TASK2VEC: Task Embedding for Model Recommendation

Subhransu Maji
College of Information and Computer Sciences
University of Massachusetts, Amherst
http://people.cs.umass.edu/smaji

https://arxiv.org/abs/1902.03545

February 19, 2019 @ ICERM, Brown University

Task Embedding for Model Recommendation

Allesandro, Michael, Rahul, Avinash, Subhransu, Charless, Stefano, Pietro

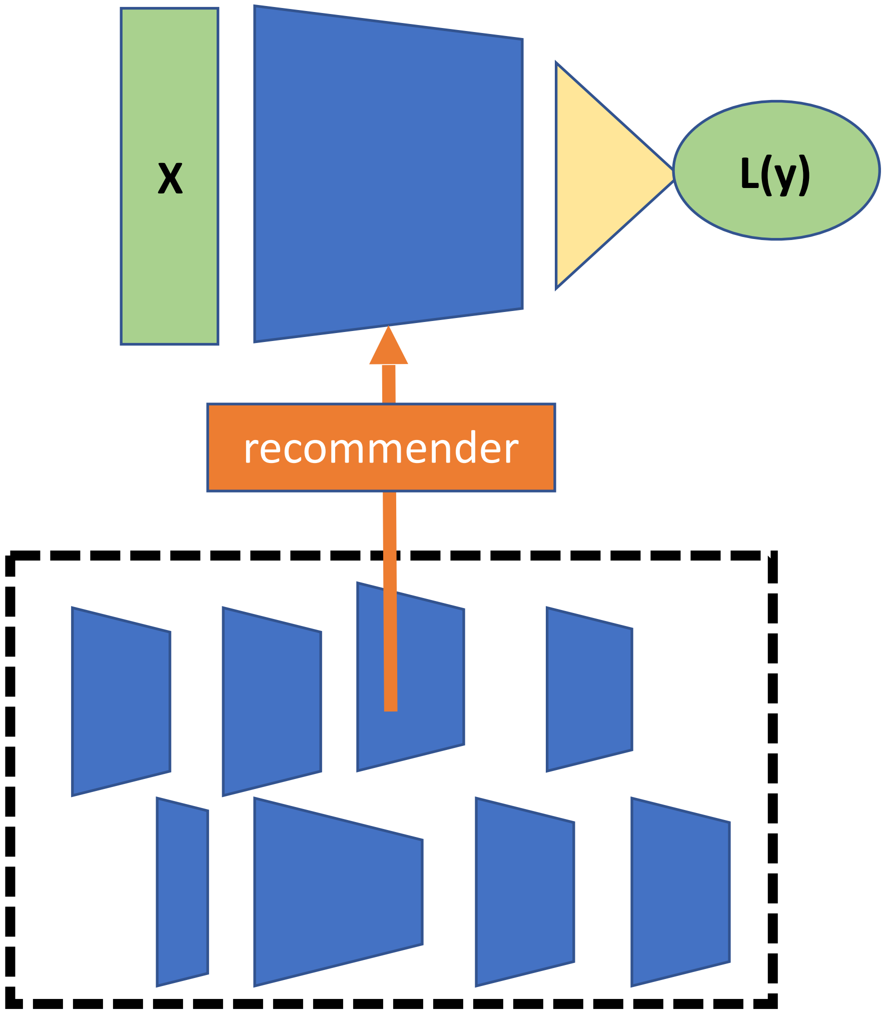
Task = {dataset, labels, loss}

$$x = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

What are similar tasks?
What architecture should I use?
What pre-training dataset?
What hyper parameters?
Do I need more training data?
How difficult is this task?
.
.
.

If we have a universal vectorial representation of tasks we can frame all sorts of interesting CV applications engineering problems as meta-learning problems

Model recommendation



Feature Extractor Zoo

Brute Force:

Input: Task = (**dataset, loss**)

For each feature extractor architecture **F**:
1. Train **classifier** on **F(dataset)**
2. Compute validation performance

Output: best performing model

Task recommendation:

Input: Task = (**dataset, loss**)

1. Compute task embedding **t = E(**Task**)**
2. Predict best extractor **F = M(t)**
2. Train **classifier** on **F(dataset)**
3. Compute validation performance

Output: best performing model

Task embedding using Fisher Information

1. Given a **task**, train a classifier with the **task loss** on features from a generic "probe network"
2. Compute gradients of probe network parameters w.r.t. task loss
3. Use statistics of the probe parameter gradients as the fixed dimensional task embedding

$$\mathrm{F} = \frac{1}{N} \sum_{i=1}^{N} \nabla \log p(x_i|\theta) \, \nabla \log p(x_i|\theta)^{\mathrm{T}}$$

Intuition: F provides information about the sensitivity of the task performance to small perturbations of parameters in the probe network

$$\mathbb{E}_{x \sim \hat{p}} \, KL \, p_{\theta'}(y|x) p_{\theta}(y|x) = \delta\theta \cdot F \cdot \delta\theta + o(\delta\theta^2),$$

Properties of TASK2VEC embedding

Dataset:

$$(x_i, y_i), i = 1 \dots n, \quad y_i \in \{0, 1\}$$

Classifier:

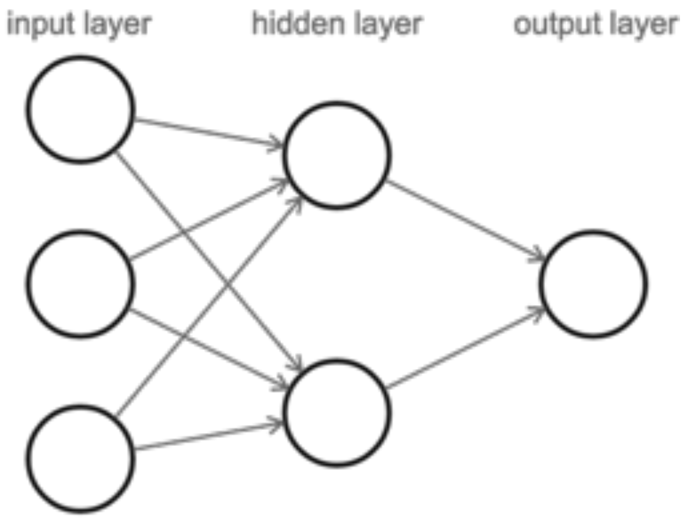
$$p_i = \sigma\left(w^T \phi(x_i)\right)$$

FIM for cross entropy loss for the last layer:

$$\frac{\partial \ell}{\partial w} = \frac{1}{N} \sum_i (y_i - p_i) \phi(x_i)$$

$$F_w = \frac{1}{N} \sum_i p_i (1 - p_i) \phi(x_i) \phi(x_i)^T$$

Two layer network



$$x \to \phi(x)$$

Properties of TASK2VEC embedding

Dataset:

$$(x_i, y_i), i = 1 \dots n, \quad y_i \in \{0, 1\}$$

Classifier:

$$p_i = \sigma\left(w^T \phi(x_i)\right)$$

FIM for cross entropy loss for the last layer:

