CS7150 Deep Learning

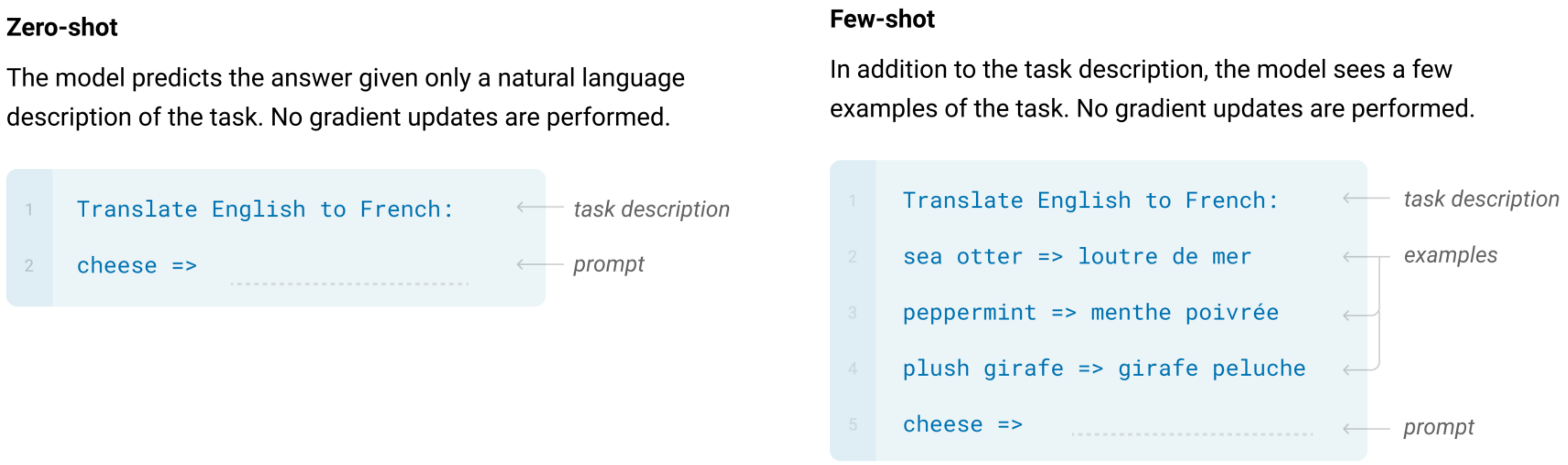
Jiaji Huang

https://jiaji-huang.github.io

03/09/2024

Recap of Last Lecture

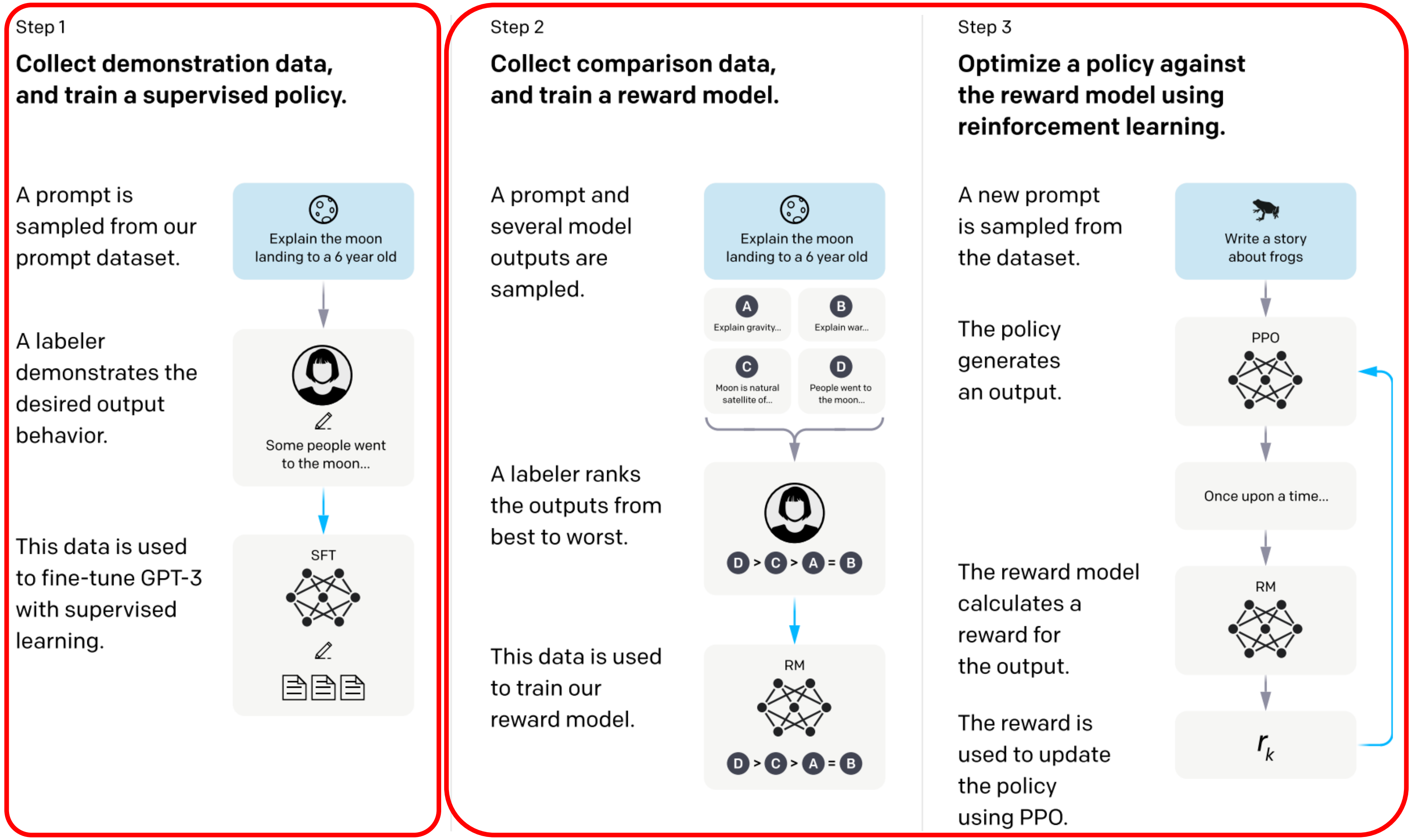
- Parameter Efficient FineTuning (PEFT)
- Pretrained LMs already solve new Tasks to some extent
  - Prompt engineering and zero/few-shot In-context Learning



Recap of Last Lecture

Yet finetuning is still necessary!

Supervised finetuning (SFT)

