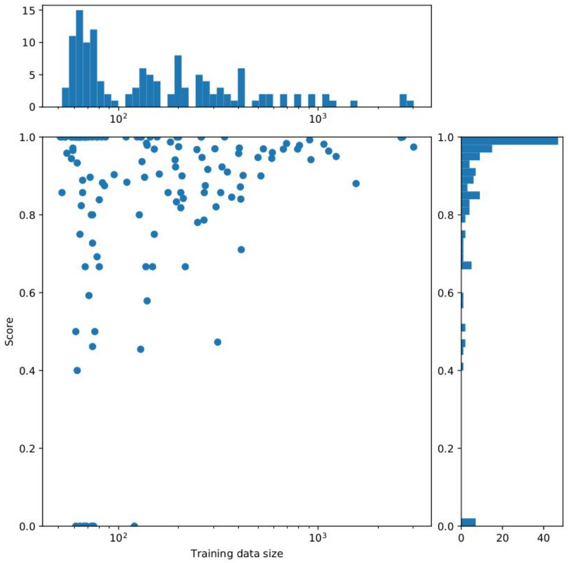
Per-level classifier | *family, subfamily*



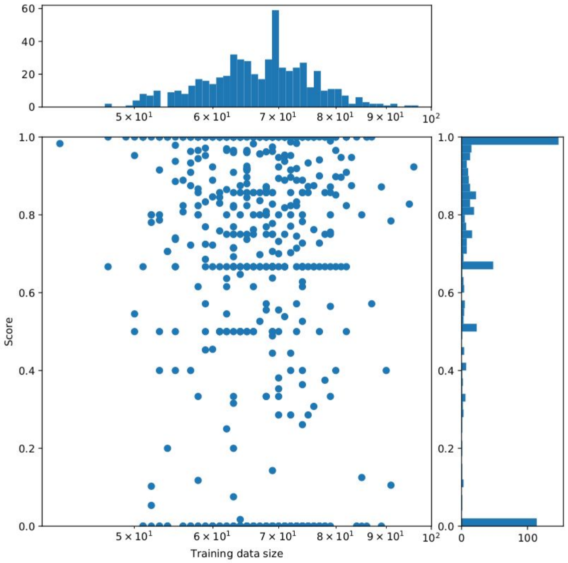
(a) *family* (b) *subfamily*

Each point represents a label from the particular level in the hierarchy. In addition, the distribution of the F1-score and training data size across labels. (x-axis: Training data size; y-axis: F1-score)

Per-level classifier | *genus, species*



(c) *genus*



(d) *genus + specific epithet*

Each point represents a label from the particular level in the hierarchy. In addition, the distribution of the F1-score and training data size across labels. (x-axis: Training data size; y-axis: F1-score)

Per-level classifier

cw	rs	m-P	m-R	m-F1	M-P	M-R	M-F1
ResNet-50							
✓	✗	0.8483	0.8483	0.8483	0.6648	0.6789	0.6411
✗	✗	0.8930	0.8930	0.8930	0.6854	0.7094	0.6677
✗	✓	**0.9084**	**0.9084**	**0.9084**	**0.7134**	**0.7223**	**0.6888**
✓	✓	0.8760	0.8760	0.8760	0.6782	0.6874	0.6537
✗	`sqrt`	0.9067	0.9067	0.9067	0.6941	0.7073	0.6700

Level	N_i	m-P	m-R	m-F1	M-P	M-R	M-F1
ResNet-50 with resampler (cw: ✗, rs: ✓)							
family	6	0.9766	0.9766	0.9766	0.9005	0.9328	0.9152
subfamily	21	0.9661	0.9661	0.9661	0.9433	0.9542	0.9424
genus	135	0.9204	0.9204	0.9204	0.8845	0.8482	0.8497
genus + specific epithet	561	0.7704	0.7704	0.7704	0.6616	0.6811	0.6382