$$\frac{\partial \ell}{\partial w} = \frac{1}{N} \sum_i (y_i - p_i) \phi(x_i)$$

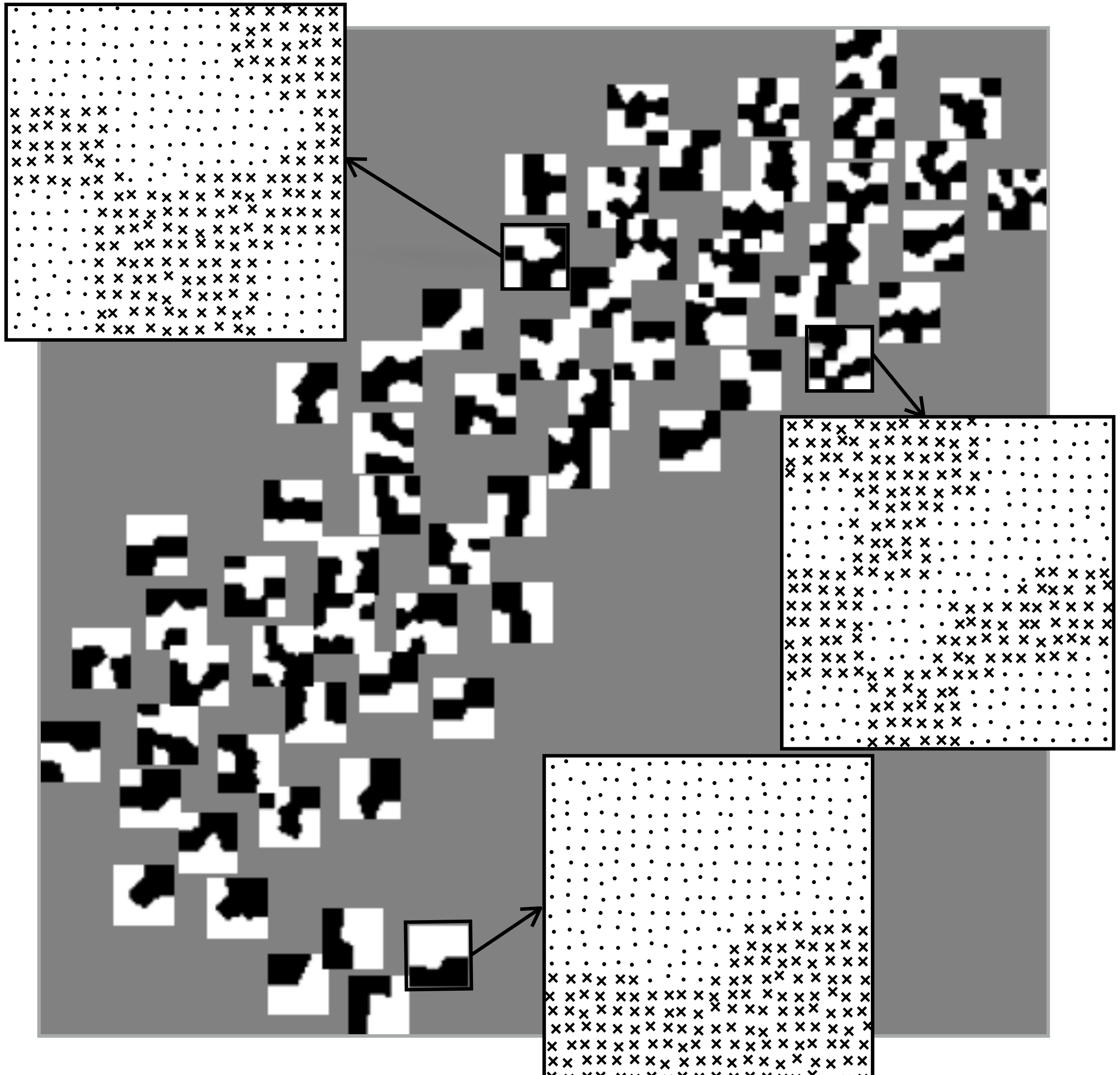
$$F_w = \frac{1}{N} \sum_i p_i (1 - p_i) \phi(x_i) \phi(x_i)^T$$

1. **Invariance** to label space
2. Encodes task **difficulty**
3. Encodes task **domain**
4. Encodes **useful features** for the task

Representative domain embedding

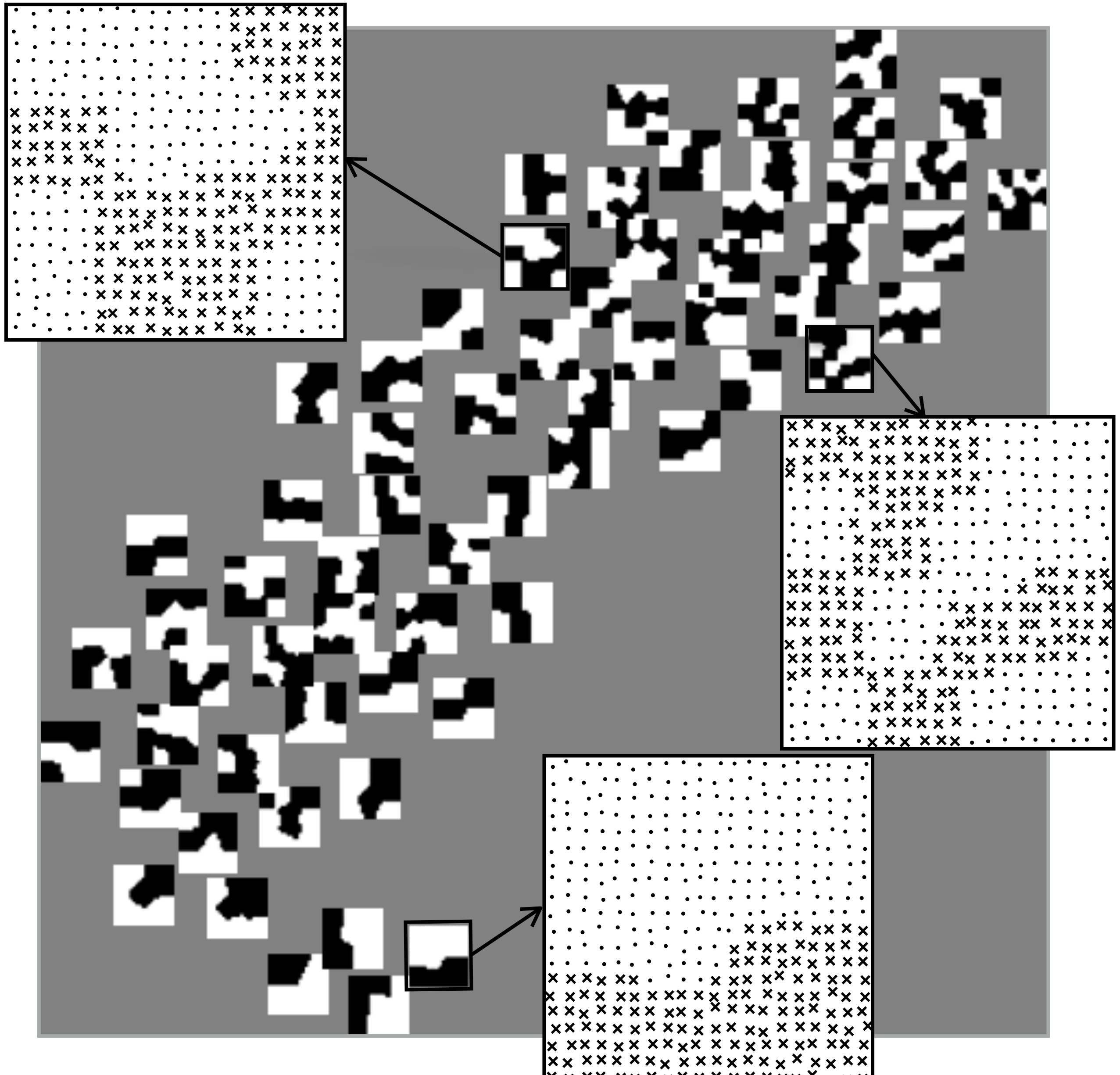
$$D = \frac{1}{N} \sum_i \phi(x_i) \phi(x_i)^T$$