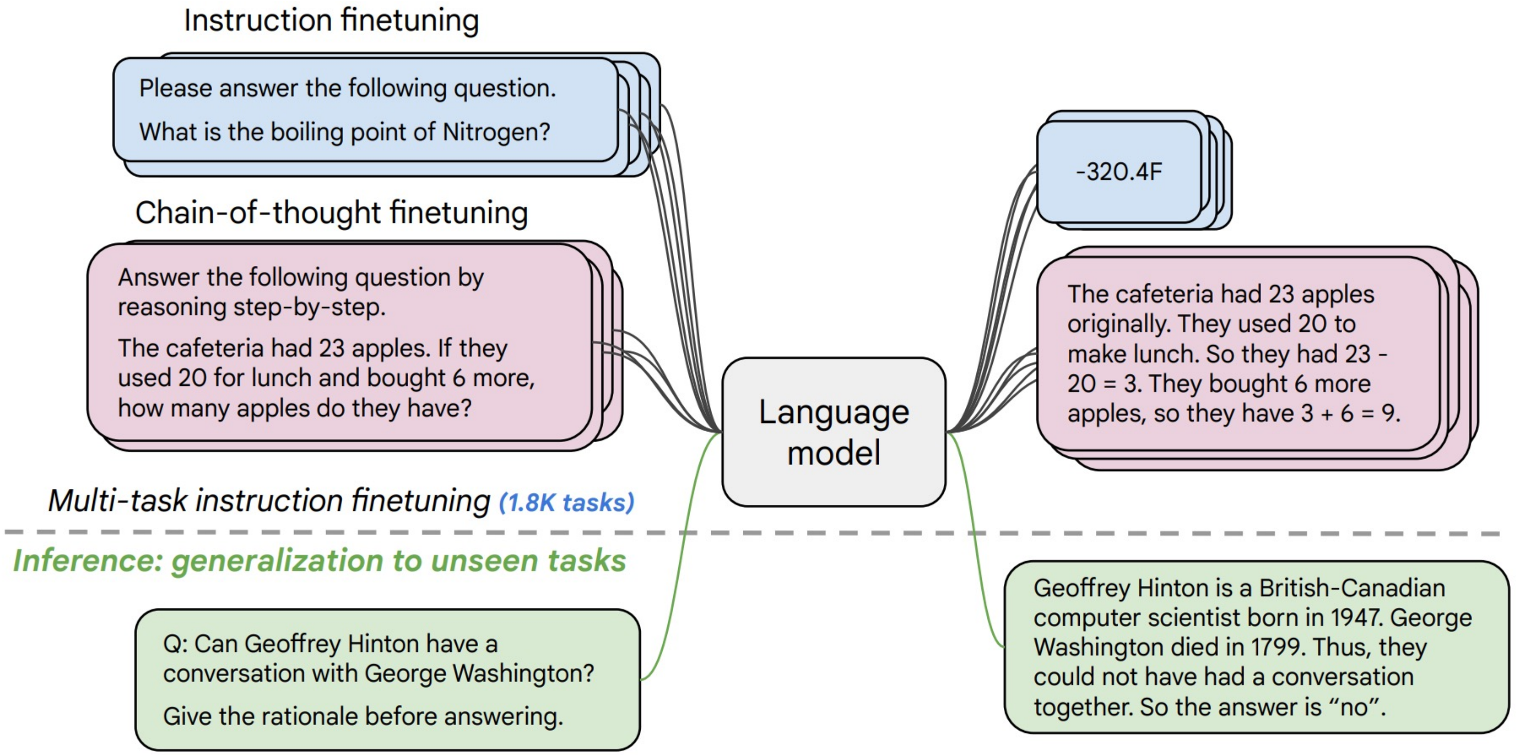


RLHF

Figure from [Ouyang et. al, 2022]

Recap of Last Lecture: on SFT

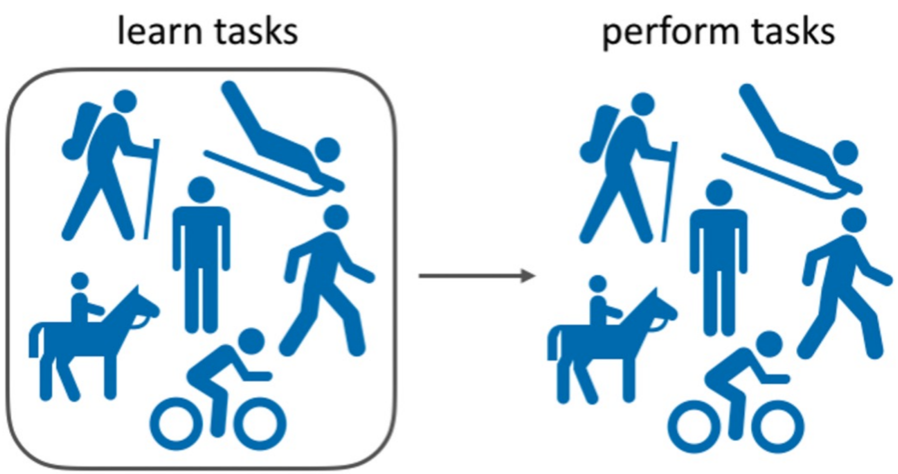
- Instruction finetuning and FLAN (multi-task training objective)
- Seeing many tasks helps for solving a new task (meta Learning)



[Chung et. al, 2022]()

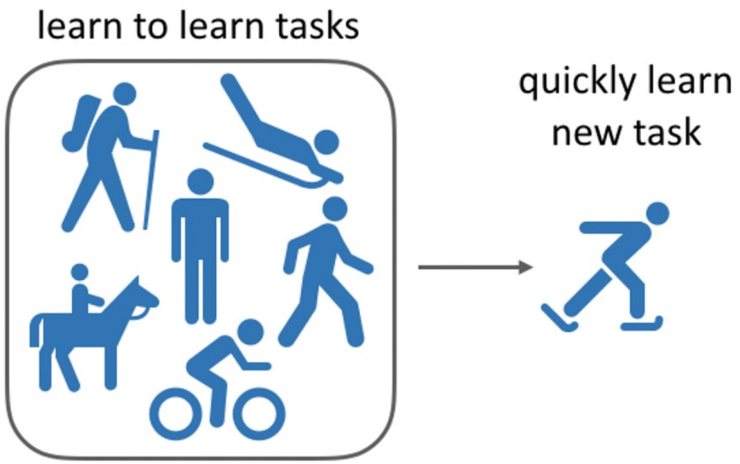
Multi-task Learning vs Meta Learning

- Multi-task Learning



- Setting: Test tasks = Training tasks
- Goal: master this set of tasks

- Meta Learning



- Setting: Test task(s) $\notin$ Training task
- Goal: Adapt to unseen task(s) quickly

Slide adapted from this [talk]

Agenda for Today

- Multi-task Learning (MTL)
- Meta Learning
- Zero-shot Learning

Formalize: Defining Tasks

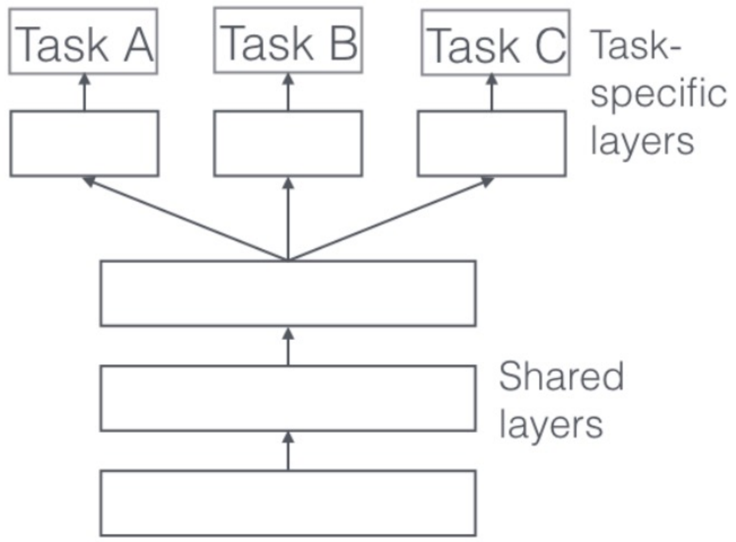
- A task has
  - Input $\boldsymbol{x} \sim p(\boldsymbol{x})$
  - Target output $\boldsymbol{y}$ given $\boldsymbol{x}$, draw from $p(\boldsymbol{y}|\boldsymbol{x})$
  - $\mathcal{T} \triangleq (p(\boldsymbol{x}),\ p(\boldsymbol{y}|\boldsymbol{x}))$
- Example: Different $p(\boldsymbol{x})$
  - Scene image classification v.s. medical image classification
- Example: Same $p(\boldsymbol{x})$ but different $p(\boldsymbol{y}|\boldsymbol{x})$
  - Scene classification: $\boldsymbol{x}$ scene images, $\boldsymbol{y}$ scene label
  - Object detection from Scene image: $\boldsymbol{y}$ object bounding box

Formalize: Multi-Task Learning (MTL)

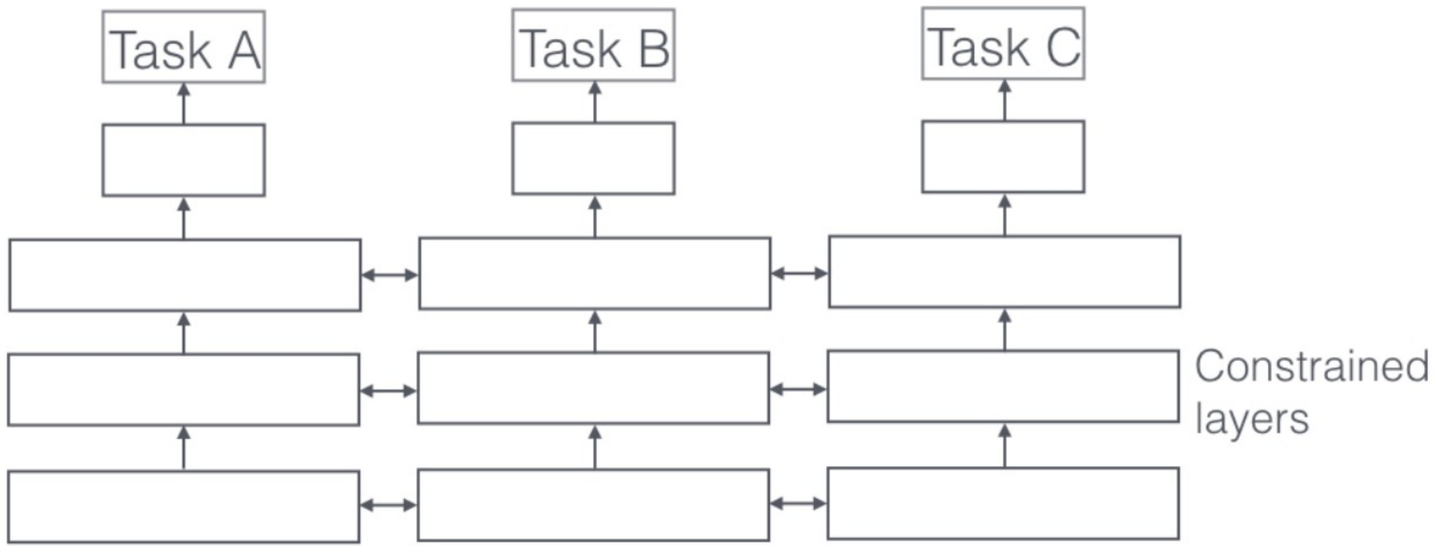
- $\mathcal{T}_i \triangleq (p_i(\boldsymbol{x}),\ p_i(\boldsymbol{y}|\boldsymbol{x})), i = 1, \dots, T$
- Training data $\mathcal{D}_i^{tr}$, testing data $\mathcal{D}_i^{te}$ draw from each $\mathcal{T}_i$
- Train on $\mathcal{D}_i^{tr}$ $(i = 1, \dots, T)$ and test on each $\mathcal{D}_i^{te}$
- Assumption: the tasks are relevant
- Otherwise, we may just train a model for each task

