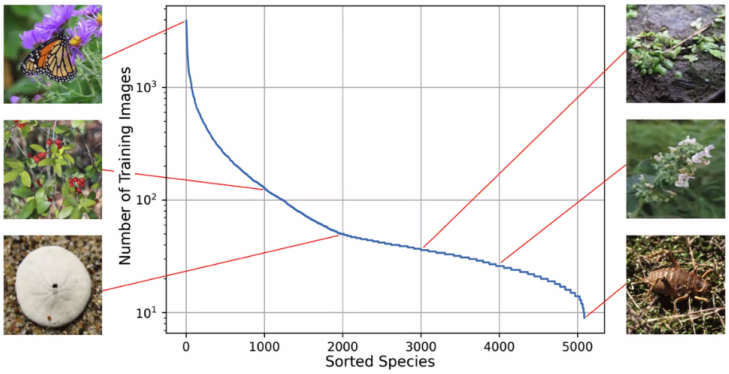
Day 15 :

Speaker : Prof Subhransu Maji , University of Massachusetts.

Title : Learning from Little data

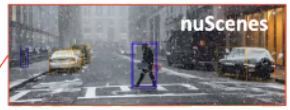
The problem with the tail

	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



The iNaturalist Species Classification and Detection Dataset, Grant Van Horn et al., CVPR 2018

Tails are everywhere ...



1.4million images, 390k lidar

CatCam dataset, Belinda et al., 2004
17 unlabeled videos

14+million images, 21k classes

23 spectral band images of 5 paintings
Polykarpos et al., 2017

Computer vision datasets: http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm

Transfer learning to the rescue

Models trained on ImageNet transfer well to other vision tasks

- What about other pre-trained models (e.g., iNaturalist)?
- What about novel domains (e.g., medical images)?

We lack a general framework to reason about tasks and domains

Why?

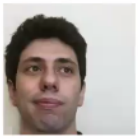
- Few-shot learning (e.g., recommend the right expert)
- Multi-tasking (e.g., which tasks are mutually beneficial)
- Semi/self-supervised learning on which dataset?

Talk outline

I. Task2Vec: Vector representations of tasks [ICCV'19]

- Captures task similarity, task difficulty, etc.
- Useful for model recommendation

aws



Alessandro Achille

II. When does self-supervision improve few-shot learning?[CVPR'20]

III. A realistic-evaluation of semi-supervised learning [CVPR'21]

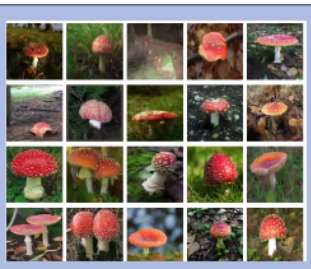
- Self/semi—supervision is effective as long as the unlabeled images are from a similar domain
- Requires a notion of domain similarity

Jong-Chyi Su

Zezhou Cheng

Task Embedding

Alessandro, Michael, Rahul, Avinash, Subhransu, Charless, Stefano, Pietro @ ICCV'19



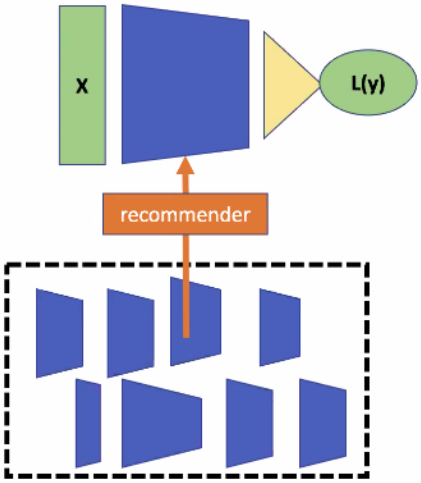
Task = {images, labels, loss}

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

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 computer vision application engineering problems as meta-learning problems.

Model recommendation



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:

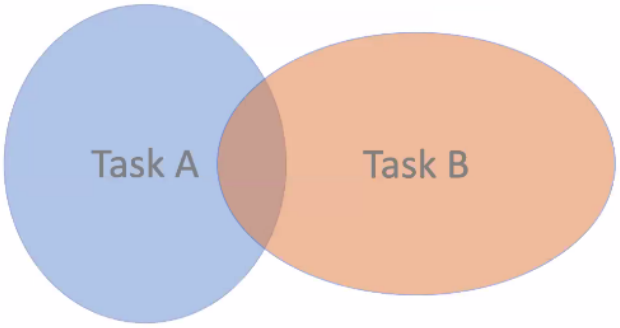
Input: Task = (dataset, loss)

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

Train ahead of time.

Similarity measures on the space of tasks



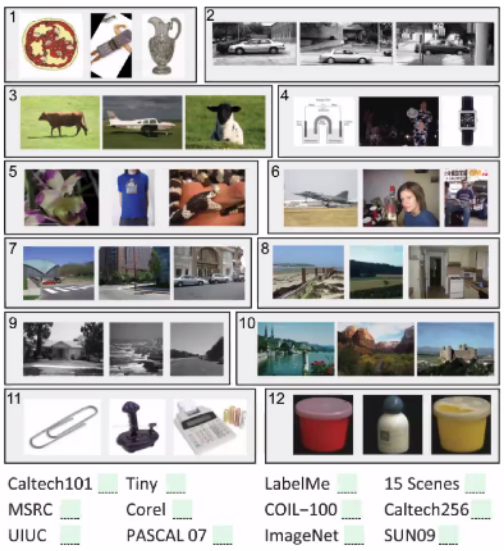
Task = {images, labels, loss}

Dataset = {images, labels}

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

Range / label similarity

- e.g., Taxonomic distance

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

D(*bird* task, *mammal* task) < D(*bird* task, *worm* task)

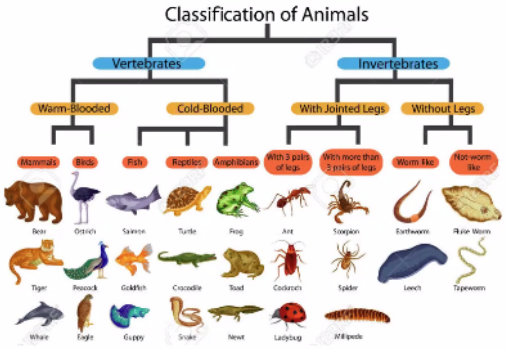


image source: https://www.123rf.com/photo_80712766_stock-vector-education-chart-of-biology-for-classification-of-animals-diagram.html