Marginalization

model	m-P	m-R	m-F1	M-P	M-R	M-F1
Models trained using grayscale images						
ResNet-50	0.8586	0.8586	0.8586	0.6071	0.6070	0.5765
Models trained using normal color images						
ResNet-50	**0.9223**	**0.9223**	**0.9223**	**0.7095**	**0.7231**	**0.6927**
ResNet-101	0.9110	0.9110	0.9110	0.7327	0.7262	0.7023
ResNet-152	0.9162	0.9162	0.9162	0.7181	0.7271	0.6954

L_1	L_2	L_3	L_4	m-F1	m-F1 L_1	m-F1 L_2	m-F1 L_3	m-F1 L_4
term L_i in loss					Per-level micro-F1			
			✓	0.9137	0.9814	0.9638	0.9134	0.7962
		✓	✓	0.9070	0.9774	0.9626	0.9077	0.7804
	✓	✓	✓	0.9207	0.9891	0.9733	0.9255	0.7948
✓	✓	✓	✓	0.9223	0.9887	0.9758	0.9273	0.7972

Masked Per-level classifier

model	m-P	m-R	m-F1	M-P	M-R	M-F1
Models trained using grayscale images						
ResNet-50	0.8443	0.8443	0.8443	0.6002	0.5931	0.5619
Models trained using normal color images						
ResNet-50	**0.9173**	**0.9173**	**0.9173**	0.7107	0.7227	0.6915
ResNet-101	0.9169	0.9169	0.9169	0.7119	0.7260	0.6921
ResNet-152	0.9152	0.9152	0.9152	**0.7167**	**0.7281**	**0.6958**

Level	N_i	m-P	m-R	m-F1	M-P	M-R	M-F1
ResNet-50 Performance Breakdown							
family	6	0.9828	0.9828	0.9828	0.9735	0.9361	0.9495
subfamily	21	0.9701	0.9701	0.9701	0.9684	0.9252	0.9356
genus	135	0.9233	0.9233	0.9233	0.8916	0.8432	0.8525
genus + specific epithet	561	0.7930	0.7930	0.7930	0.6548	0.6838	0.6409

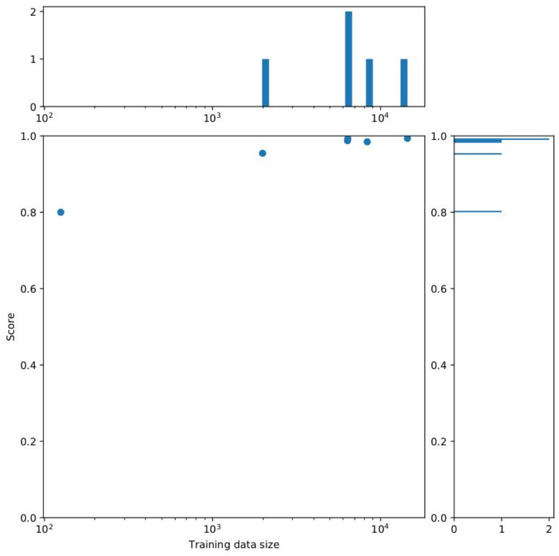
L_1	L_2	L_3	L_4	m-F1	m-F1 L_1	m-F1 L_2	m-F1 L_3	m-F1 L_4
term L_i in loss					Per-level micro-F1			
			✓	0.0633	0.2325	0.0162	0.0022	0.0022
		✓	✓	0.1043	0.3058	0.0410	0.0386	0.0319
	✓	✓	✓	0.0848	0.0970	0.0919	0.0879	0.0622
✓	✓	✓	✓	0.9098	0.9808	0.9616	0.9116	0.7853