Sharing Model Parameters for MTL

- Hard sharing



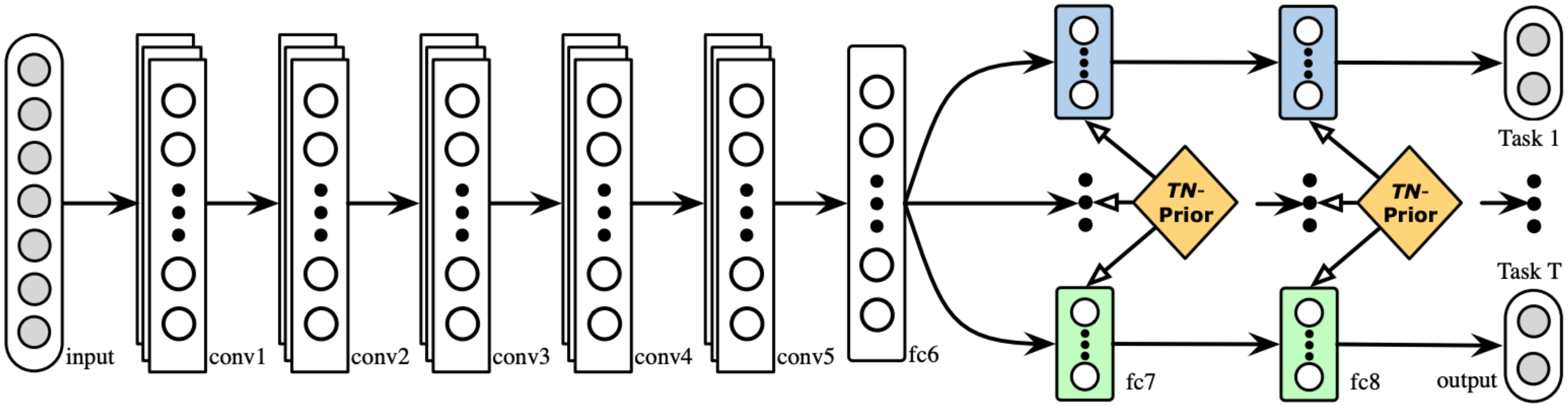
- Soft sharing



[Ruder 2017]

Example of Hard Parameter Sharing

- Deep Relation Network (Long and Wang, 2015)

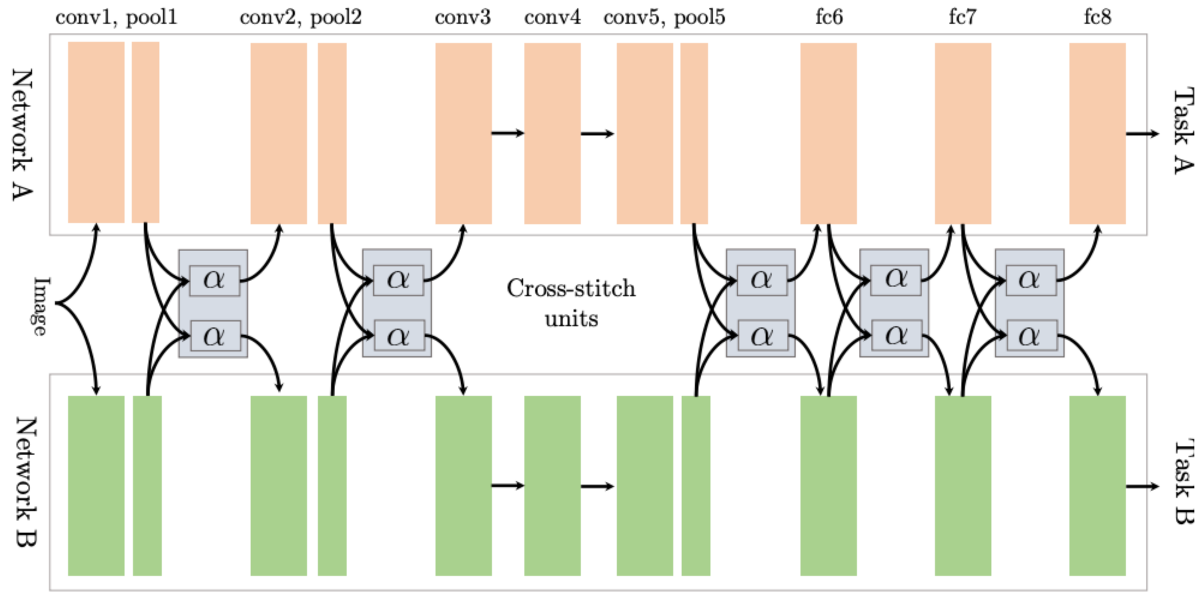


- Share conv layers
- Prior on Fc7, fc8’s weight matrices: encodes task relationship

Example of Soft Parameter Sharing

- Cross-stitch Network (Misra et. al, 2016)
- Start from two networks (same architecture) for two tasks
- Learn linear combinator $\boldsymbol{\alpha}$ for feature maps

$$\begin{bmatrix} \tilde{x}_{\mathrm{A}}^{ij} \\ \tilde{x}_{\mathrm{B}}^{ij} \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_{\mathrm{AA}} & \alpha_{\mathrm{AB}} \\ \alpha_{BA} & \alpha_{\mathrm{BB}} \end{bmatrix}}_{\text{Encodes our knowledge of task relevance}} \begin{bmatrix} x_{\mathrm{A}}^{ij} \\ x_{\mathrm{B}}^{ij} \end{bmatrix}$$



What Parameters/Layers to be Shared

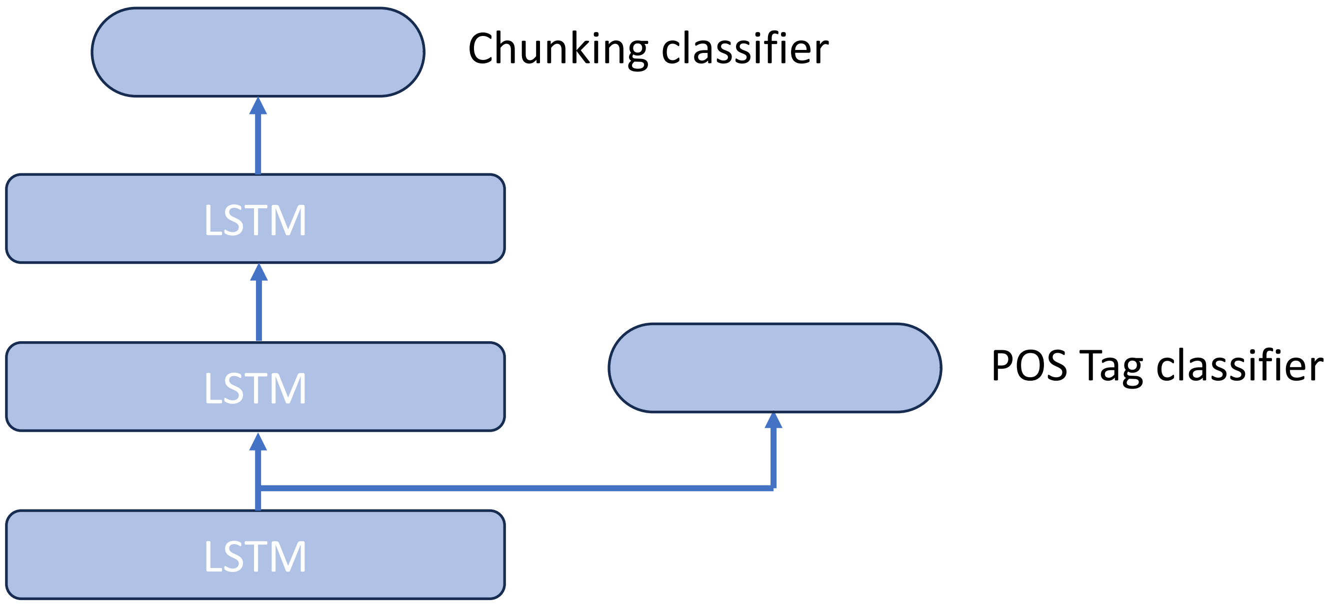
- Common to share the bottom layers, with task-specific "head"
- Sometimes a task is more fundamental than the others (Søgaard and Goldberg, 2016)
- E.g., Chunking works on top of POS (part of speech) tags

POS Tag	Abbr.	words
Determiner	DT	a, an, the, this
Adjective	JJ	big, kind, cool, ...
Noun	NN	dog, cat
preposition	IN	at, Into, over, ...
Verb (past tense)	VBD	walked, talked, ...