Similarity measures on the space of tasks

Domain similarity

Range / label similarity

- e.g., Taxonomic distance

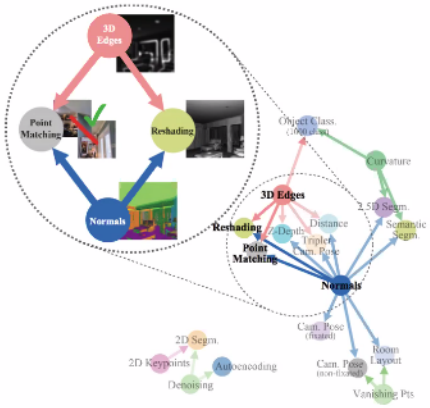
$$D_{\text{tax}}(t_a, t_b) = \min_{i \in S_a, j \in S_b} d(i, j).$$

Transfer "distance"

- Fine-tune on task *a* followed by *b*

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

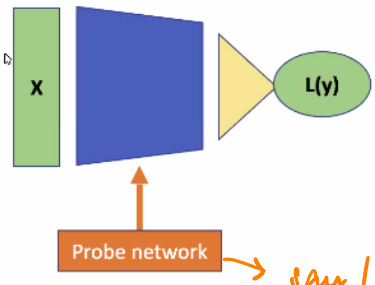
Computationally Expensive



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

Task embedding using a probe network

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**



say (any general net like AlexNet, ResNet etc)

find out which part of P.N is useful for solving the task

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

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

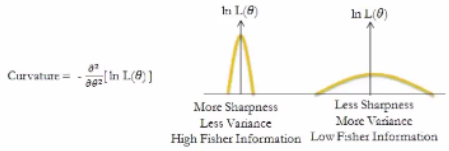


Figure source: https://www.gaussianwaves.com/2012/10/score-and-fisher-information-estimator-sensitivity

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. **Challenge:** 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:

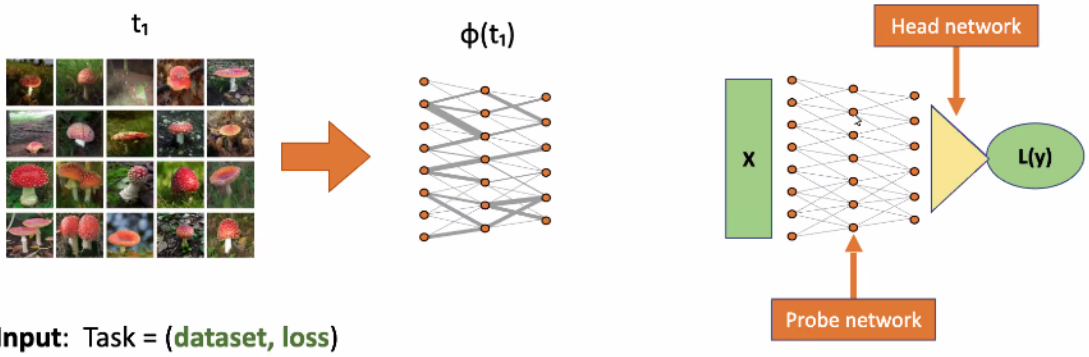
$$L(\hat{w};\Lambda) = \mathbb{E}_{w\sim\mathcal{N}(\hat{w},\Lambda)}[H_{p_w,\hat{p}}p(y|x)] + \beta\, KL(\mathcal{N}(0,\Lambda)\,\|\,\mathcal{N}(0,\lambda^2 I))$$

Optimal Λ satisfies:

$$\frac{\beta}{2N}\Lambda = F + \frac{\beta\lambda^2}{2N}I$$

"Trivial Embedding"

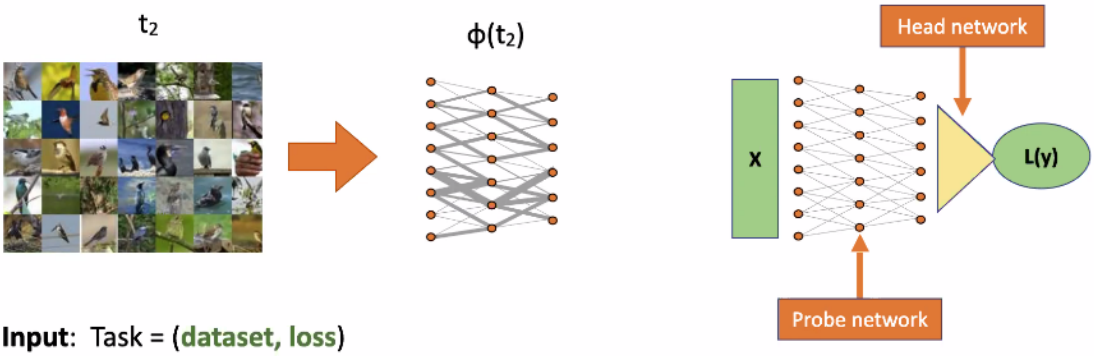
Task embedding illustration



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

1. Initialize the **probe network** and the **head network** (e.g., **linear classifier**)
2. Train the **head network** by minimizing the loss
3. Compute the (approximate) FIM of the **probe network**