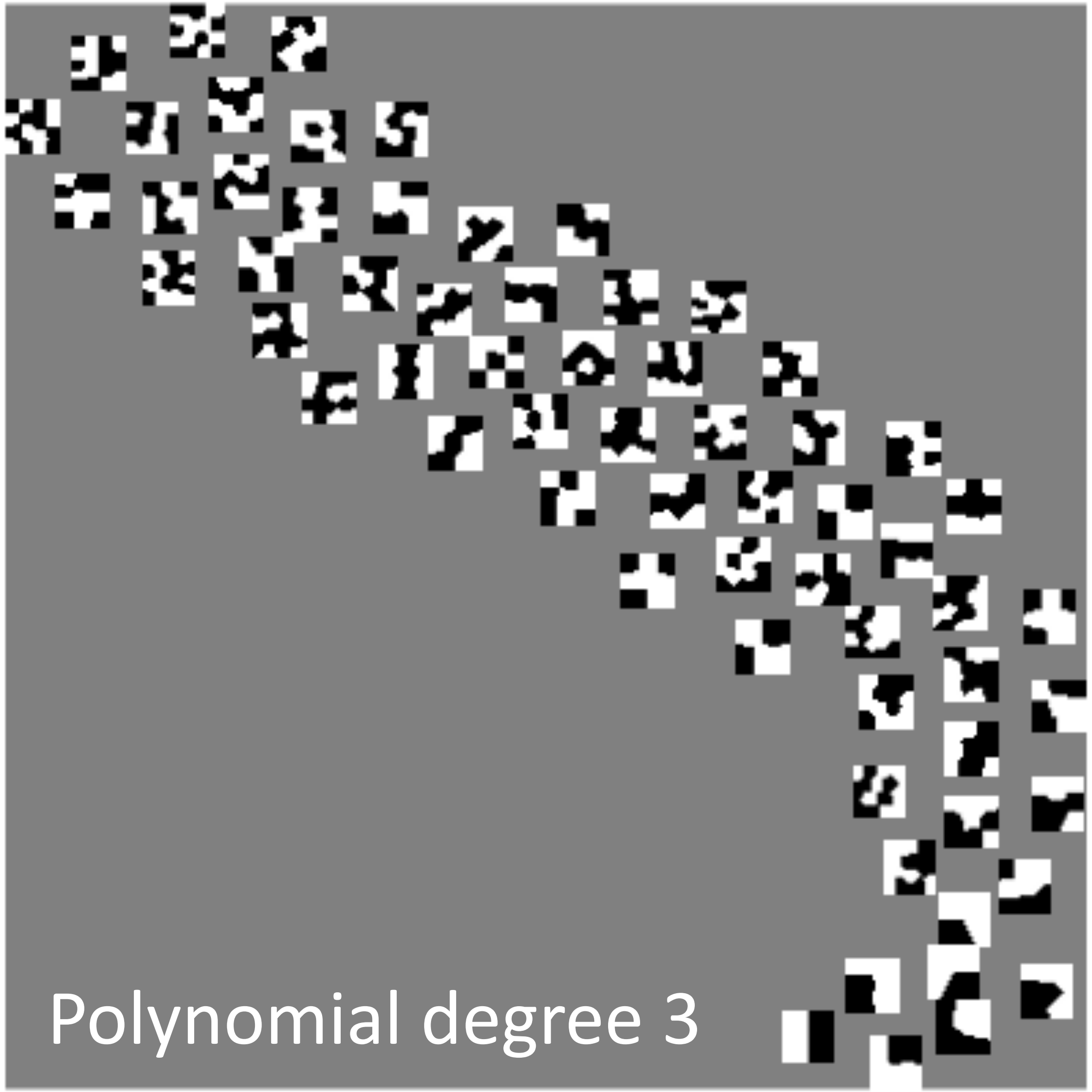
Properties of TASK2VEC embedding



1. Binary tasks on unit square, i.e., each tile is a task
2. 10 Random ReLU features, i.e., $\phi_i = \max(0, a_i x + b_i y + c_i)$
3. T-SNE to map 10x10 FIM to 2D

Properties of TASK2VEC embedding





Robust Fisher Computation

1. For realistic CV tasks we want to use deep CNNs (e.g., ResNet) and estimate FIM for all the parameters.
2. <u>Challenge:</u> FIM can be hard to estimate (noisy loss landscape; high dimensions; small training set)
3. Robust FIM
   1. Restrict it to a diagonal
   2. Restrict it a single value per filter (CNN layer)
   3. Robust estimation via perturbation

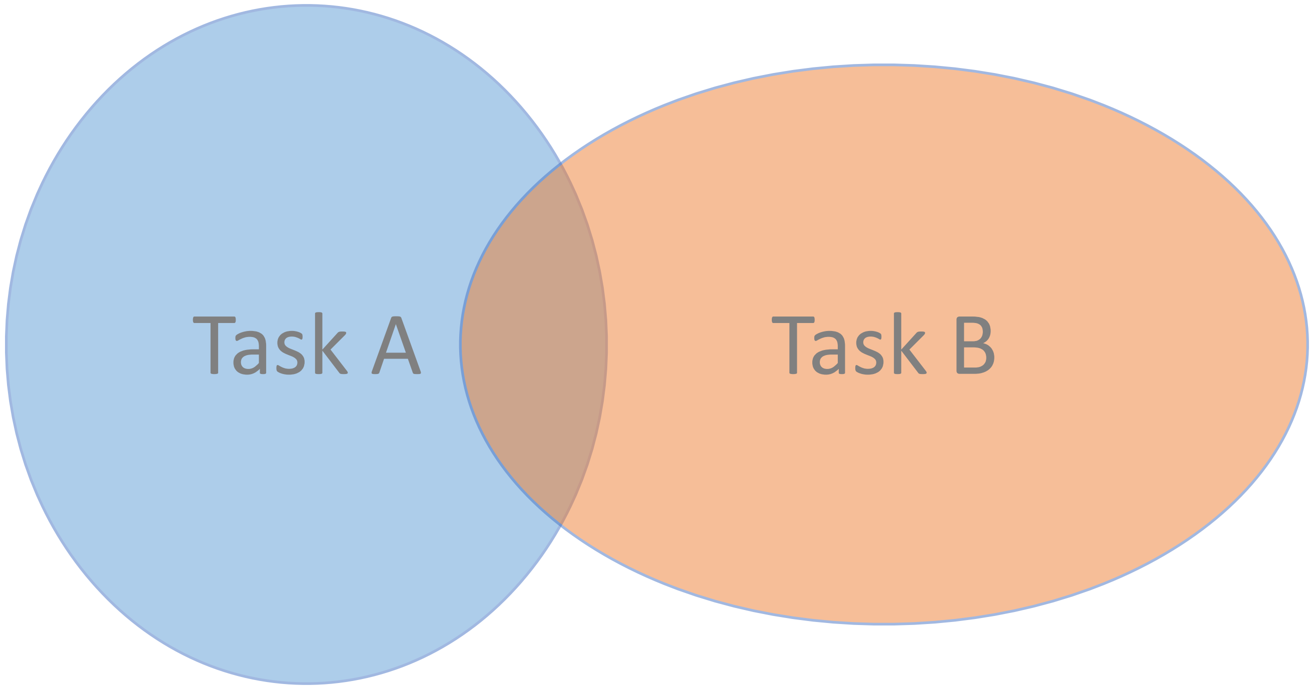
Estimate Λ of a Gaussian perturbation:

$$L(\hat{w};\Lambda) = \mathbb{E}_{w\sim\mathcal{N}(\hat{w},\Lambda)}[H_{p_w,\hat{p}}\,p(y|x)] + \beta\, KL(\mathcal{N}(0,\Lambda)\,\|\,\mathcal{N}(0,\lambda^2 I))$$