Hierarchical Softmax

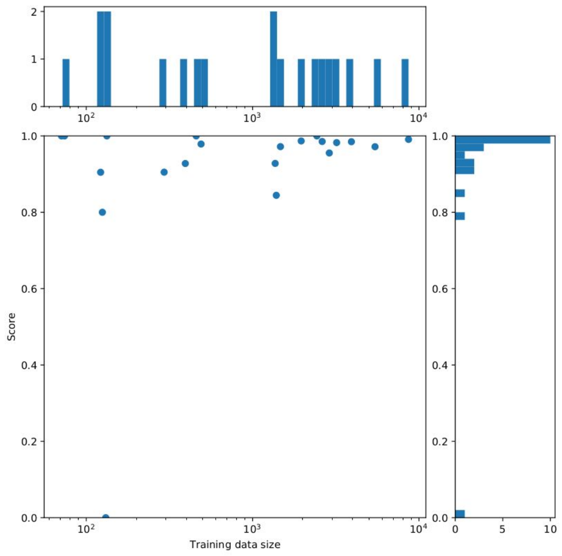
model	m-P	m-R	m-F1	M-P	M-R	M-F1
ResNet-50	0.9055	0.9055	0.9055	0.6899	0.7049	0.6723
ResNet-101	0.9122	0.9122	0.9122	0.7049	0.7072	0.6780
ResNet-152	**0.9180**	**0.9180**	**0.9180**	**0.7119**	**0.7174**	**0.6869**

Level	N_i	m-P	m-R	m-F1	M-P	M-R	M-F1
ResNet-152 with Hierarchical Softmax — Performance Breakdown							
family	6	0.9879	0.9879	0.9879	0.9605	0.9452	0.9522
subfamily	21	0.9731	0.9731	0.9731	0.9605	0.9452	0.9522
genus	135	0.9253	0.9253	0.9253	0.8972	0.8504	0.8574
genus + specific epithet	561	0.7855	0.7855	0.7855	0.6572	0.6756	0.6347

Hierarchical Softmax | *family, subfamily*



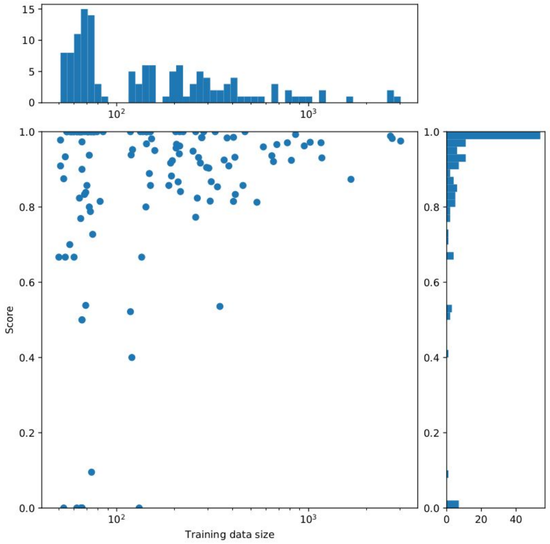
(a) *family*



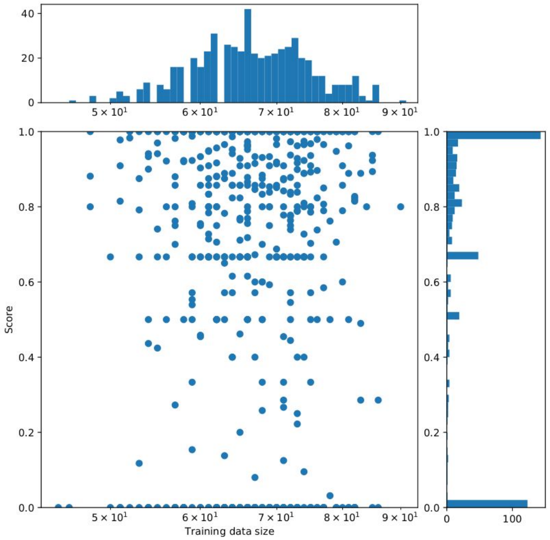
(b) *subfamily*

Each point represents a label from the particular level in the hierarchy. In addition, the distribution of the F1-score and training data size across labels. (x-axis: Training data size; y-axis: F1-score)