Example from medium

Jointly learn chunking with POS tagging

- Discussion: how do we share the parameters/layers?



Jointly learn chunking with POS tagging

- Results

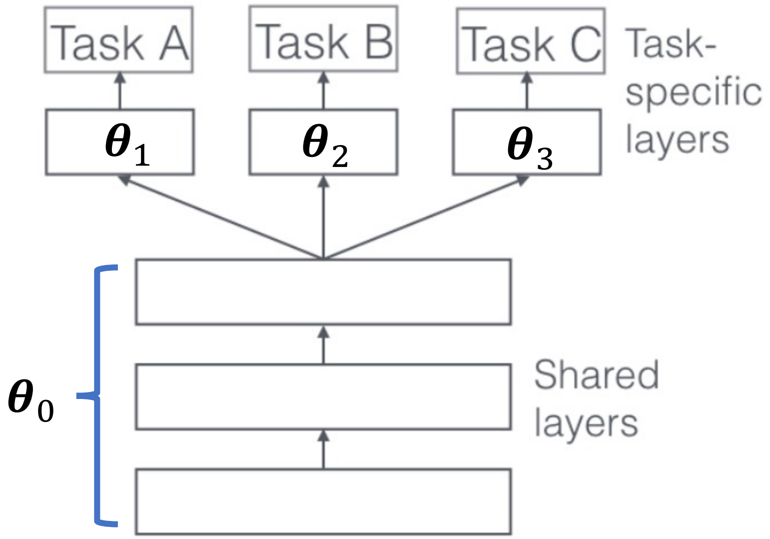
	Layers		Domains			
	Chunks	POS	Broadcast (6)	Bc-news (8)	Magazines (1)	Weblogs (6)
bi-LSTM	3	-	88.98	91.84	90.09	90.36
	3	3	88.91	91.84	90.95	90.43
	3	1	**89.48**	**92.03**	**91.53**	**90.78**

- More helpful to use low-level task at lower layer

MTL Objective functions

- $\boldsymbol{\theta}_0$: shared parameter, $\boldsymbol{\theta}_i$: task-specific parameter
- Commonly seen, additive

$$\min_{\boldsymbol{\theta}_0,\dots,\boldsymbol{\theta}_T} \sum_{i=1}^{T} w_i \left\{ \mathcal{L}_i \triangleq \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_i^{tr}} \ell_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i; \boldsymbol{x}, \boldsymbol{y}) \right\}$$



- w_i: importance of the i-th task
- w_i such that tasks with similar gradient magnitude (Chen et. al, 2018)

Optimize the Objective

$$\min_{\boldsymbol{\theta}_0,\dots,\boldsymbol{\theta}_T} \sum_{i=1}^{T} w_i \left\{ \mathcal{L}_i \triangleq \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_i^{tr}} \ell_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i; \boldsymbol{x}, \boldsymbol{y}) \right\}$$

- Sample a minibatch of tasks, indices $\mathcal{I} \subseteq \{1, \dots, T\}$
- For each task $i \in \mathcal{I}$, sample a batch of $(\boldsymbol{x}, \boldsymbol{y})$'s, denoted as $\mathcal{X}_i \subseteq \mathcal{D}_i^{tr}$
- Compute (stochastic) loss

$$\hat{\mathcal{L}} = \sum_{i\in\mathcal{I}} w_i \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{X}_i} \ell_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i; \boldsymbol{x}, \boldsymbol{y})$$

- Back-prop to compute gradients, $\frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\theta}_0}$ and $\frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\theta}_i}$ $(i \in \mathcal{I})$
- Update the $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_i$'s with Adam, etc.

Slide adapted from [CS330]

Potential Issues

- Choice of w_i can be tricky
- Tasks may compete (negative transfer), i.e.,

$$\mathcal{L}_1(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) < \mathcal{L}_1(\overline{\boldsymbol{\theta}_0}, \overline{\boldsymbol{\theta}_1})$$

but

$$\mathcal{L}_2(\boldsymbol{\theta}_0, \boldsymbol{\theta}_2) > \mathcal{L}_2(\overline{\boldsymbol{\theta}_0}, \overline{\boldsymbol{\theta}_2})$$

Improve for task 1, but harm task 2

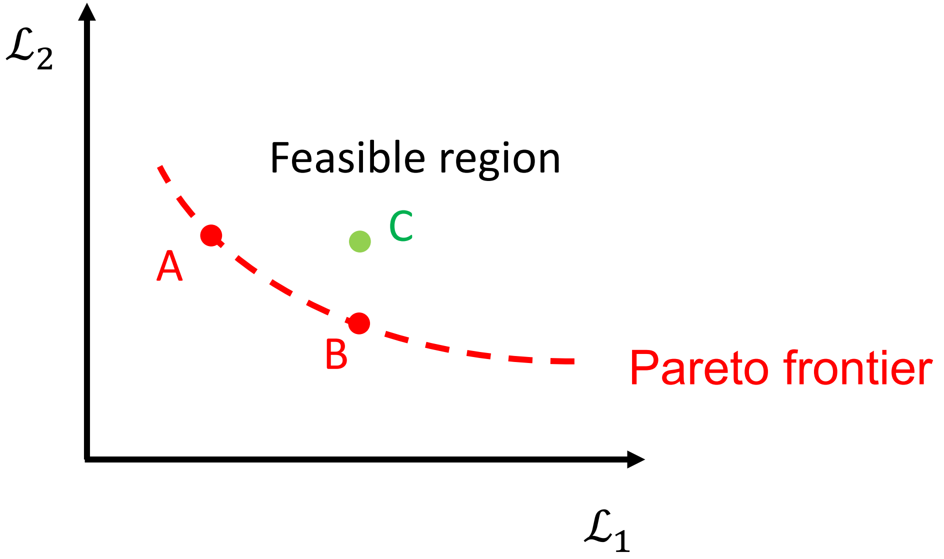
Multi-objective MTL (Sener and Koltun, 2018)

$$\min_{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T} \{\mathcal{L}_1(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1), \ldots, \mathcal{L}_T(\boldsymbol{\theta}_0, \boldsymbol{\theta}_T)\}$$

- Pareto optimality:

$(\boldsymbol{\theta}_0^*, \boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_T^*)$ is Pareto optimal if any other $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T)$ harms at least one task

- Optimize so we arrive onto the Pareto frontier



Optimizer for the Multi-objective MTL

Pareto stationary point $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T)$ satisfies (KKT condition):

- For task-specific parameters:

$$\nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i) = 0 \text{ for all task } i = 1, \ldots, m$$

- For shared parameters:

Exist $w_1, \ldots, w_T \geq 0$ where $\sum_{i=1}^{T} w_i = 1$ such that

$$\sum_{i=1}^{T} w_i \nabla_{\boldsymbol{\theta}_0} \mathcal{L}_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i) = 0$$

Optimizer for the Multi-objective MTL

While not converged:

Update task specific $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \eta_i \nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i)$

Solve for $w_1, \ldots, w_T \geq 0$ where $\sum_{i=1}^{T} w_i = 1$ such that

$$\min_{w_1, \ldots, w_T} \left\| \sum_{i=1}^{T} w_i \nabla_{\boldsymbol{\theta}_0} \mathcal{L}_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i) \right\|^2$$

Update $\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_0 - \eta \sum_{i=1}^{T} w_i \nabla_{\boldsymbol{\theta}_0} \mathcal{L}_i(\boldsymbol{\theta}_0, \boldsymbol{\theta}_i)$