Optimal Λ satisfies:

$$\frac{\beta}{2N}\Lambda = F + \underbrace{\frac{\beta\lambda^2}{2N} I}_{\text{"Trivial Embedding"}}$$

Similarity measures on the space of tasks



Task = {dataset, labels, loss}

Similarity measures on the space of tasks

Domain similarity

Unbiased look at dataset bias, Torralba and Efros, CVPR 11

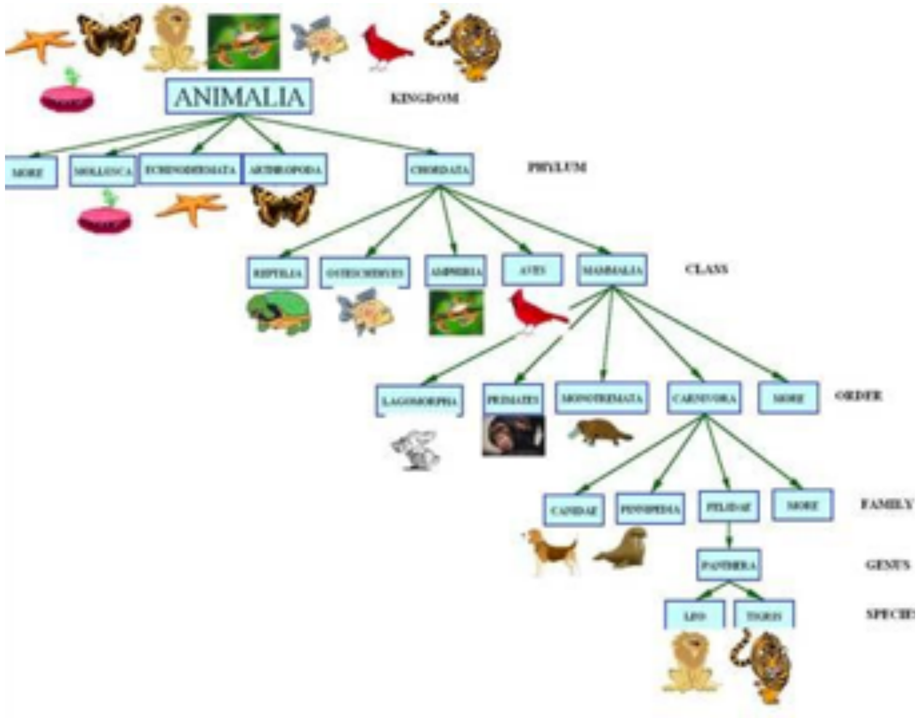
Similarity measures on the space of tasks

Domain similarity

<u>Range / label similarity</u>

- e.g., Taxonomic distance

$$D_{\text{tax}}(t_a, t_b) = \min_{i \in S_a, j \in S_b} d(i, j),$$



https://www.pinterest.com/pin/520799144386337065/

Similarity measures on the space of tasks

Domain similarity

Range / label similarity

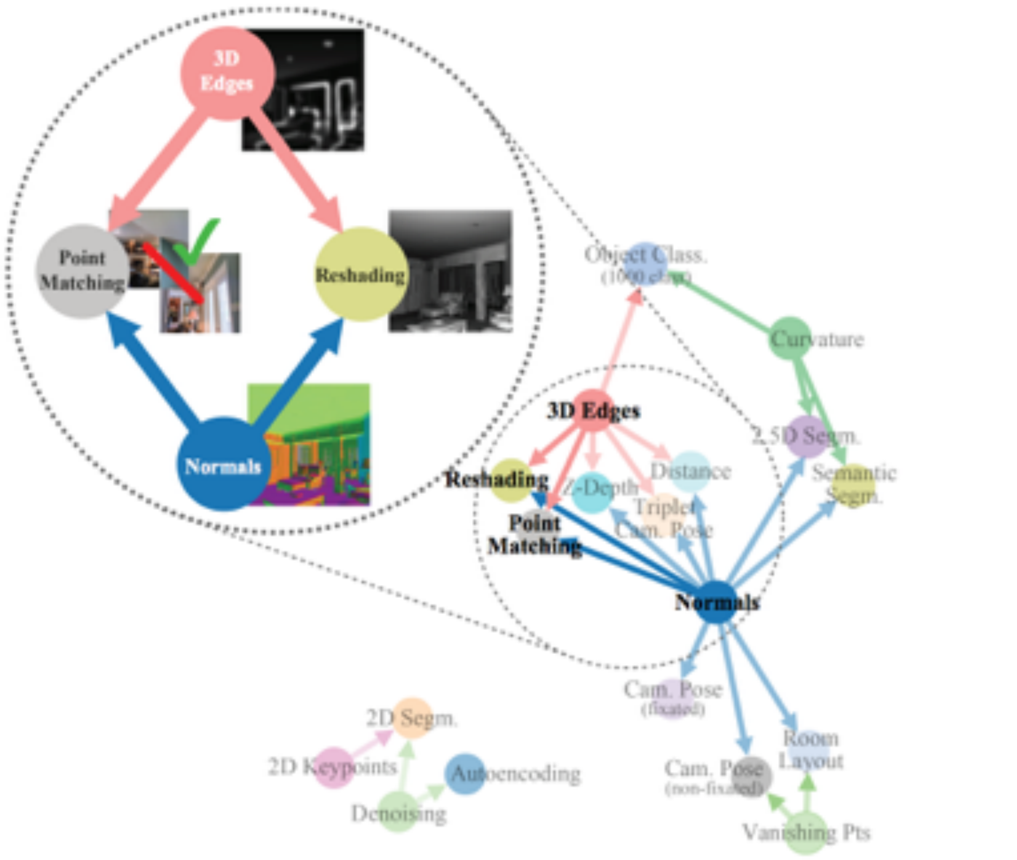
- e.g., Taxonomic distance

$$D_{\text{tax}}(t_a, t_b) = \min_{i \in S_a, j \in S_b} d(i, j),$$

<u>Transfer "distance"</u>

- Fine-tune on a followed by b

$$D_{\text{ft}}(t_a \to t_b) = \frac{\mathbb{E}[\ell_{a \to b}] - \mathbb{E}[\ell_b]}{\mathbb{E}[\ell_b]}$$



Taskonomy: Disentangling Task Transfer Learning, Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, Silvio Savarese, CVPR 18

Distance measures on TASK2VEC embedding

Symmetric distance

$$d_{\text{sym}}(F_a, F_b) = d_{\cos}\Big(\frac{F_a}{F_a + F_b}, \frac{F_b}{F_a + F_b}\Big)$$

Asymmetric "distance"

$$d_{\text{asym}}(t_a \to t_b) = d_{\text{sym}}(t_a, t_b) - \alpha d_{\text{sym}}(t_a, t_0)$$