Hierarchical Softmax | *genus, species*



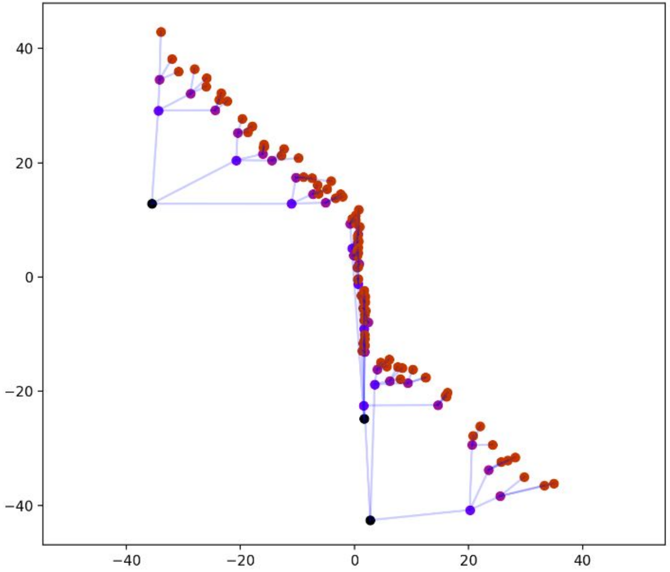
(c) *genus*



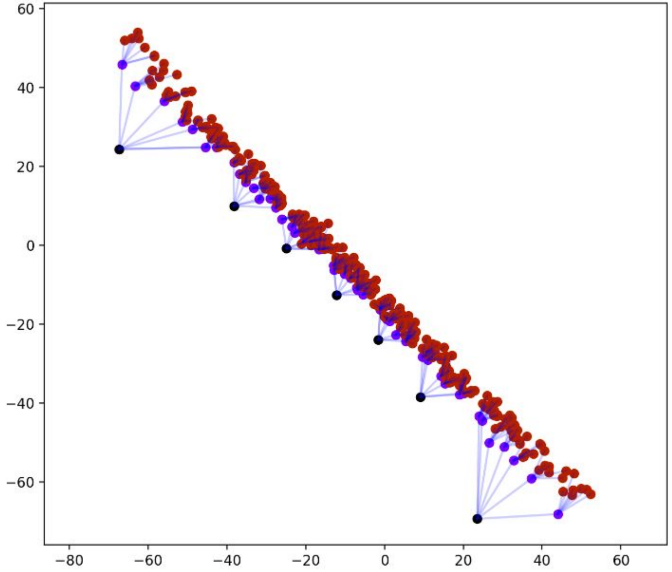
(d) *genus + specific epithet*

Each point represents a label from the particular level in the hierarchy. In addition, the distribution of the F1-score and training data size across labels. (x-axis: Training data size; y-axis: F1-score)