- Note: we can convert the above to stochastic gradient descent

Results

- Experiment on multiMNIST dataset

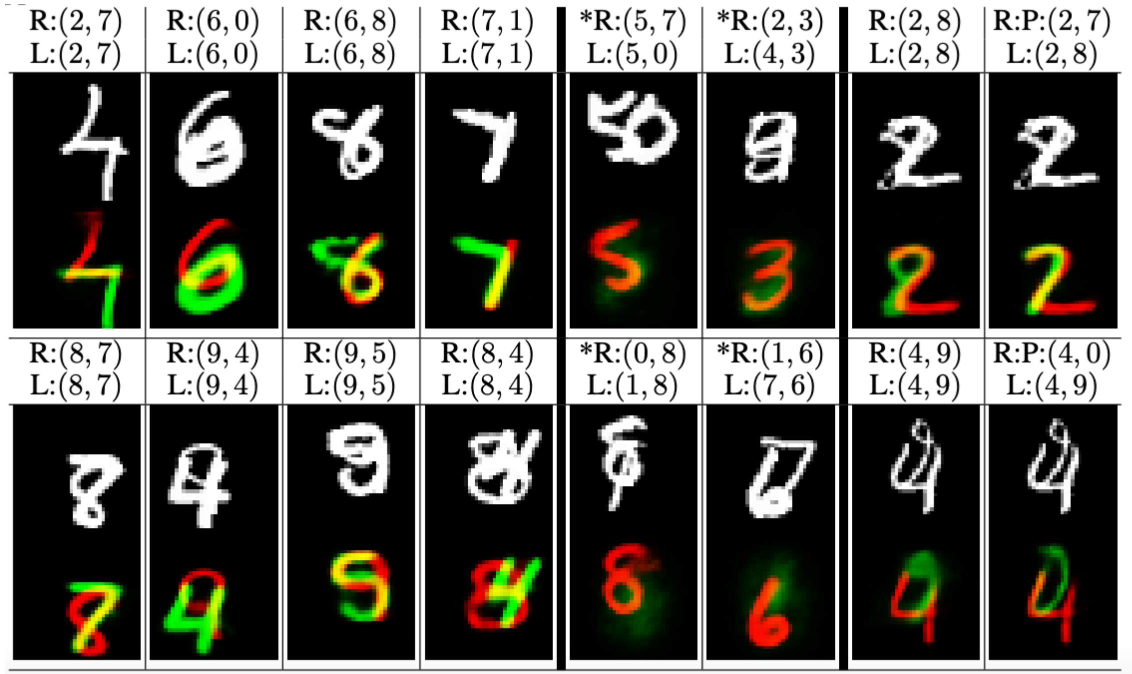


Fig. 3 from (Sener and Koltun, 2018)

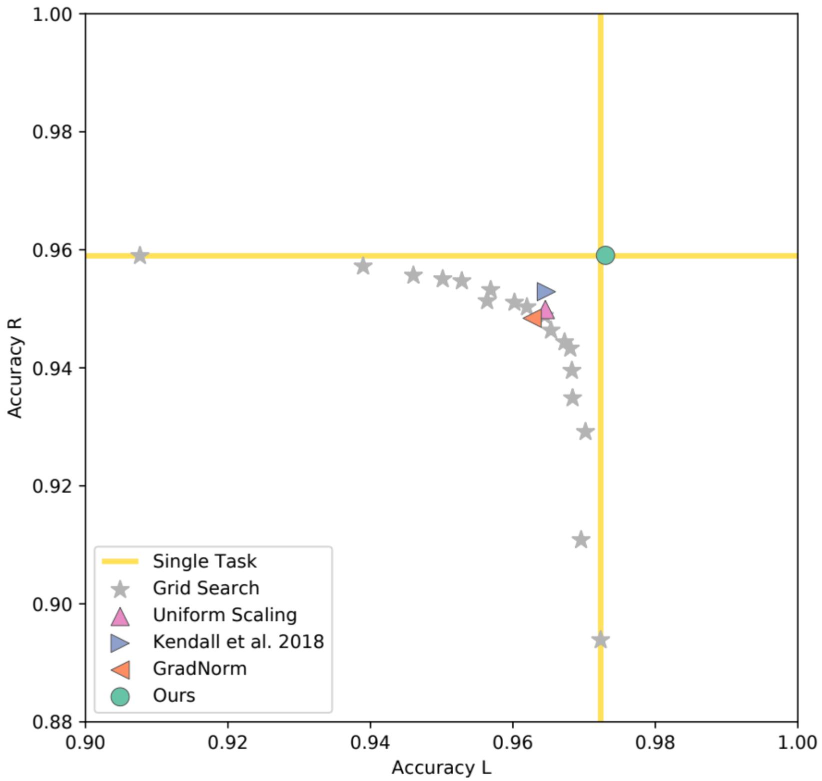


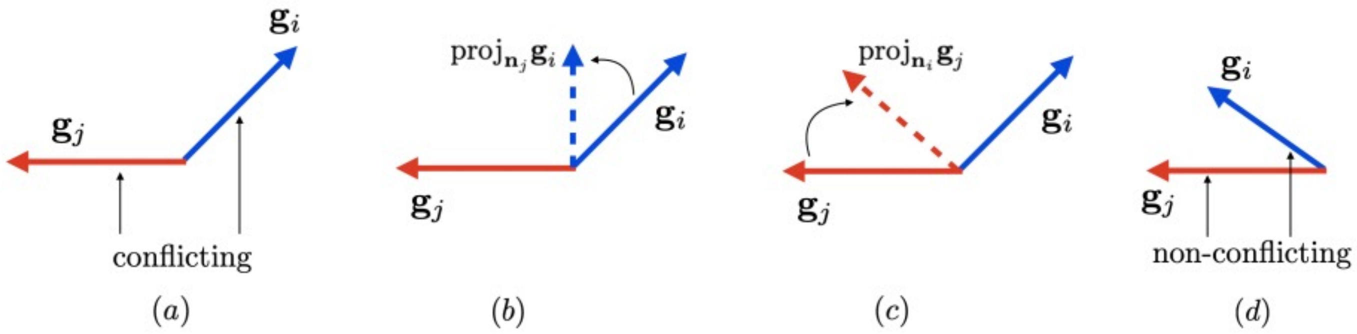
Figure 3: **MultiMNIST accuracy profile.** We plot the obtained accuracy in detecting the left and right digits for all baselines. The grid-search results suggest that the tasks compete for model capacity. Our method is the only one that finds a solution that is as good as training a dedicated model for each task. Top-right is better.

Diagnose Negative Transfer via Gradients

- Again Consider additive objective

$$\min_{\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_T} \sum_{i=1}^{T} w_i \mathcal{L}_i$$

- Remove conflicting components (Yu, et. al, 2022)



Results

	% accuracy
task specific, 1-fc [46]	42
task specific, all-fc [46]	49
cross stitch, all-fc [40]	53
independent	67.7
PCGrad (proposed)	71

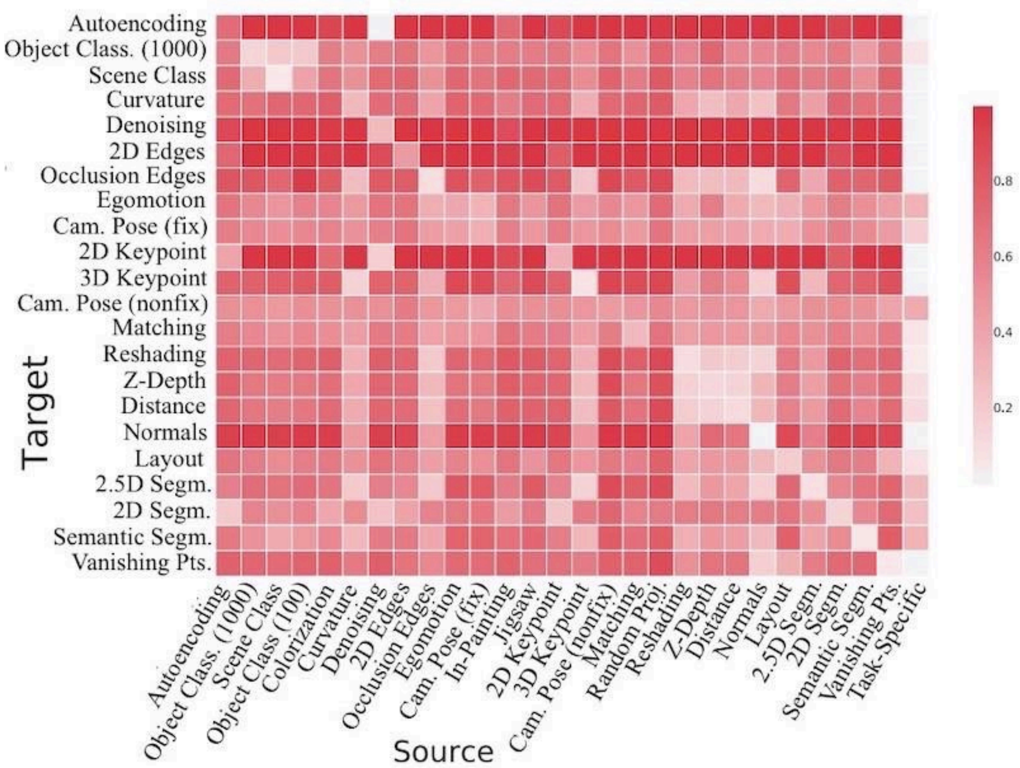
Naïve MTL inferior to independently trained

- But can we predict if two tasks are relevant?