MODEL2VEC: Joint embedding of tasks and models

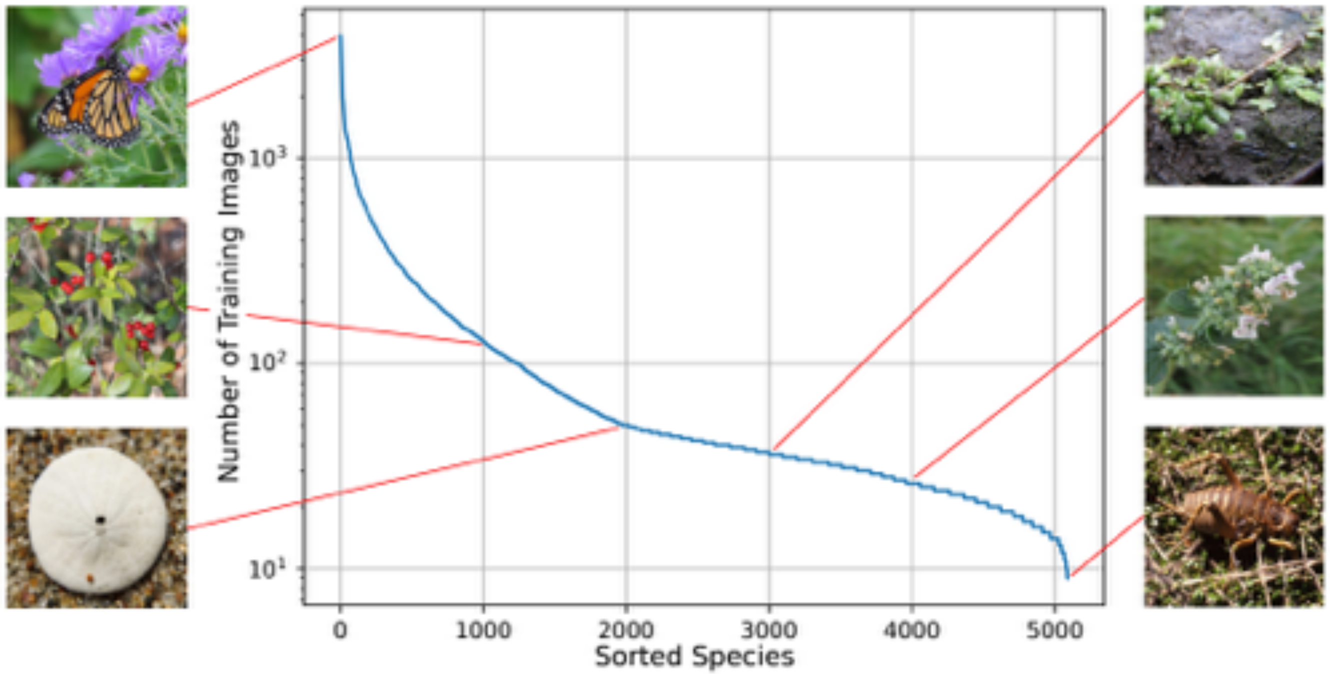
1. So far we have been associating models (feature extractors) with the tasks they are trained on.
2. How about
   1. legacy / black-box feature extractors? E.g., SIFT, HOG, Fisher vector
   2. models of different complexity trained on the same dataset
3. MODEL2VEC: Jointly embed feature extractors (encoded as one-hot-vectors) and tasks such that similarity reflects a meta-task objective.
   1. Needs training data

Task Zoo

- Tasks [1460]
  - iNaturalist [207]
  - CUB 200 [25]
  - iMaterialist [228]
  - DeepFashion [1000]

Task Zoo

- Tasks [1460]
  - **iNaturalist [207]**
  - CUB 200 [25]
  - iMaterialist [228]
  - DeepFashion [1000]



Super-Class	Class
Plantae	2,101
Insecta	1,021
Aves	964
Reptilia	289
Mammalia	186
Fungi	121
Amphibia	115
Mollusca	93
Animalia	77
Arachnida	56
Actinopterygii	53
Chromista	9
Protozoa	4

Task Zoo

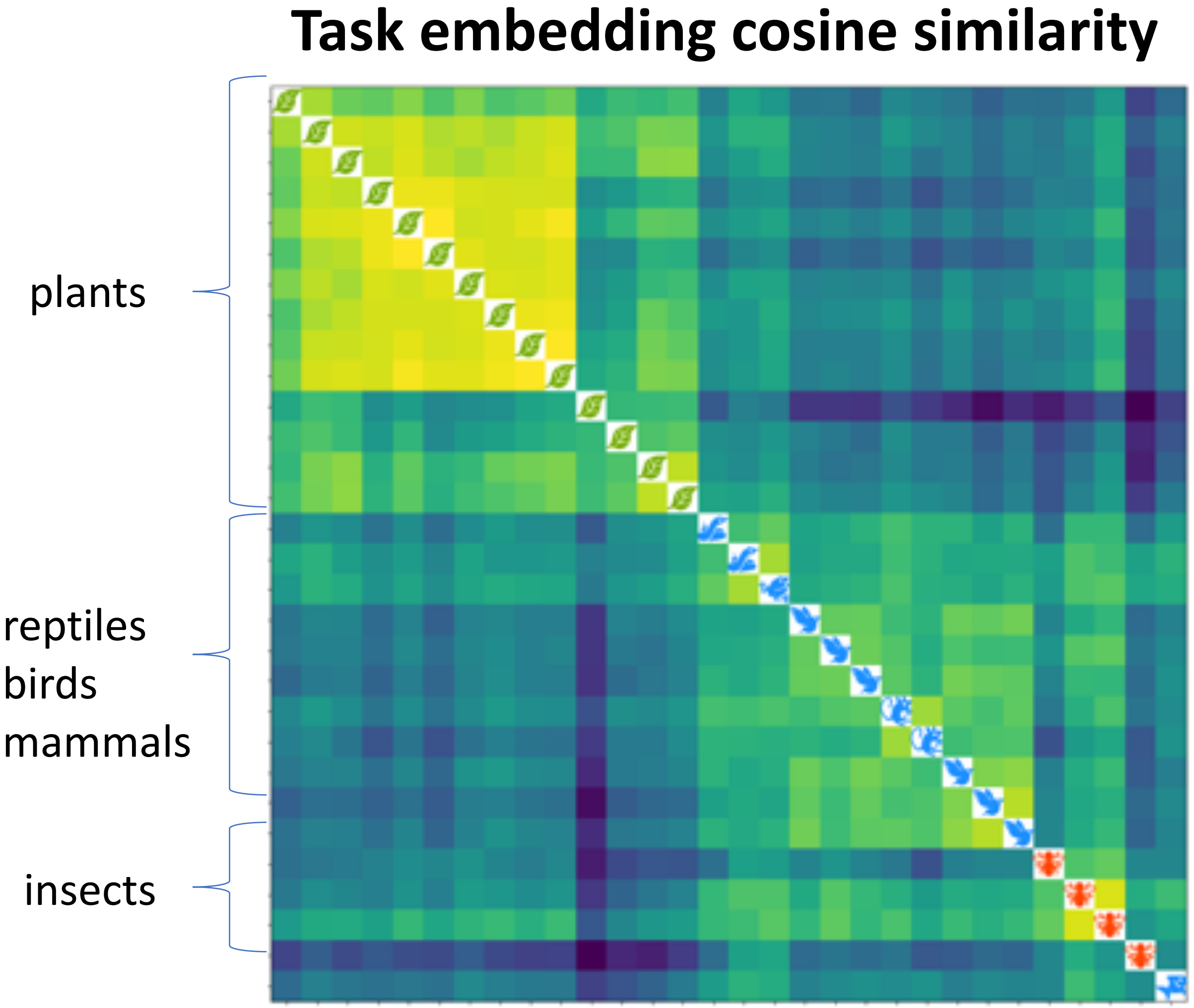
- Tasks [1460]
  - iNaturalist [207]
  - CUB 200 [25]
  - iMaterialist [228]
  - **DeepFashion [1000]**

Task Zoo

- Tasks [1460]
  - iNaturalist [207]
  - CUB 200 [25]
  - iMaterialist [228]
  - **DeepFashion [1000]**