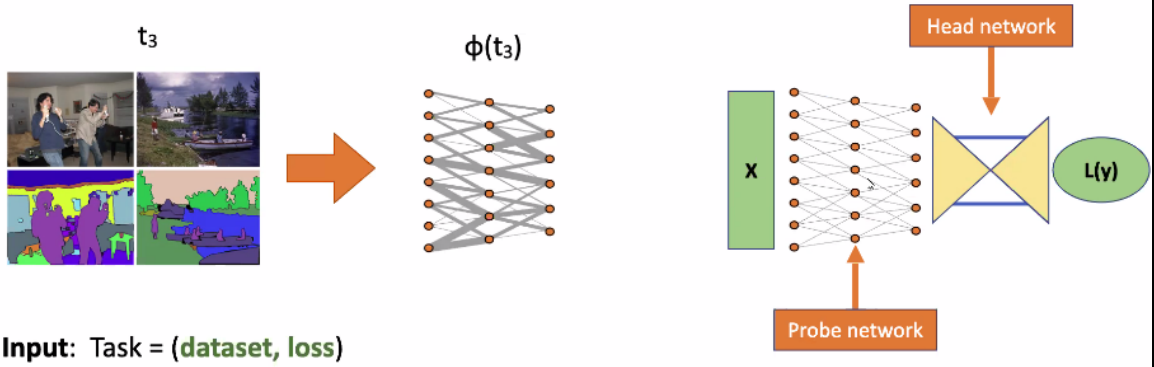
Task embedding illustration



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

1. Initialize the **probe network** and the **head network** (e.g., **linear classifier**)
2. Train the **head network** by minimizing the loss
3. Compute the (approximate) FIM of the **probe network**

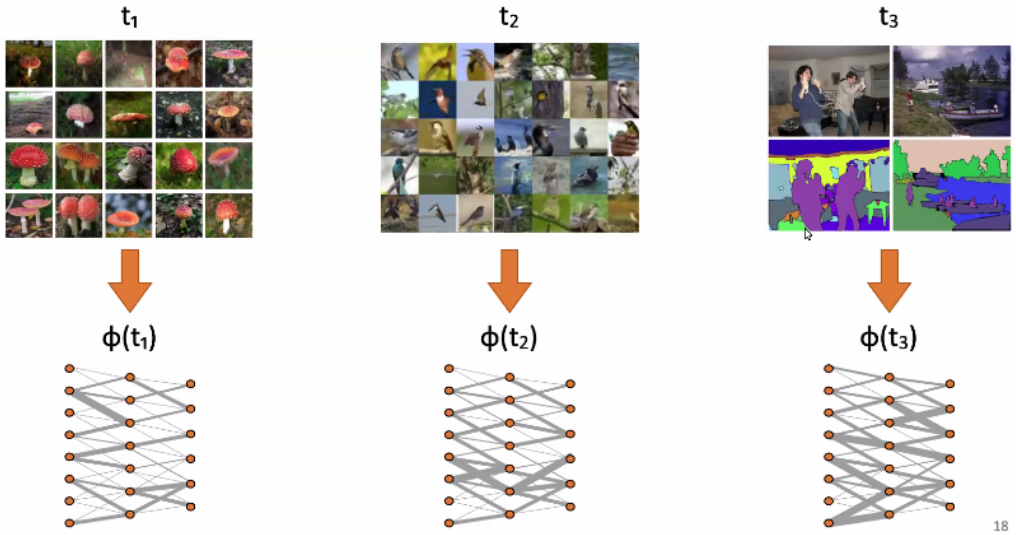
Task embedding illustration



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

1. Initialize the **probe network** and the **head network** (e.g., **UNet**)
2. Train the **head network** by minimizing the loss
3. Compute the (approximate) FIM of the **probe network**

Task embedding illustration



Properties of TASK2VEC embedding

Dataset

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

Classifier

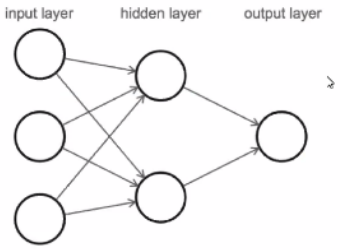
$$p_i = \sigma\left(w^T \phi(x_i)\right)$$

FIM of the last layer (cross-entropy)

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

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

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 of the last layer (cross-entropy)

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

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

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

Representative "domain embedding"

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

Distance measures on TASK2VEC embedding

Symmetric distance

Task Segmentation

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

Asymmetric "distance"

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

task embedding for the "trivial" task

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]**

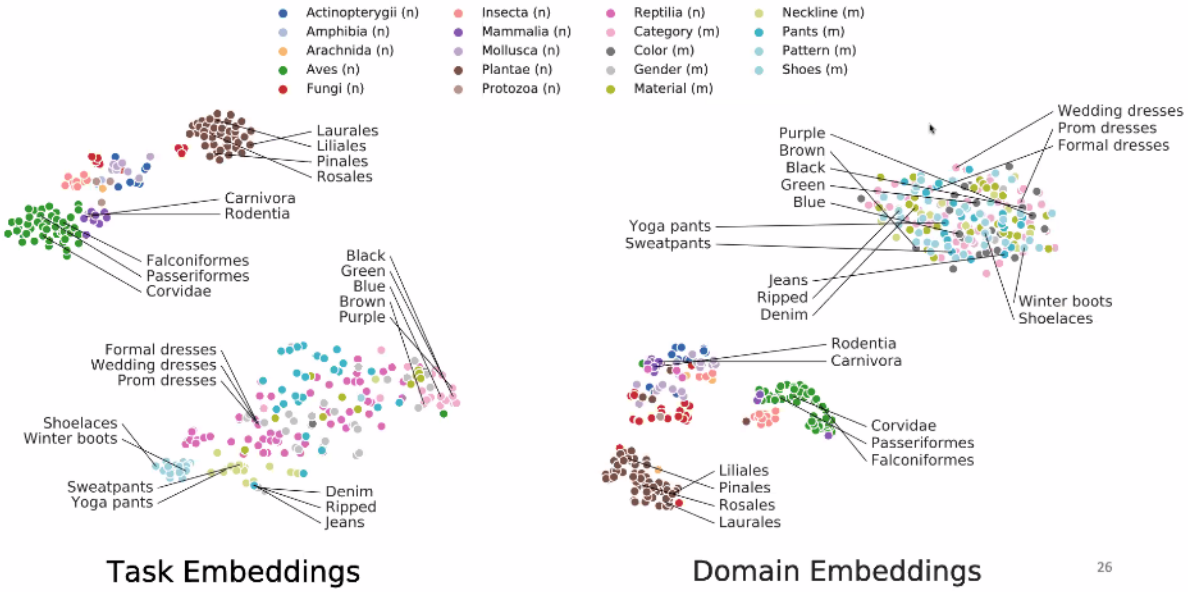
Task Zoo

Tasks [1460]