Yu, et. al, 2022

On Task Relevance

- [Taskonomy](#) by Stanford
- Measured as performance of transfer learning



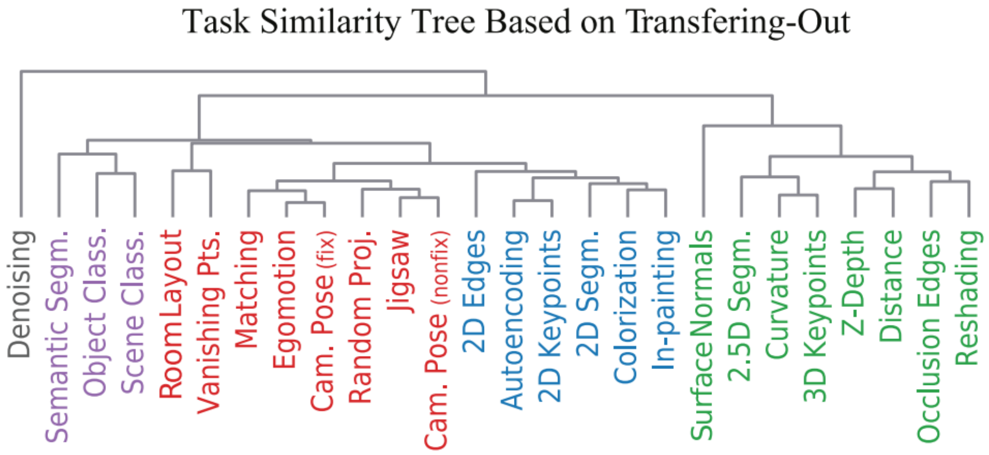
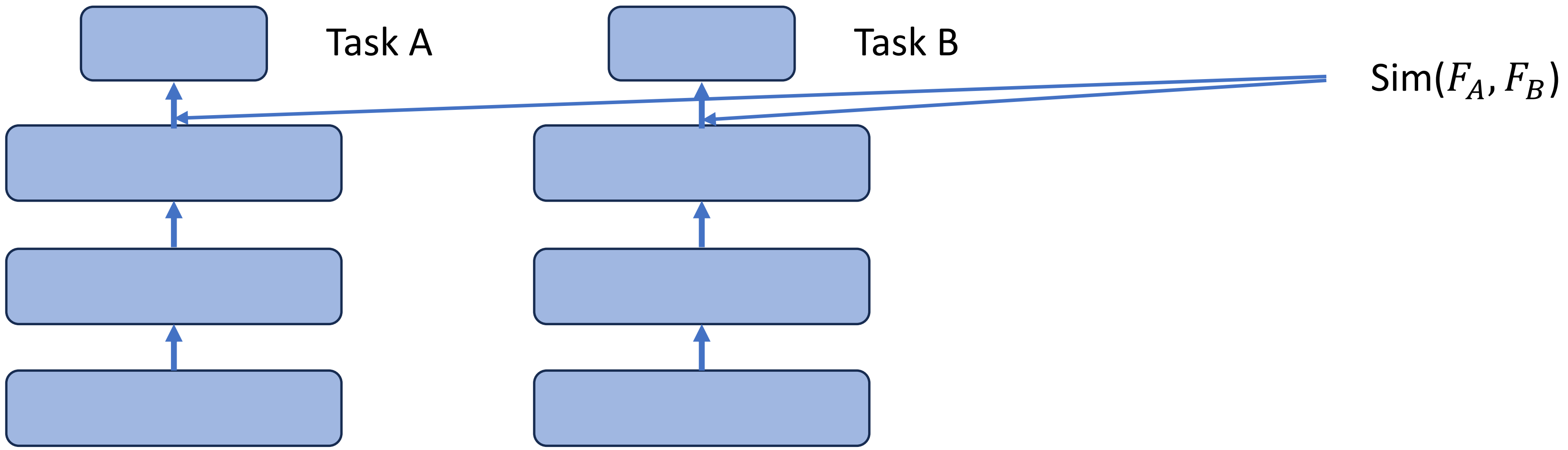


Figure 13: **Task Similarity Tree.** Agglomerative clustering of tasks based on their transferring-out patterns (i.e. using columns of normalized affinity matrix as task features). 3D, 2D, low dimensional geometric, and semantic tasks clustered together using a fully computational approach.

Task Relevance: More Analytical way

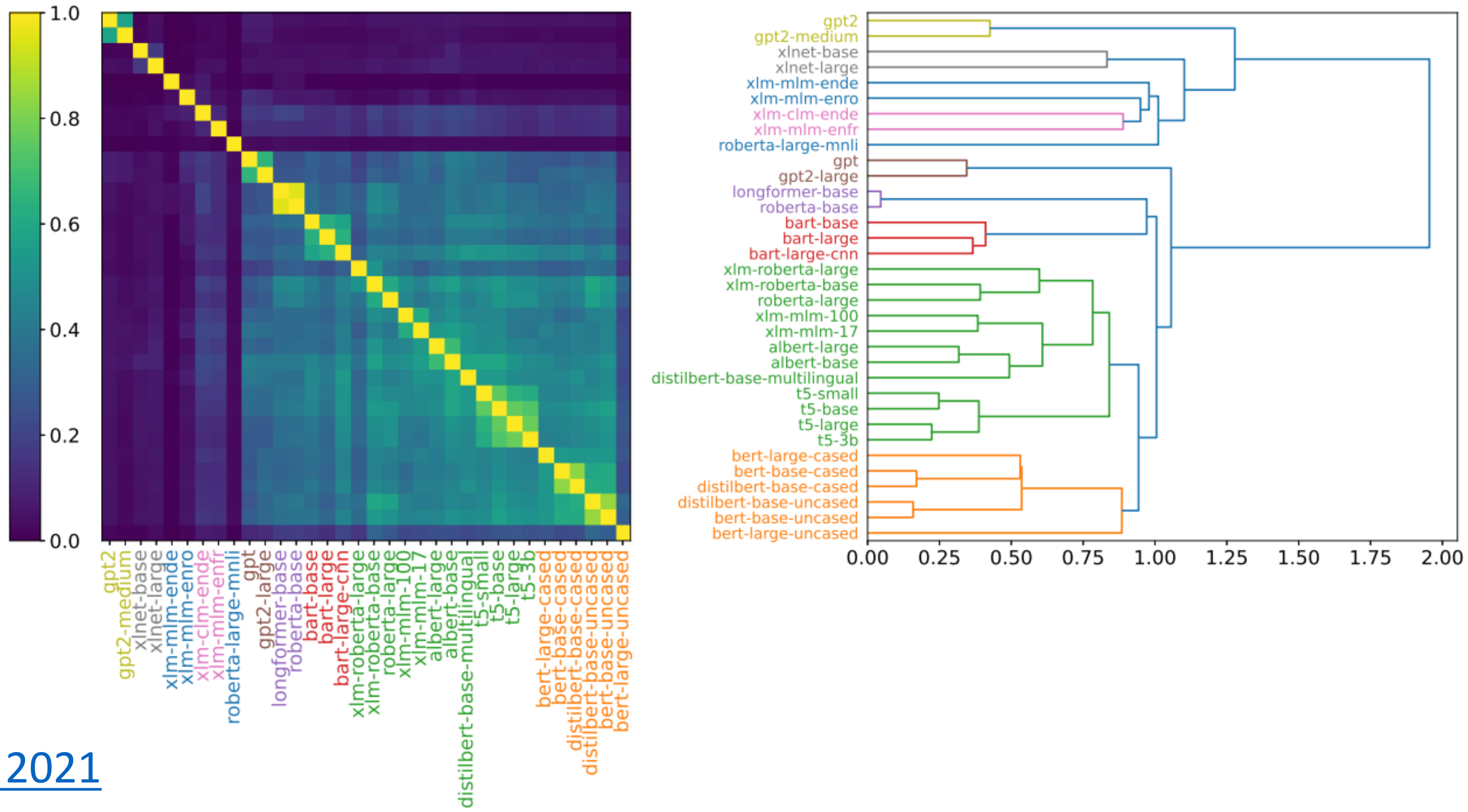
- Consider two tasks with same $p(\boldsymbol{x})$, but different $p(\boldsymbol{y}|\boldsymbol{x})$'s
- Assume we have trained a model for each of the two tasks