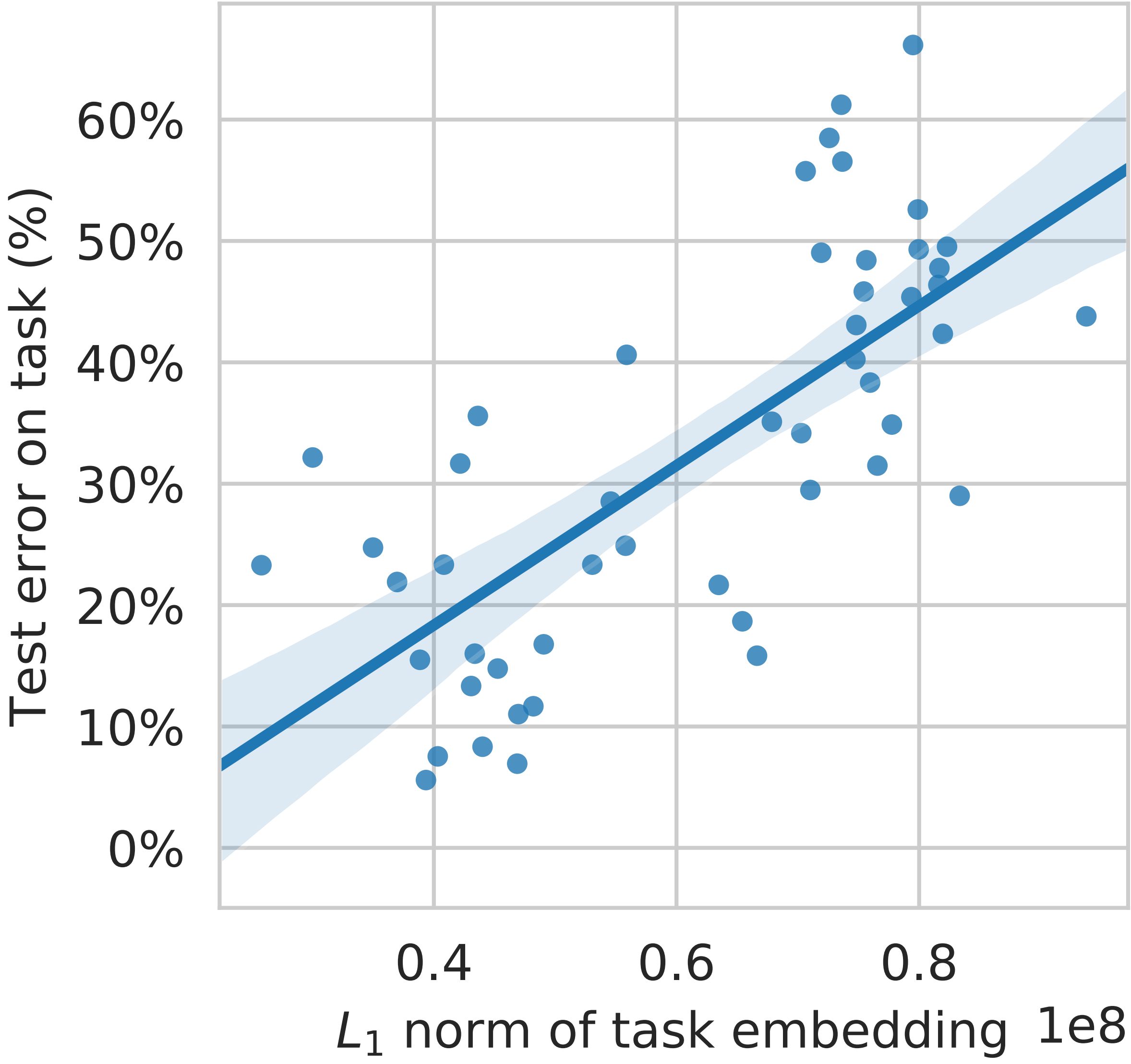
- Few tasks > 10K training samples but most have 100-1000 samples

Experiment: TASK2VEC recapitulates iNaturalist taxonomy



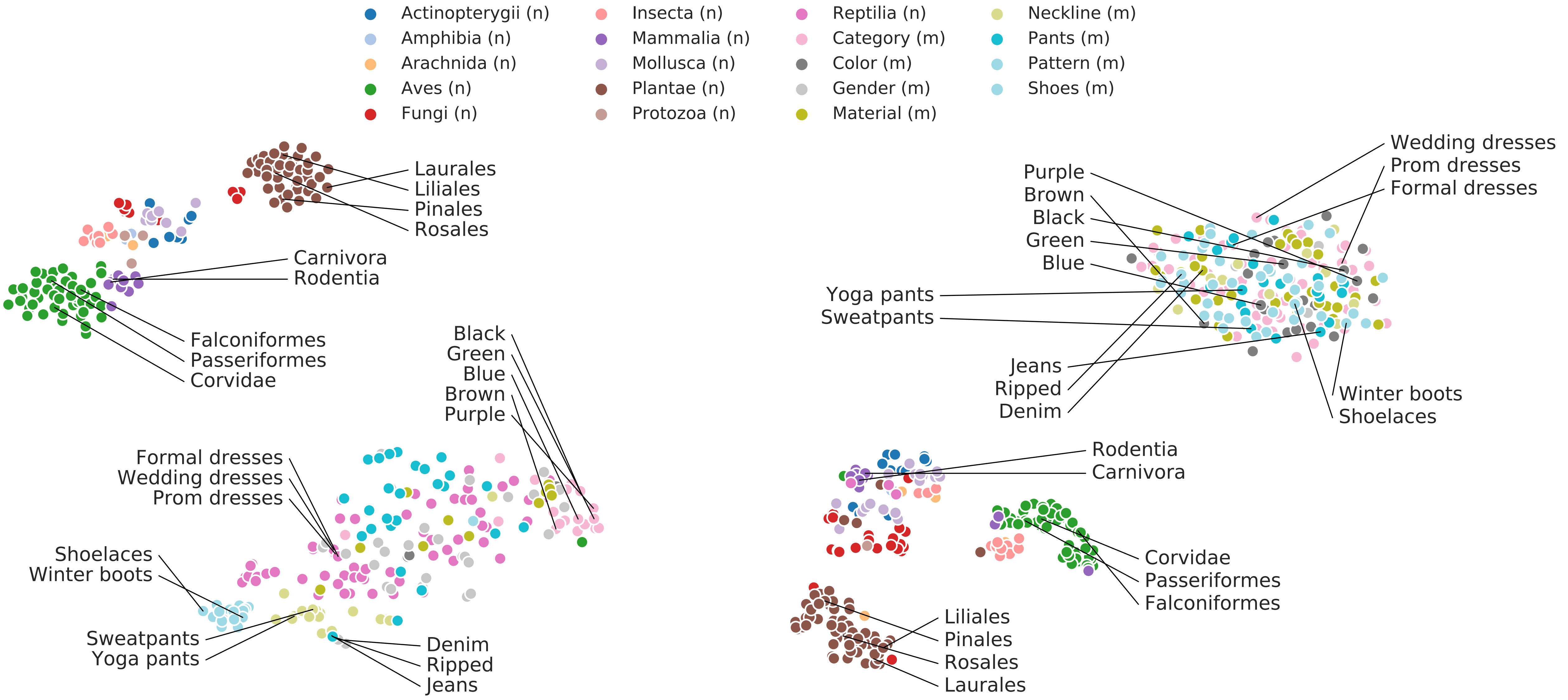
ResNet trained on ImageNet as probe network