Embedding toy-graphs

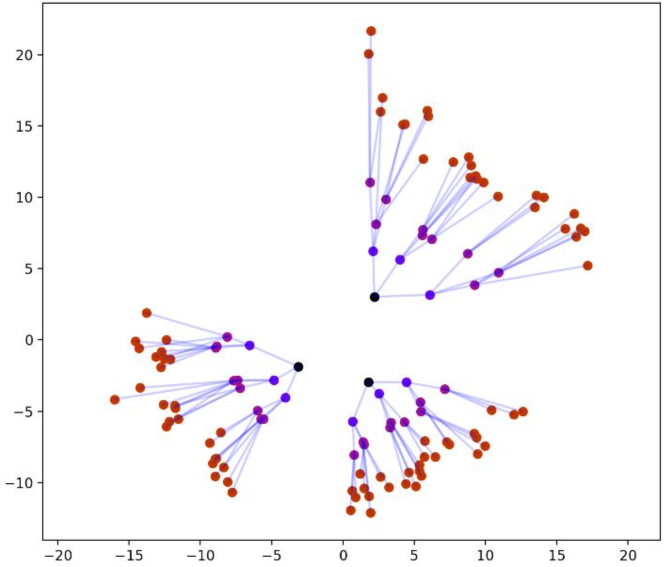


(a) *Order-embeddings L=4, b=3*

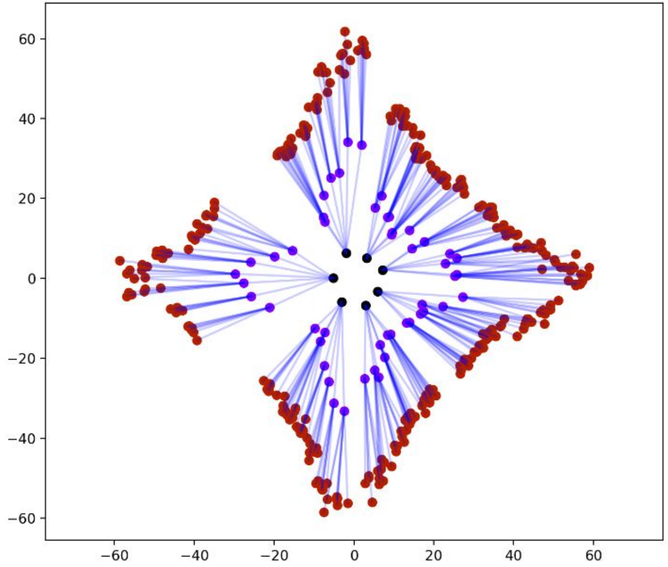


(b) Order-embeddings L=3, b=7

Embedding toy-graphs

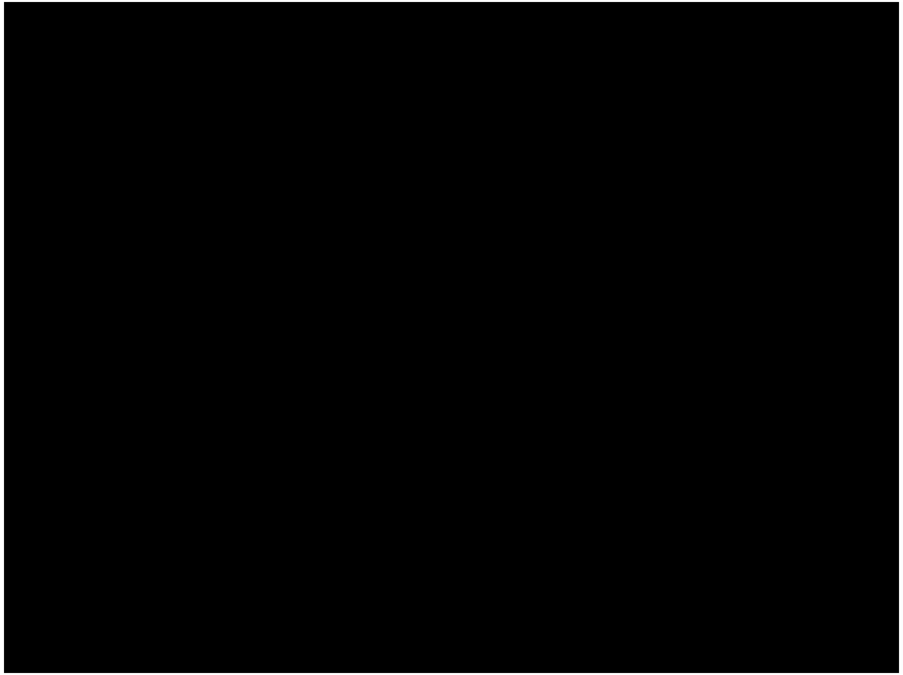


(c) Euclidean cones L=4, b=3

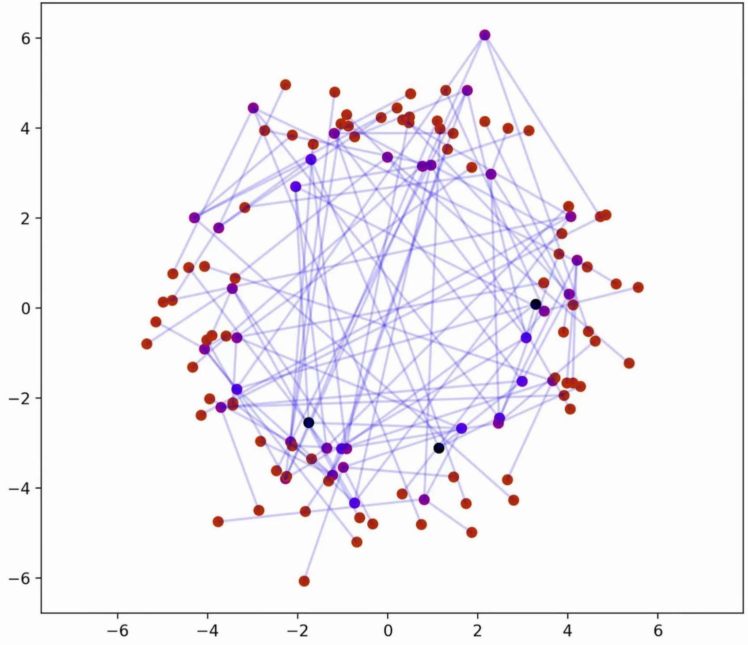


(d) Euclidean cones L=3, b=7

Synthetic Trees (L=4,b=3)