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

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

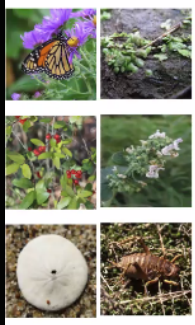
Experiment: Task2vec vs Domain2vec



Task Embeddings

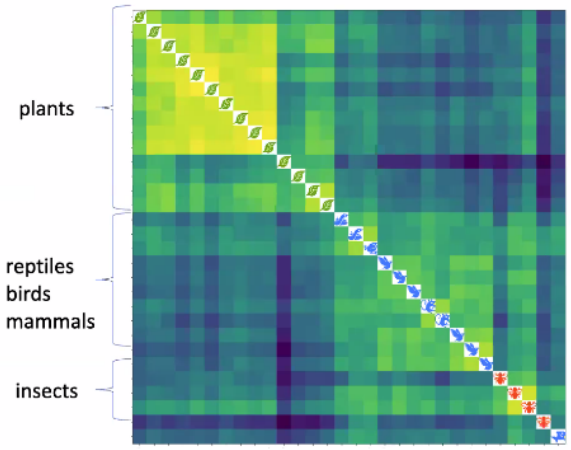
Domain Embeddings

Experiment: Task2vec recapitulates iNaturalist taxonomy



Super-Class
Plantae
Insecta
Aves
Reptilia
Mammalia

Task embedding cosine similarity

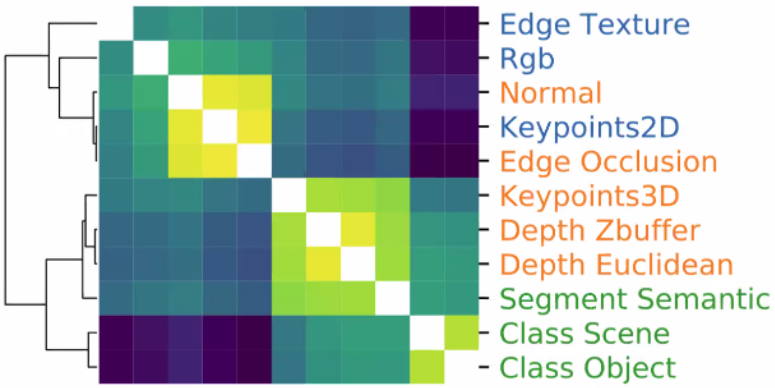


ResNet trained on ImageNet as probe network

Experiment: Task2vec recovers "Taskonomy"

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

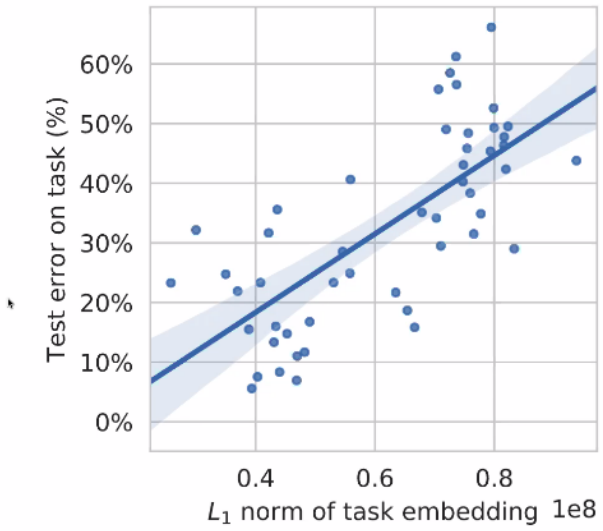
Task embedding cosine similarity



Classifier "head" replaced by a fully-convolutional layers.

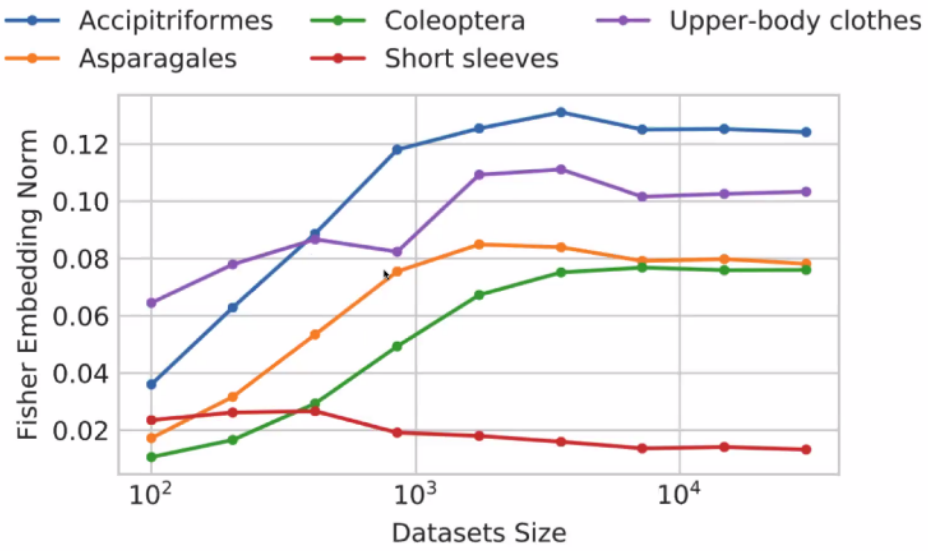
Requires far less compute (5 GPU hours for the whole matrix).