Experiment: TASK2VEC norm encodes task difficulty



ResNet trained on ImageNet as probe network

Experiment: TASK2VEC vs DOMAIN2VEC



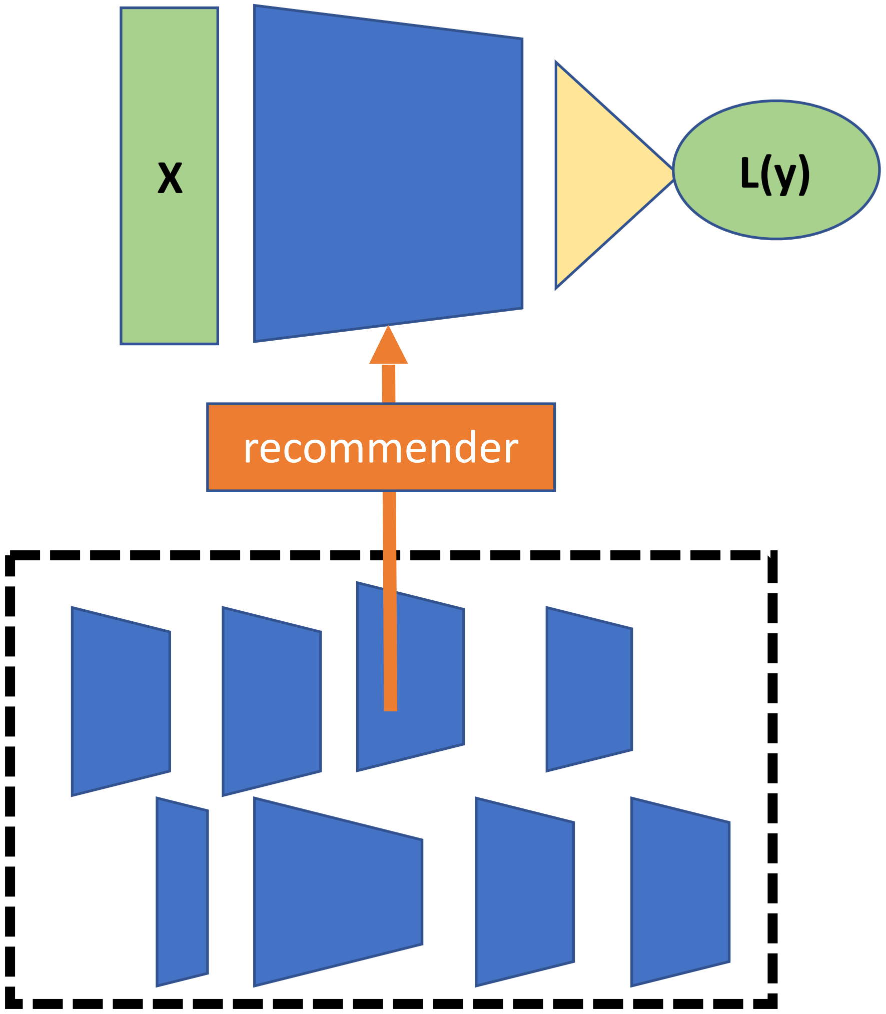
Task Embeddings

Domain Embeddings

Task and Feature Zoo

- Tasks [1460]
  - iNaturalist [207]
  - CUB 200 [25]
  - iMaterialist [228]
  - DeepFashion [1000]

- Feature Zoo [156 experts]
  - ResNet-34 pertained on ImageNet
  - Followed by fine-tuned on tasks with enough examples