- Measure task relevance using features' similarity (Huang et. al, 2021)

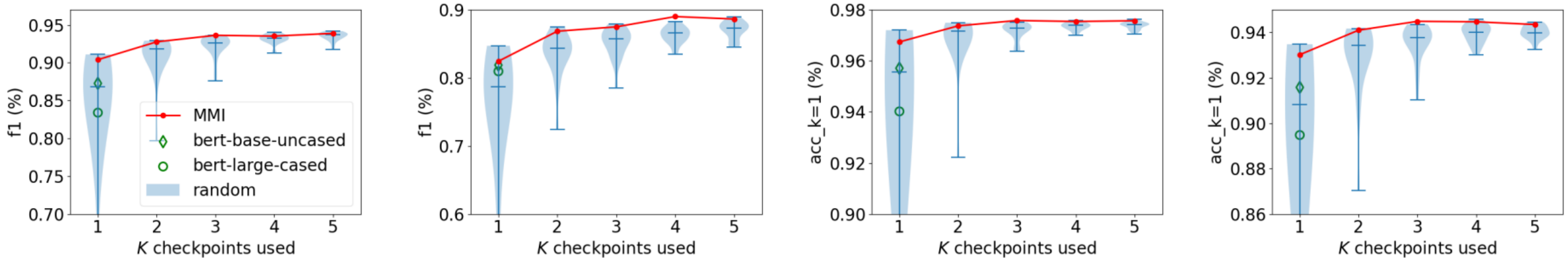
Task Relevance: More Analytical way

- On Sim(F_A, F_B): invariance w.r.t linear transform (revisit in next lecture)

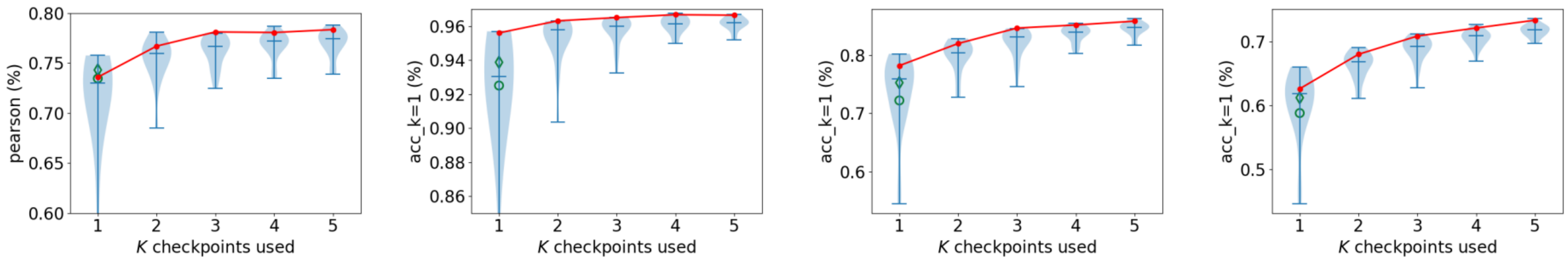


Huang et. al, 2021

Pick checkpoints for new tasks



(a) Chunking (b) NER (c) POS Tagging (d) Semantic tagging



(e) Event factuality (f) Tagging parent in phrase-structure tree (g) Tagging grandparent in phrase-structure tree (h) Tagging great grandparent in phrase-structure tree

Red: selected ckpt ; Blue: randomly picked ckpt

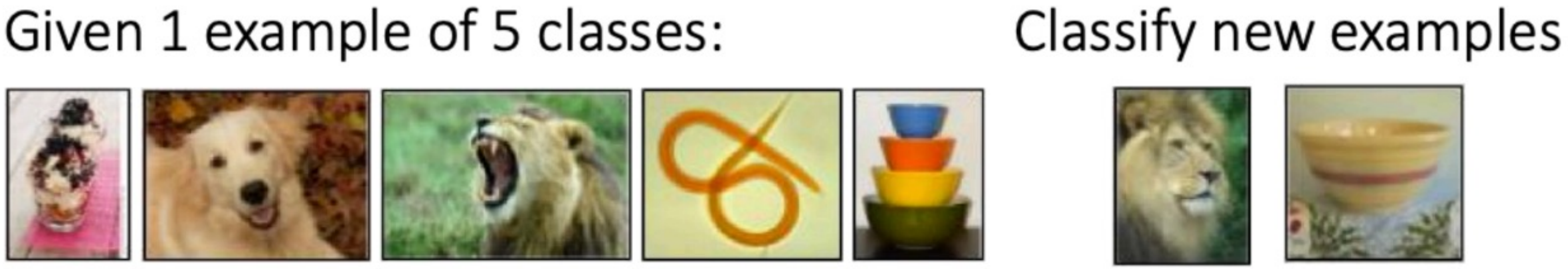
[Huang et. al, 2021](#)

Agenda for Today

- Multi-task Learning (MTL)
- Meta Learning
- Zero-shot Learning

Motivating Meta Learning

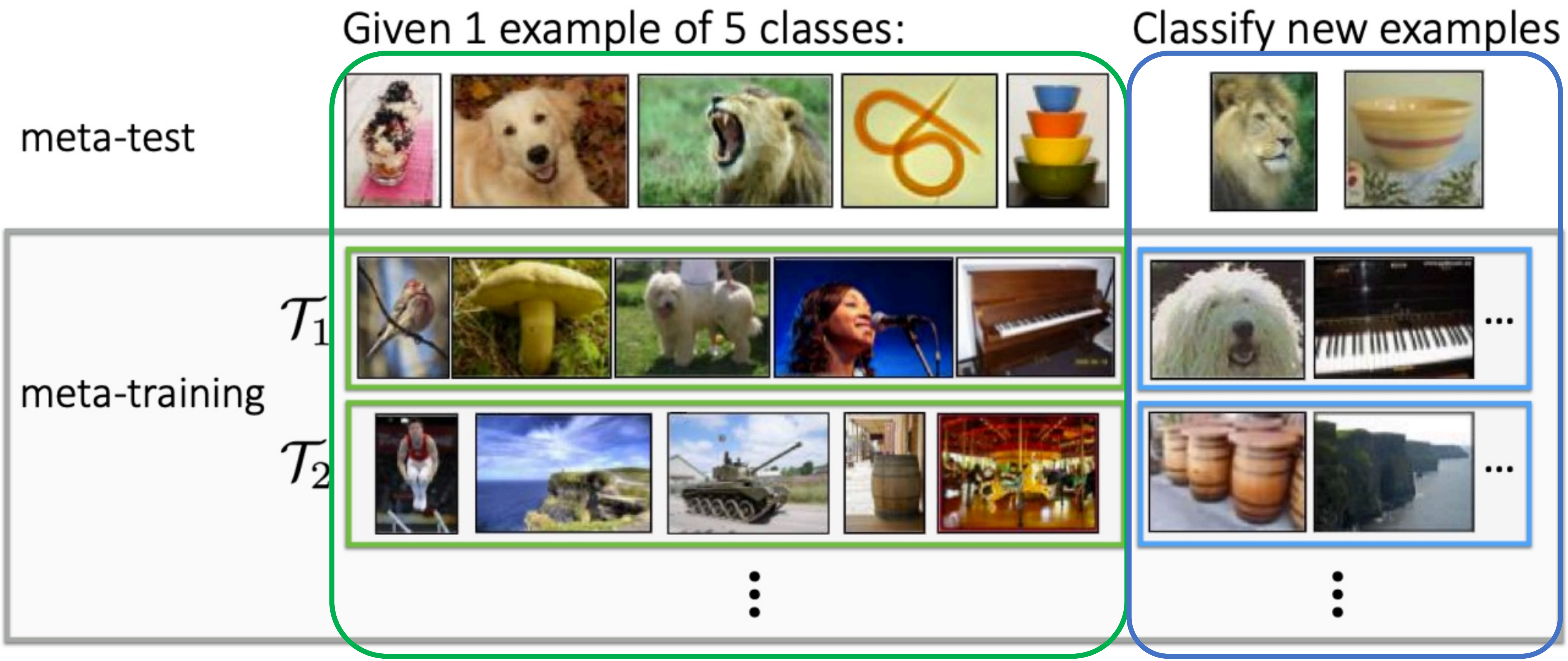
- Sometimes, we may have to learn a model from very few samples
- i.e., few-shot learning
- e.g., 5-way, 1-shot classification



- Seems very hard if we train a randomly initialized network!
- Can we start from a network that is good at few-shot learning?