Experiment: TASK2VEC norm encodes task difficulty



ResNet trained on ImageNet as probe network

Experiment: TASK2VEC norm encodes task difficulty



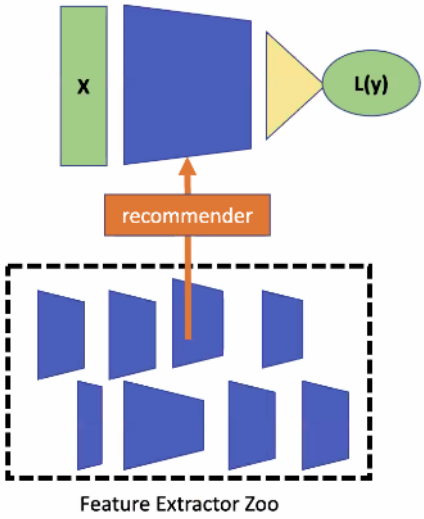
Task and Model Zoo

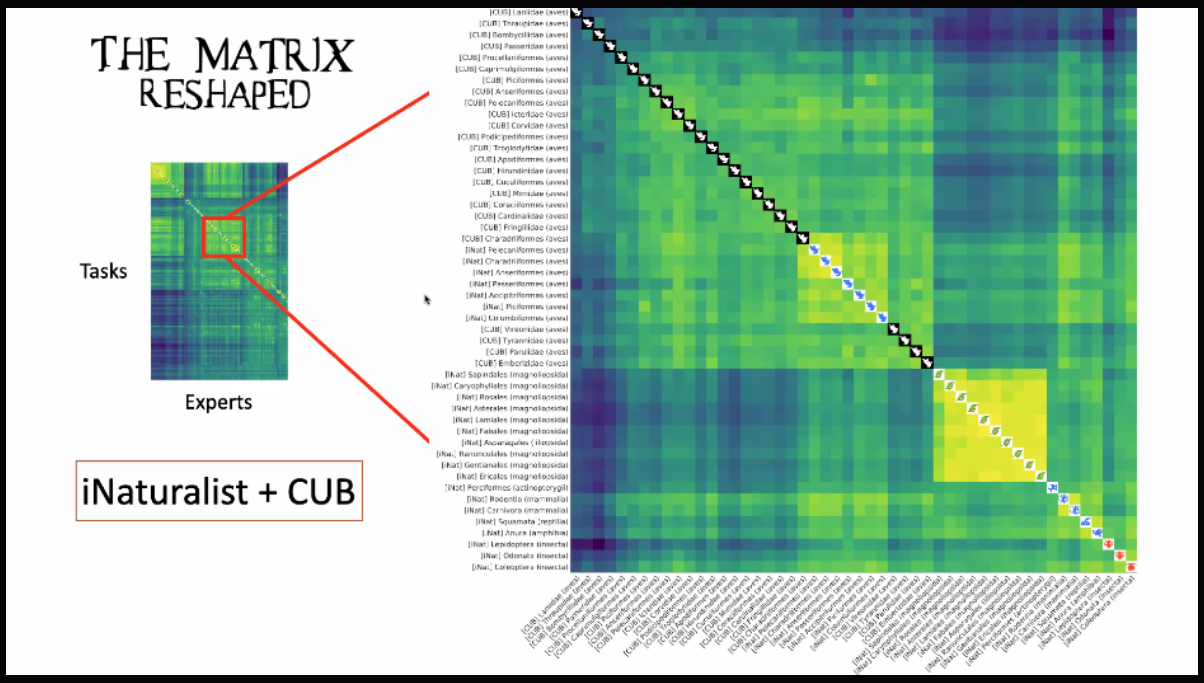
Tasks [1460]

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

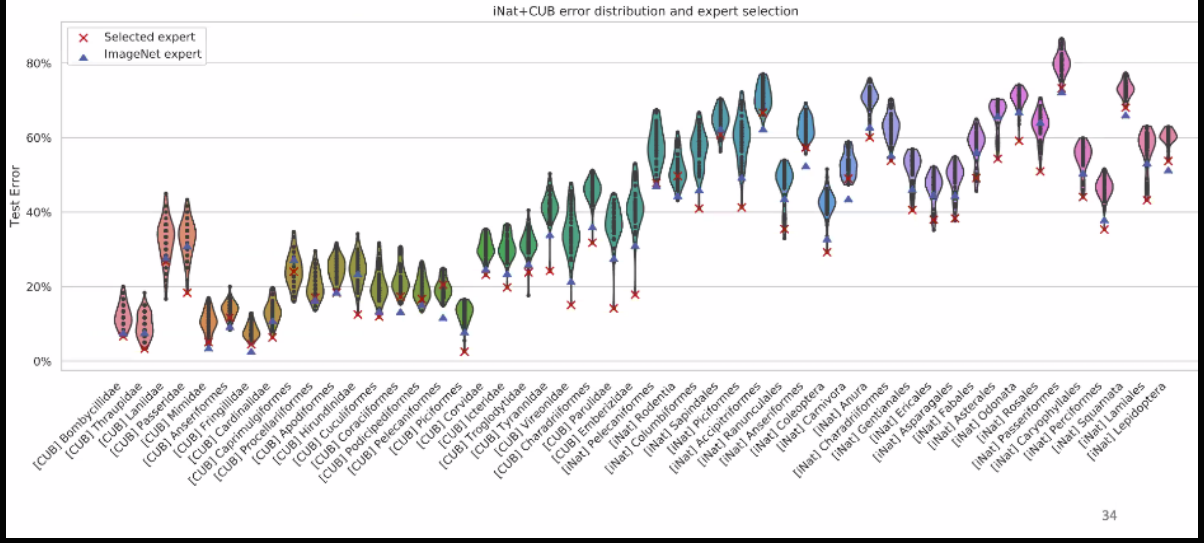
Model Zoo [156 experts]

- ResNet-34 pertained on ImageNet
- Followed by fine-tuning on tasks with enough examples

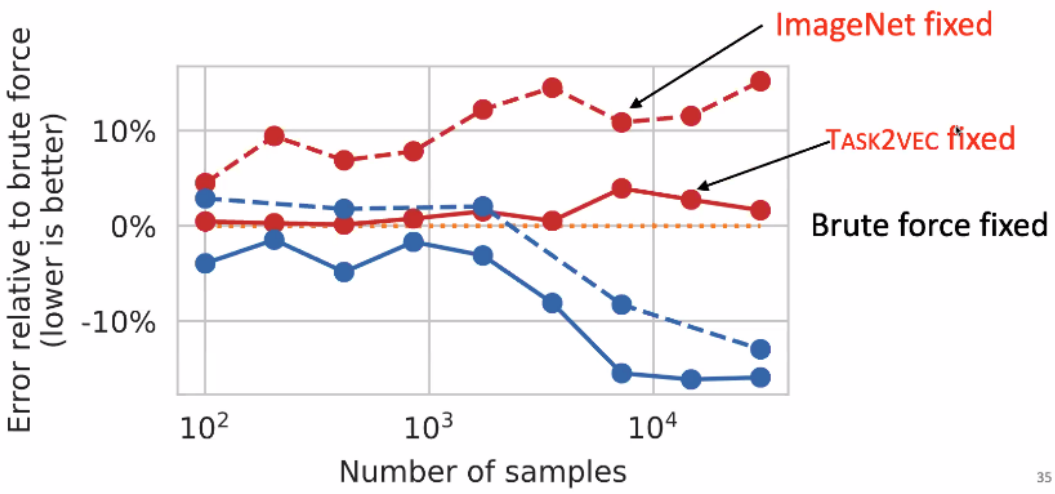




ImageNet expert is *usually good*, but on many tasks the best expert handily *outperforms* the ImageNet expert