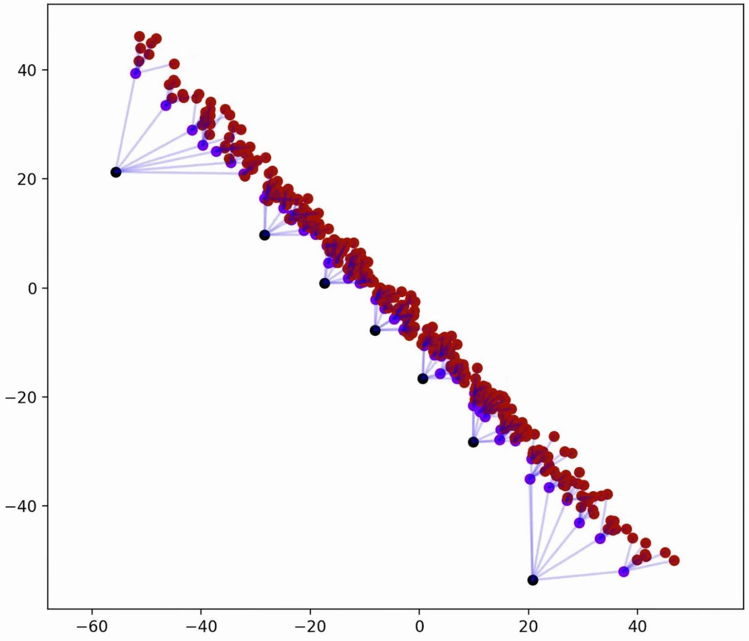


2D Order-Embeddings

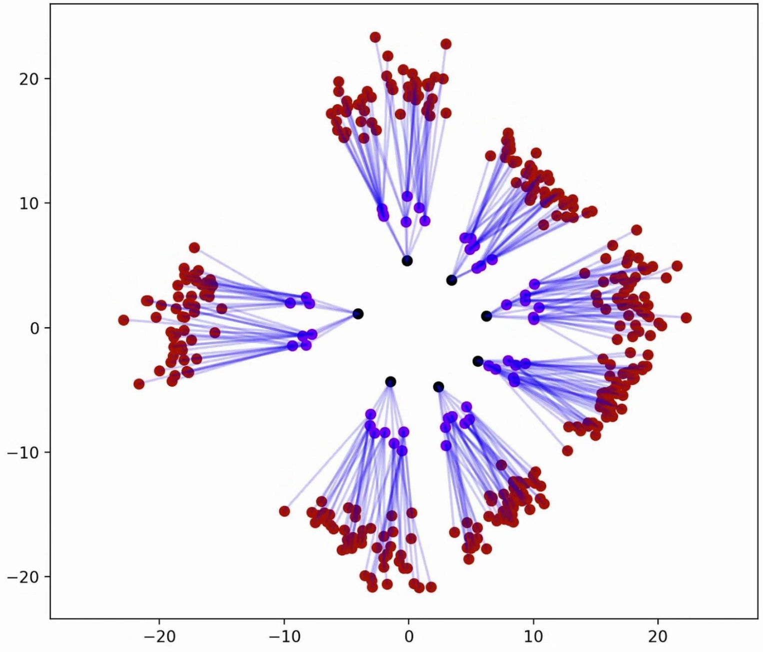


2D Euclidean Cones

Synthetic Trees (L=3,b=7)