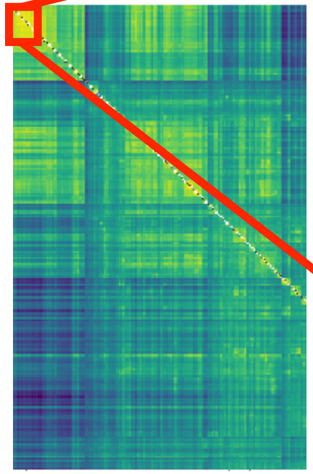


Feature Extractor Zoo


The Matrix
Tasks
Feature extractors

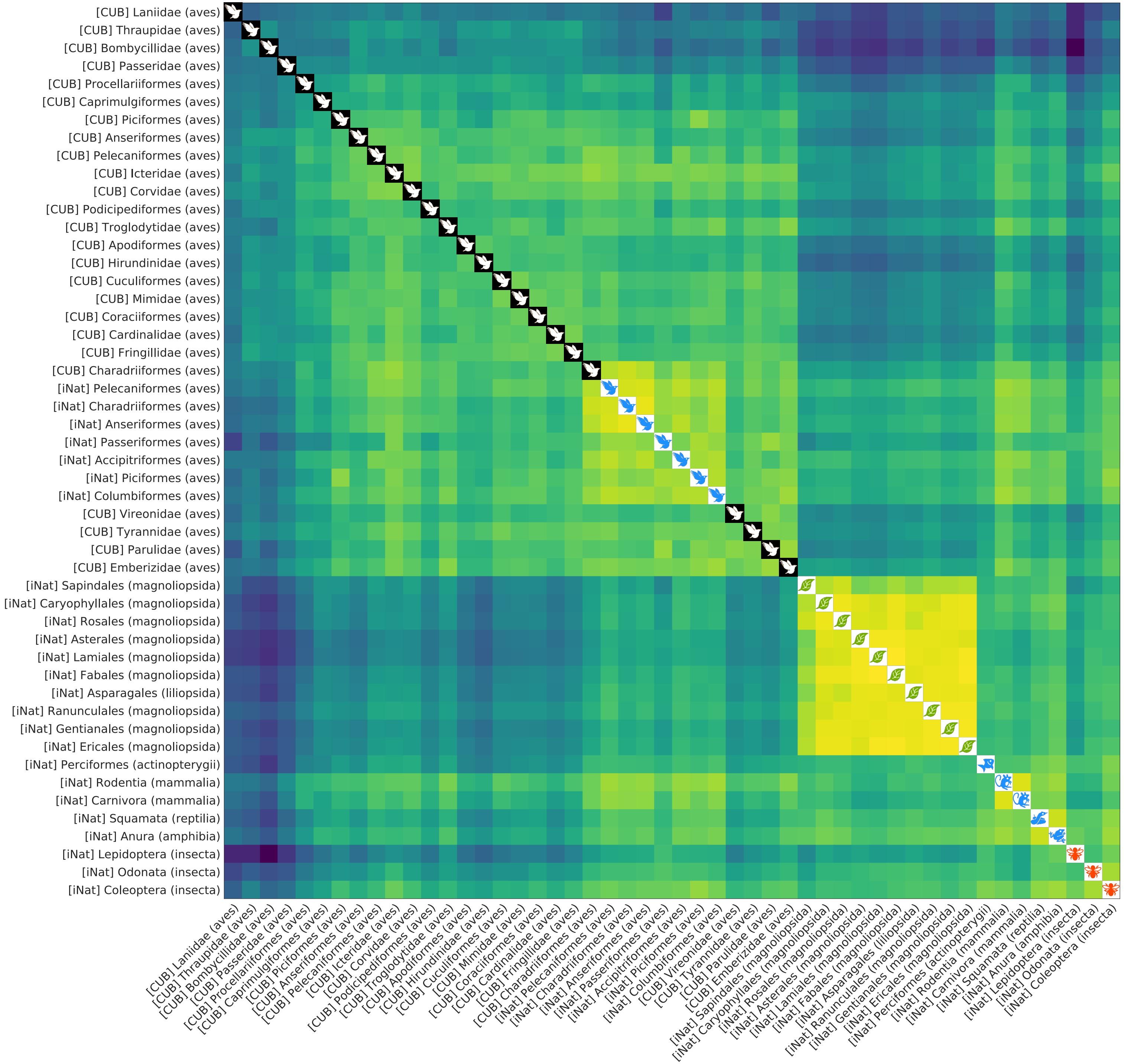
The Matrix

Tasks

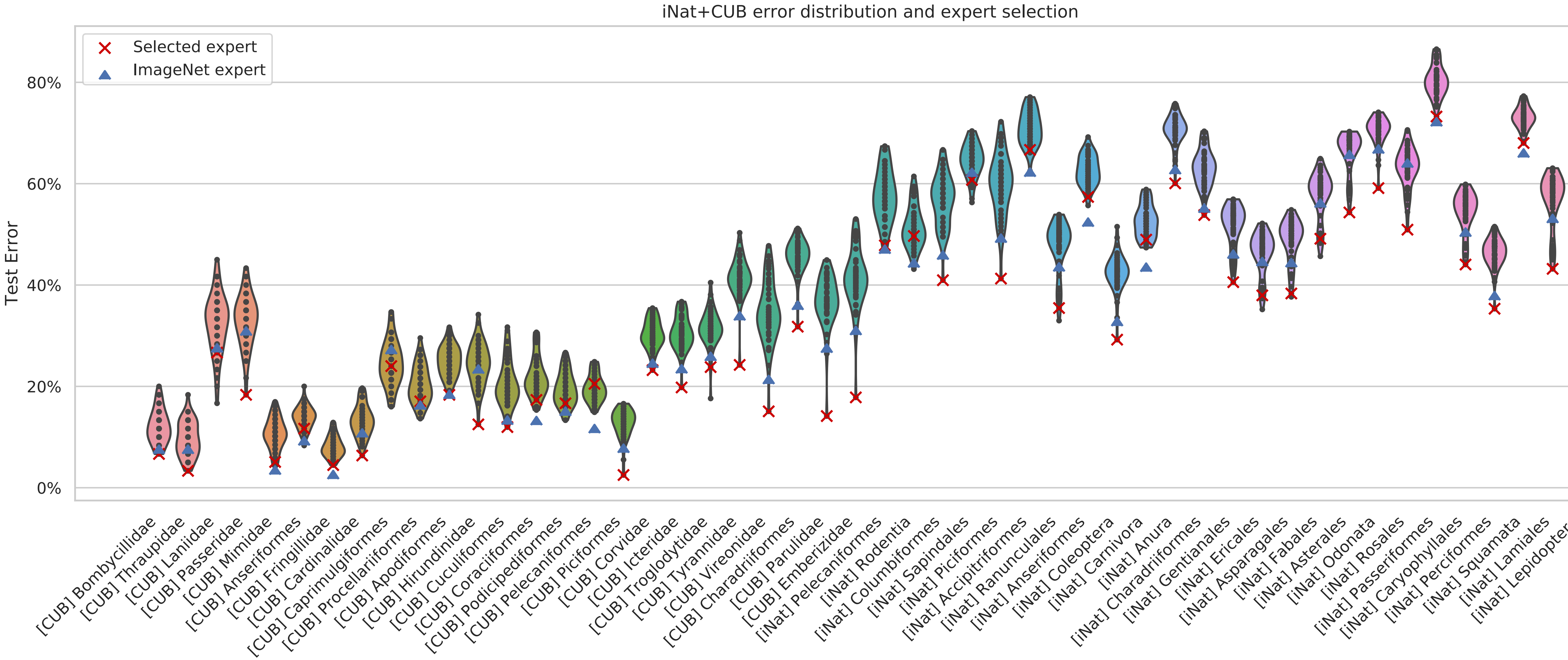


Experts

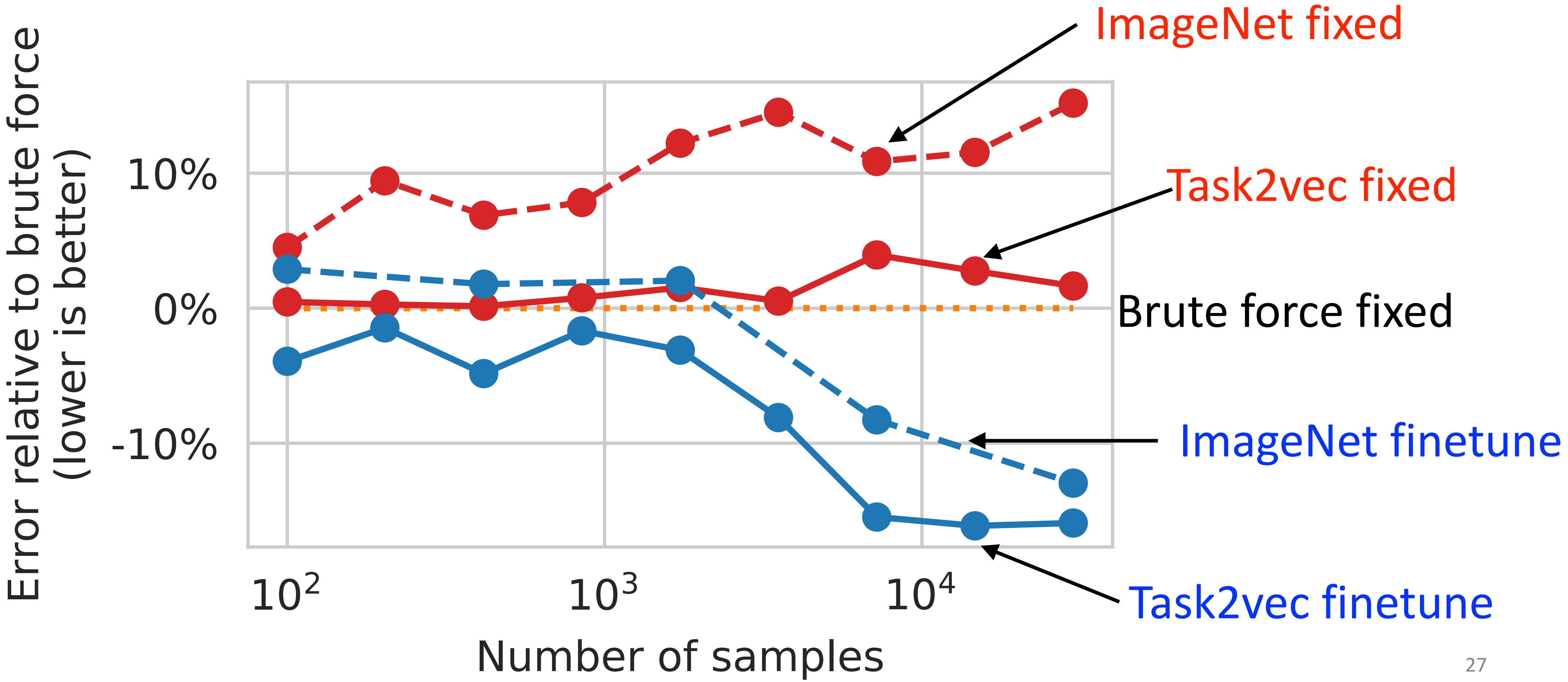
iNaturalist + CUB



ImageNet expert is usually good but on many tasks the best expert handily outperforms the ImageNet expert



Data efficiency of TASK2VEC



Choice of distance for TASK2VEC

Meta-task	Optimal	Chance	ImageNet	TASK2VEC	Asymmetric TASK2VEC	MODEL2VEC
iNat + CUB	31.24	+59.52%	+30.18%	+42.54%	+9.97%	**+6.81%**
Mixed	22.90	+112.49%	+75.73%	+40.30%	+29.23%	**+27.81%**

Relative error increase over the oracle (best choice)

Choice of the probe network on TASK2VEC

Probe network	Top-10	All
Chance	+13.95%	+59.52%
VGG-13	+4.82%	+38.03%
DenseNet-121	+0.30%	+10.63%
ResNet-13	**+0.00%**	**+9.97%**

Relative error increase over the oracle (best choice)

Thank you!







Task2Vec: Task Embedding for Meta-Learning, Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, Pietro Perona (https://arxiv.org/abs/1902.03545)