Data efficiency of TASK2VEC

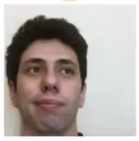


Talk outline

aws

I. Task2Vec: vector representations of tasks [ICCV'19]

- Captures task similarity, task difficulty, etc.
- Useful for model recommendation



Alessandro Achille

II. When does self-supervision improve few-shot learning?[CVPR'20]

III. A realistic-evaluation of semi-supervised learning [CVPR'21]

- Self/semi—supervision is effective as long as the unlabeled images are from a similar domain
- Requires a notion of domain similarity

Jong-Chyi Su

Zezhou Cheng

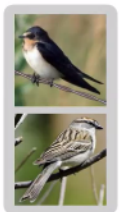
Few-shot learning

Generalization to novel classes

- **Base classes:** many labeled examples
- **Novel classes:** few labeled examples
- **Base** and **novel classes** are disjoint

Train on
Base classes

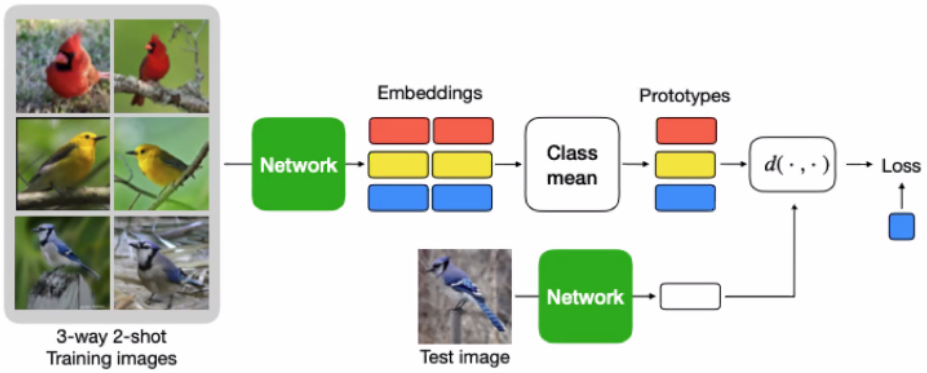
Test on
Novel classes



Many approaches based on meta-learning and metric learning

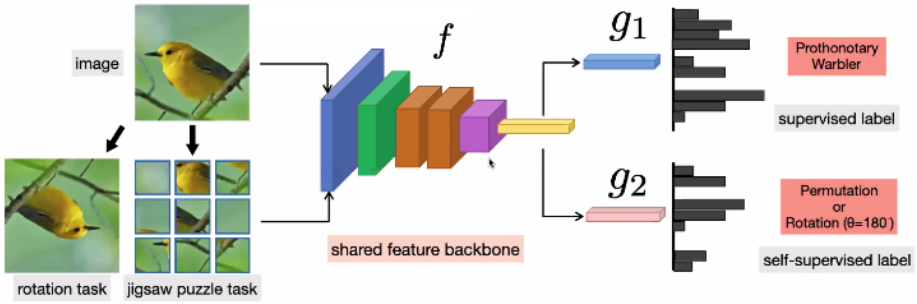
Prototypical networks [Snell et al., NeurIPS'17]

- Meta-learning with a nearest-mean classifier
- Sample tasks from the **base set**, minimize **error** on novel images



Self-supervision for few-shot learning

Jointly train the feature backbone with the few-shot and SSL objective



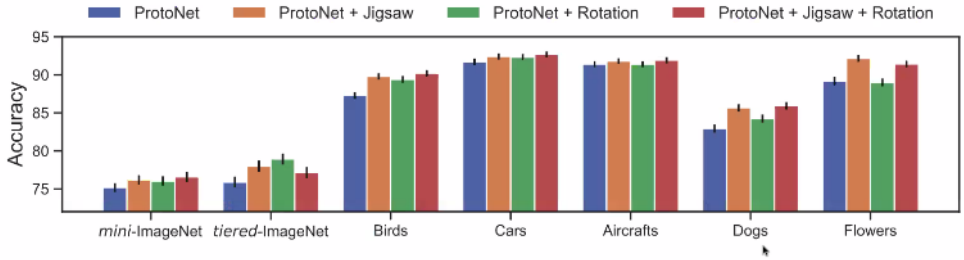
$$\min_{f,g_1,g_2} \mathbb{E}_{(x,y)\sim D_s, x'\sim D_{ss}} [L_s(g_1 \circ f(x), y) + L_{ss}(g_2 \circ f(x'), y')]$$

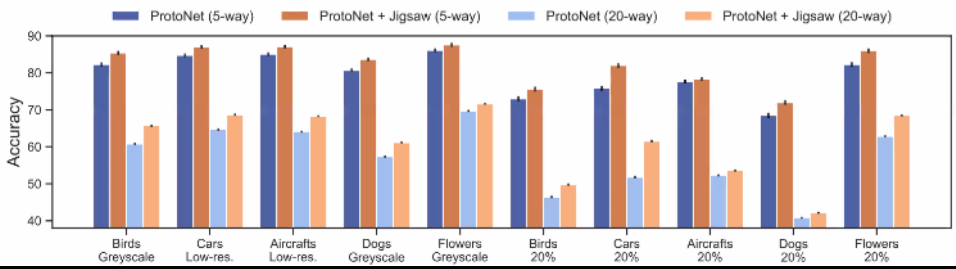
supervised domain — self-supervised domain

Also see: Gidaris et al., ICCV 2019

No extra data!