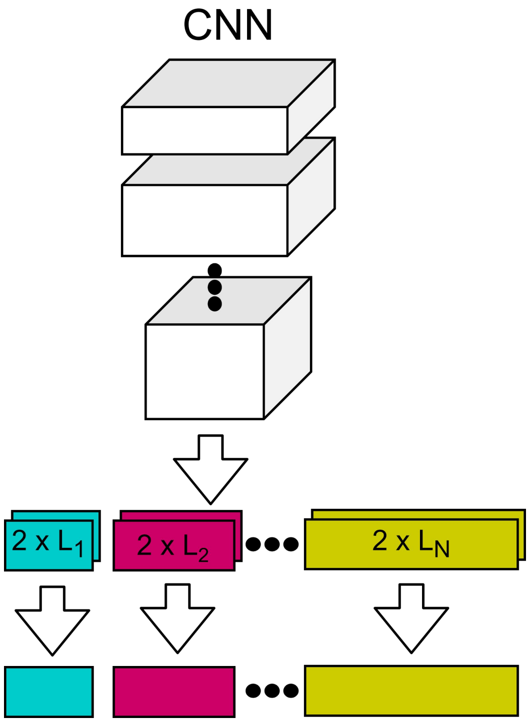


2D Order-Embeddings



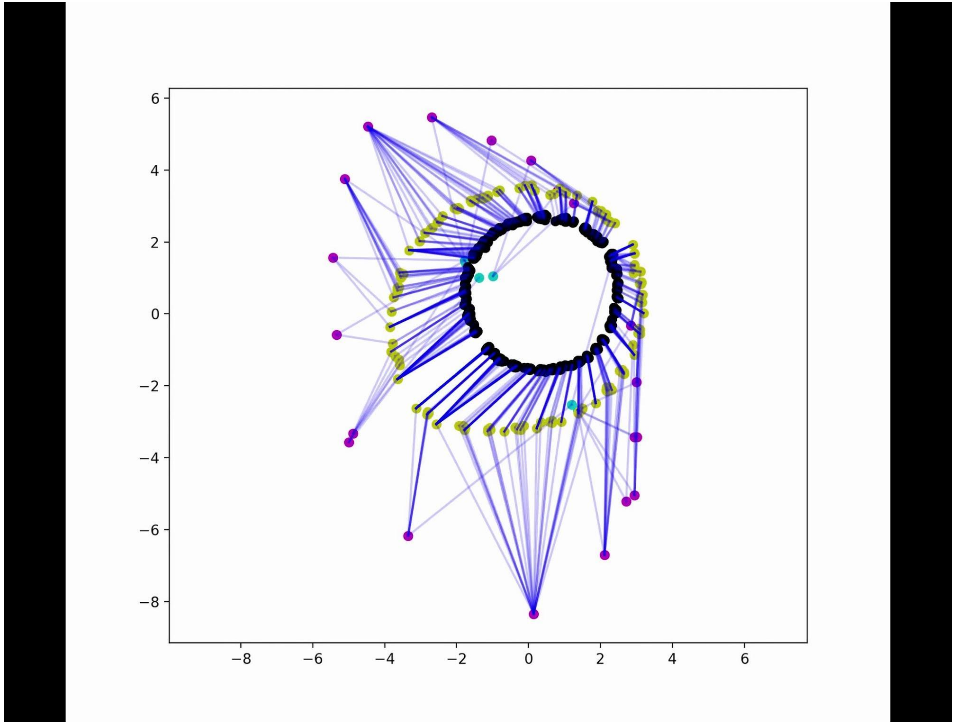
2D Euclidean Cones

Cosine Embeddings



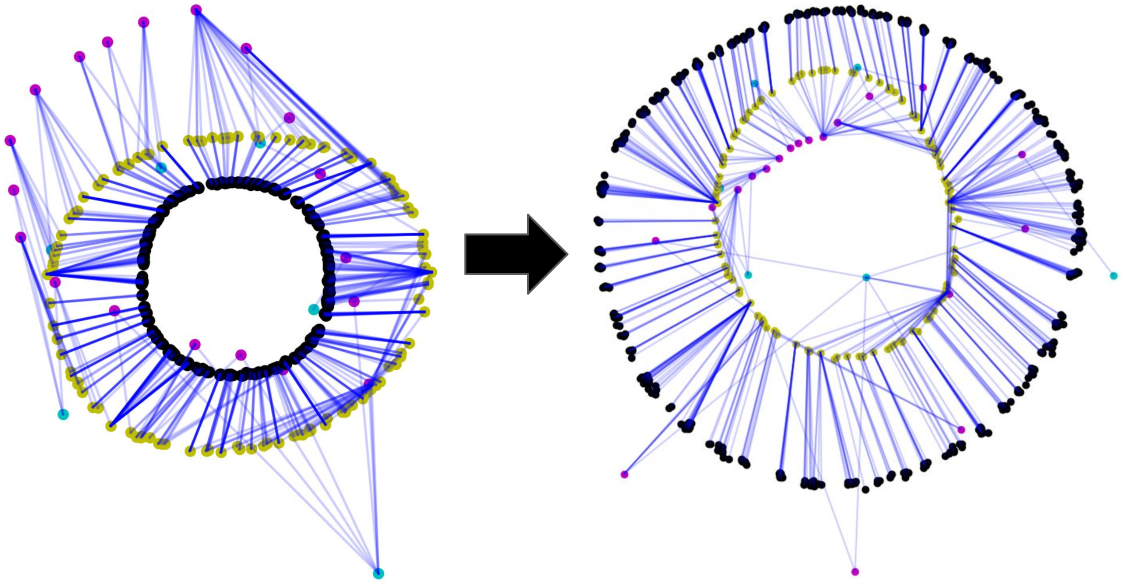
- Use multi-level classifier CNN from the baseline
- Add a set of linear layers whose weights live in 2 dimensions
- One such layer for every level in the hierarchy
- These weights represent the latent space learned while being trained for image classification

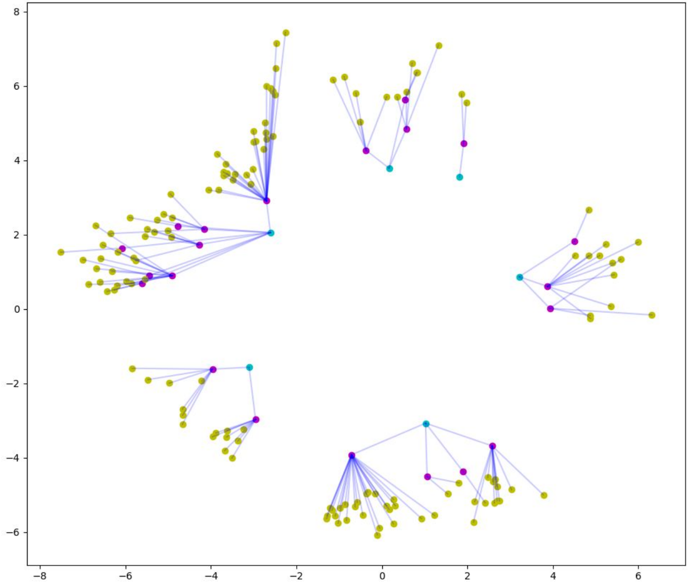
Embedding Labels | Cosine Embeddings



Evolution of 2-dimensional Cosine Embeddings over time.

Inverted Cosine embeddings





$$x_{\text{inverted}} = \frac{r * x * ||x_{\max}||}{||x||}$$

Inverted Cosine Embeddings resemble the Euclidean cones.

Performance | Embedding labels only

Model	d=2	d=3	d=5	d=10	d=100
Order-embeddings	0.8271	0.9302	0.9457	0.9920	0.9920
Euclidean Cones	0.8550	0.9979	0.9593	0.9919	0.9752

- Micro-F1 score on *test* set consisting of +ve and -ve edges from DAG
- DAG represents label-hierarchy in the ETHEC dataset

Training details | Joint embeddings

- alpha: EC=1.0, HC=0.1
- EC: 200 epochs, lr_img=10^-3, lr_labels=10^-2 with Adam
- HC: 100 epochs, lr_img=10^-3, lr_labels=10^-4 with Adam
- 10 negative (=5*(u, v’) + 5*(u’, v)) per positive with pick-per-level strategy
- Initialize the labels with the labels-only training
- For HC use Adam over RSGD -> converging faster, better performance