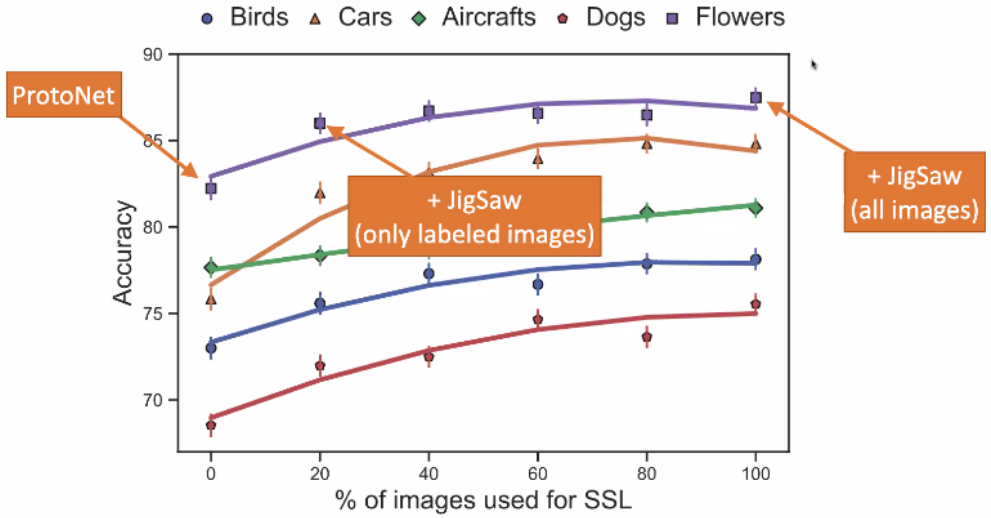
SSL helps even in the few-shot regime $D_s = D_{ss}$





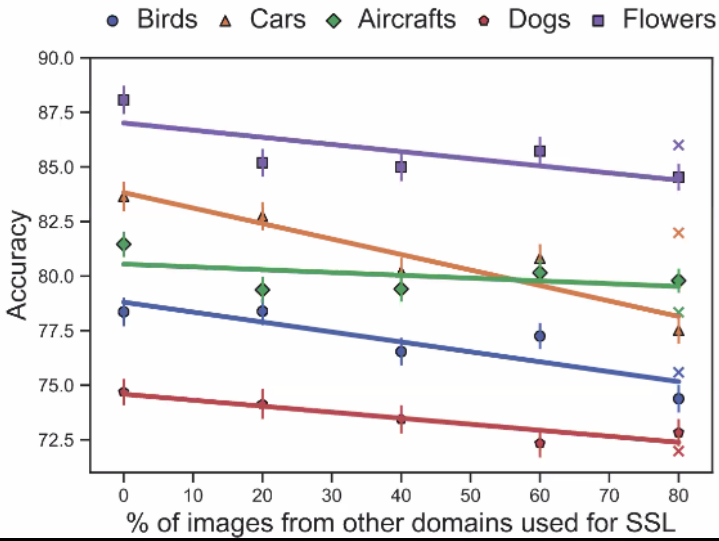
extra data

SSL helps even in the few-shot regime $D_s = D_{ss}$



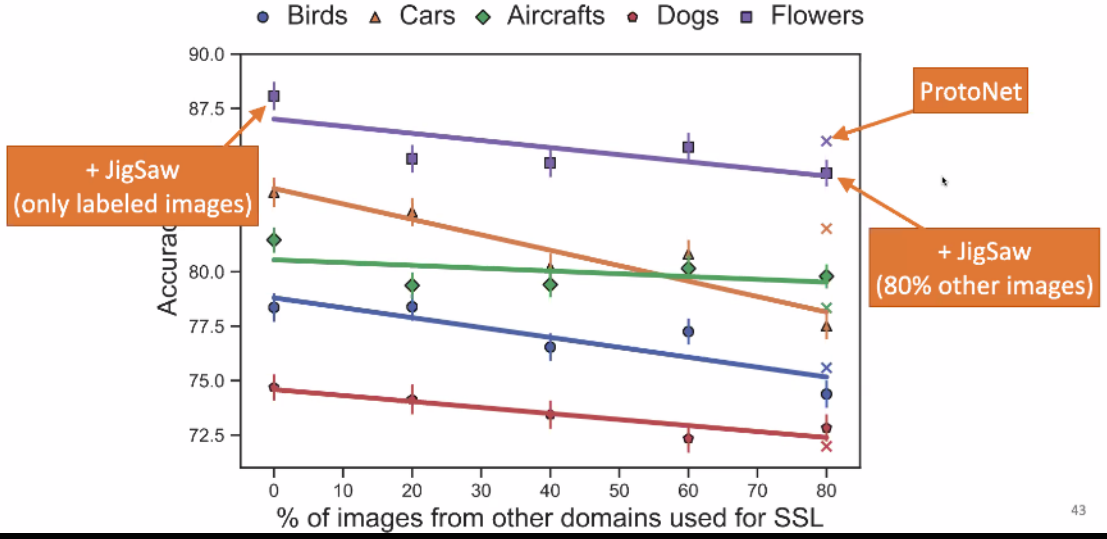
extra data

Domain shifts have a negative impact $D_s \neq D_{ss}$



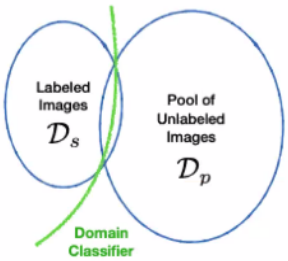
extra data

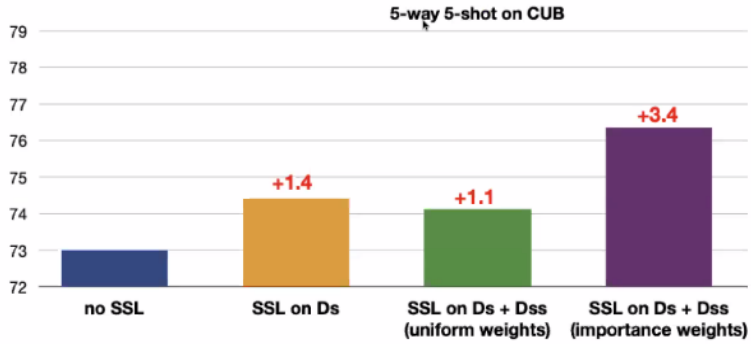
Domain shifts have a negative impact $D_s \neq D_{ss}$



Selecting images for self-supervision

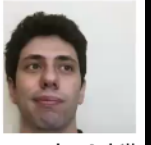
- **Labeled images (D_s):** 20% CUB dataset
- **Unlabeled images (D_p):** OpenImages + iNaturalist (> 2M images)
- Select unlabeled images **(D_{ss})** for self-supervision





Talk outline

aws



Alessandro Achille

I. Task2Vec: Vector representations of tasks [ICCV'19]

- Captures task similarity, task difficulty, etc.
- Useful for model recommendation

II. When does self-supervision improve few-shot learning?[CVPR'20]

III. A realistic-evaluation of semi-supervised learning [CVPR'21]

- Self/semi—supervision is effective as long as the unlabeled images are from a similar domain
- Requires a notion of domain similarity

Jong-Chyi Su

Zezhou Cheng

Benchmarking semi-supervised learning

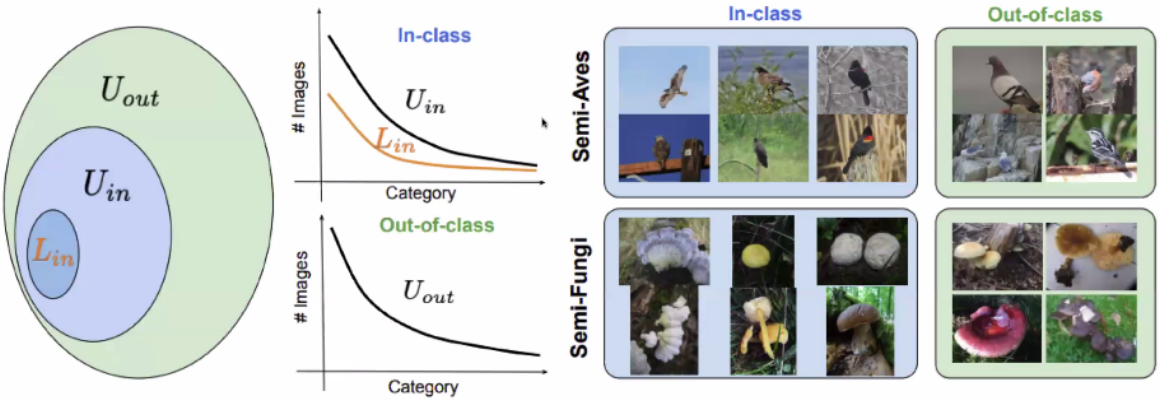
A problem with existing benchmarks

- Curated datasets: CIFAR, MNIST, SVHN, ImageNet
- Uniform class distribution
- Unlabeled data does not contain novel classes
- Role of transfer learning?

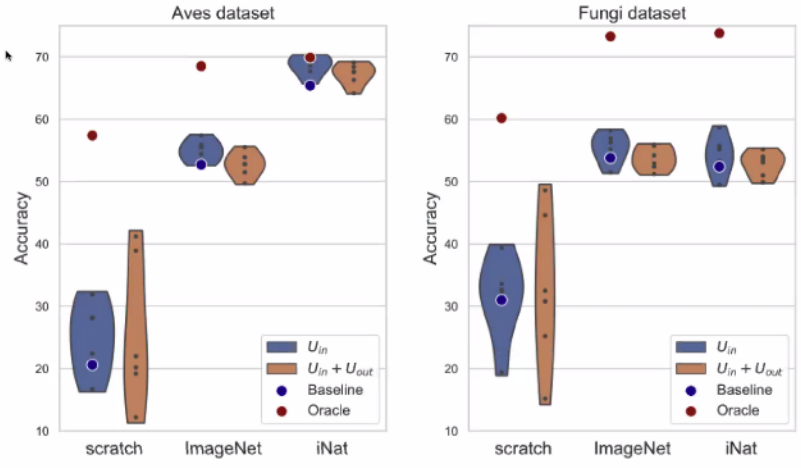
Motivated the FGVC7 and FGVC8 semi-supervised learning challenges

A realistic benchmark

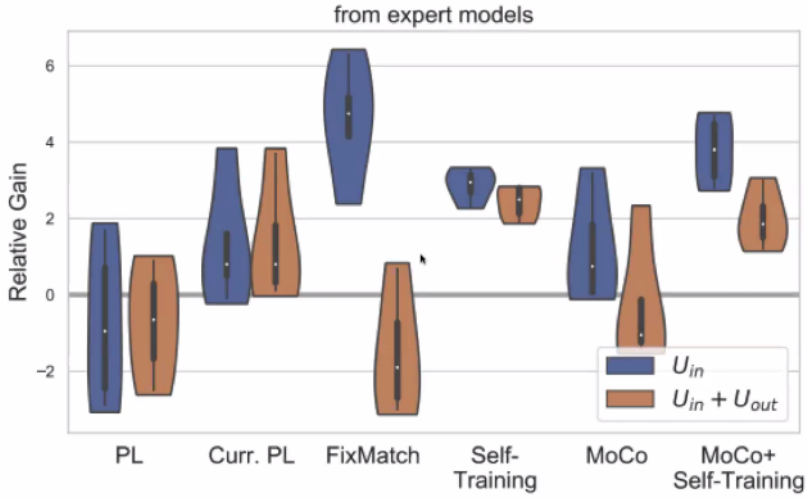
Semi-Aves and Semi-Fungi datasets (CVPR'20)



Takeaway #1: Transfer learning is effective, levels the playing field for semi-supervised learning



Takeaway #2: Methods are not robust to domain shifts (e.g., presence of novel classes)



Thank you!

I. **Task2Vec:** Vector representations of tasks [ICCV'19]

II. **When does self-supervision improve few-shot learning?**[CVPR'20]

III. **A realistic-evaluation of semi-supervised learning** [CVPR'21]

+ Many others

aws NSF Adobe UMassAmherst College of Information and Computer Sciences NVIDIA