Illustration from [CS330 slides]

Motivating Meta Learning



Task training sets, “context”, “support set” Task test sets, “query”

Illustration from [CS330 slides]

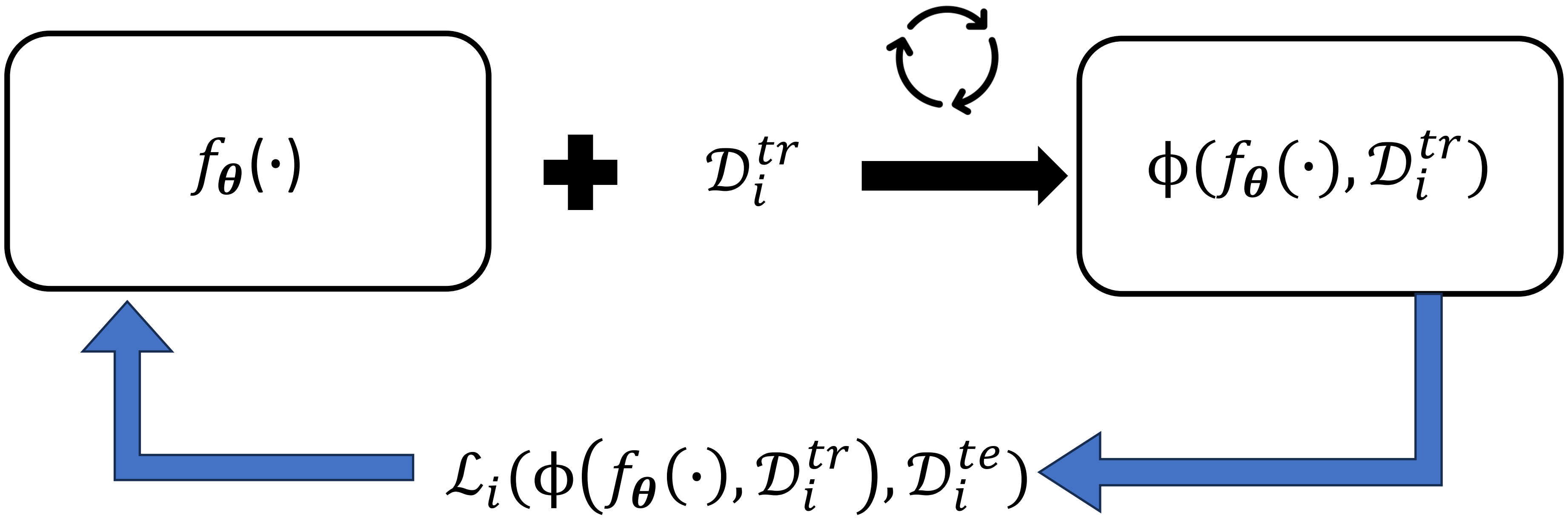
Formalize: Meta Learning

- Meta Training Set
  - Tasks $\mathcal{T}_1, \dots, \mathcal{T}_T$, datasets $\mathcal{D}_1, \dots, \mathcal{D}_T$;
  - Each $\mathcal{D}_i = \mathcal{D}_i^{tr} \cup \mathcal{D}_i^{te}$ (task training and test sets)
- Meta Test Set
  - New task $\mathcal{T}_{T+1}$, training samples $\mathcal{D}_{T+1}^{tr}$, test samples $\mathcal{D}_{T+1}^{te}$
- Objective

Find a network $f_{\boldsymbol{\theta}}(\cdot)$, so that if we few-shot train it on $\mathcal{D}_i^{tr}$, test result on $\mathcal{D}_i^{te}$ is good

Meta Learning: General Framework

1. Few shot Training
2. Get loss on task's test set
3. Back-prop loss to update $\boldsymbol{\theta}$



Meta Learning: General Algorithm

While not converged:

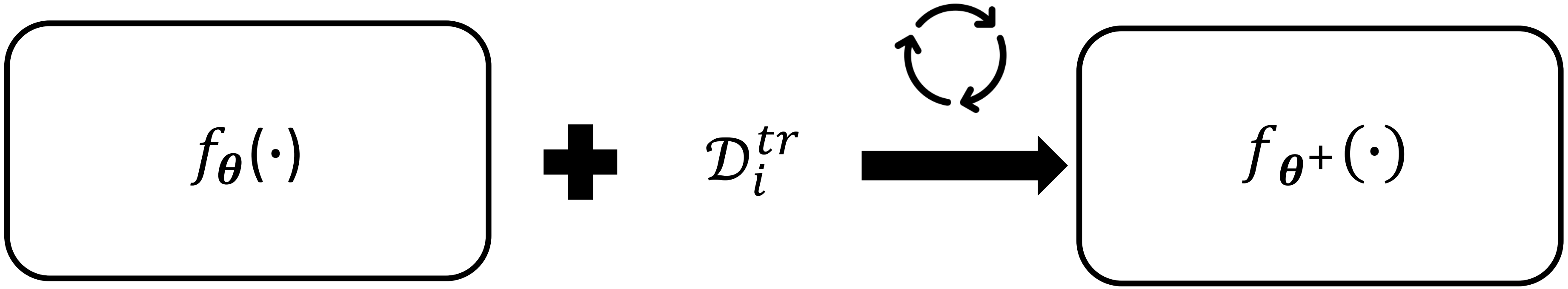
1. Sample task $\mathcal{T}_i$
2. Starting from current network $f_{\boldsymbol{\theta}}(\cdot)$, few-shot train on $\mathcal{D}_i^{tr}$,
   Denote the task-specific model as $\phi(f_{\boldsymbol{\theta}}(\cdot), \mathcal{D}_i^{tr})$
3. Get test loss of $\phi(f_{\boldsymbol{\theta}}(\cdot), \mathcal{D}_i^{tr})$ on $\mathcal{D}_i^{te}$, denoted as
   $$\mathcal{L}_i(\phi(f_{\boldsymbol{\theta}}(\cdot), \mathcal{D}_i^{tr}), \mathcal{D}_i^{te})$$
4. Update $\boldsymbol{\theta}$ via gradient descent

Question: How is it different from transfer learning?

Model Agnostic Meta Learning (MAML)

- $\phi(f_{\boldsymbol{\theta}}(\cdot), \mathcal{D}_i^{tr})$ is simply a gradient step on $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{+} = \boldsymbol{\theta} - \eta \nabla \mathcal{L}_i(\boldsymbol{\theta}; \mathcal{D}_i^{tr})$$



- Evaluate test loss by

$$\mathcal{L}_i(\boldsymbol{\theta}^{+}; \mathcal{D}_{\mathrm{i}}^{te})$$

Optimization for MAML Training Loss

- Overall training loss

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} \{\widetilde{\mathcal{L}}_i(\boldsymbol{\theta}; \mathcal{D}_i^{te}) \triangleq \mathcal{L}_i(\boldsymbol{\theta}_i^+; \mathcal{D}_i^{te})\}$$

with $\boldsymbol{\theta}_i^+ = \boldsymbol{\theta} - \eta \nabla \mathcal{L}_i(\boldsymbol{\theta}; \mathcal{D}_i^{tr})$

- $\nabla \widetilde{\mathcal{L}}_i(\boldsymbol{\theta}; \mathcal{D}_i^{te}) = [1 - \eta \underbrace{\cancel{\nabla \otimes \nabla \mathcal{L}_i(\boldsymbol{\theta}; \mathcal{D}_i^{tr})}}_{\text{Hessian}}] \nabla \mathcal{L}_i(\boldsymbol{\theta}_i^+; \mathcal{D}_i^{te})$

big, let's ignore it

Results on mini-ImageNet

MiniImagenet (Ravi & Larochelle, 2017)	5-way Accuracy 1-shot	5-way Accuracy 5-shot
fine-tuning baseline	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
nearest neighbor baseline	$41.08 \pm 0.70\%$	$51.04 \pm 0.65\%$
matching nets (Vinyals et al., 2016)	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
meta-learner LSTM (Ravi & Larochelle, 2017)	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML, first order approx.	$\mathbf{48.07 \pm 1.75\%}$	$\mathbf{63.15 \pm 0.91\%}$

Agenda for Today

- Multi-task Learning (MTL)
- Meta Learning
- Zero-shot Learning

Setup: Zero-Shot Learning (ZSL)

- Training: input $\boldsymbol{x}_i$, label $y_i \in \mathcal{V}$
- Test: input $\boldsymbol{x}$, predict label $y \notin \mathcal{V}$
- Impossible if the labels are just categorical
- What if labels have semantics?

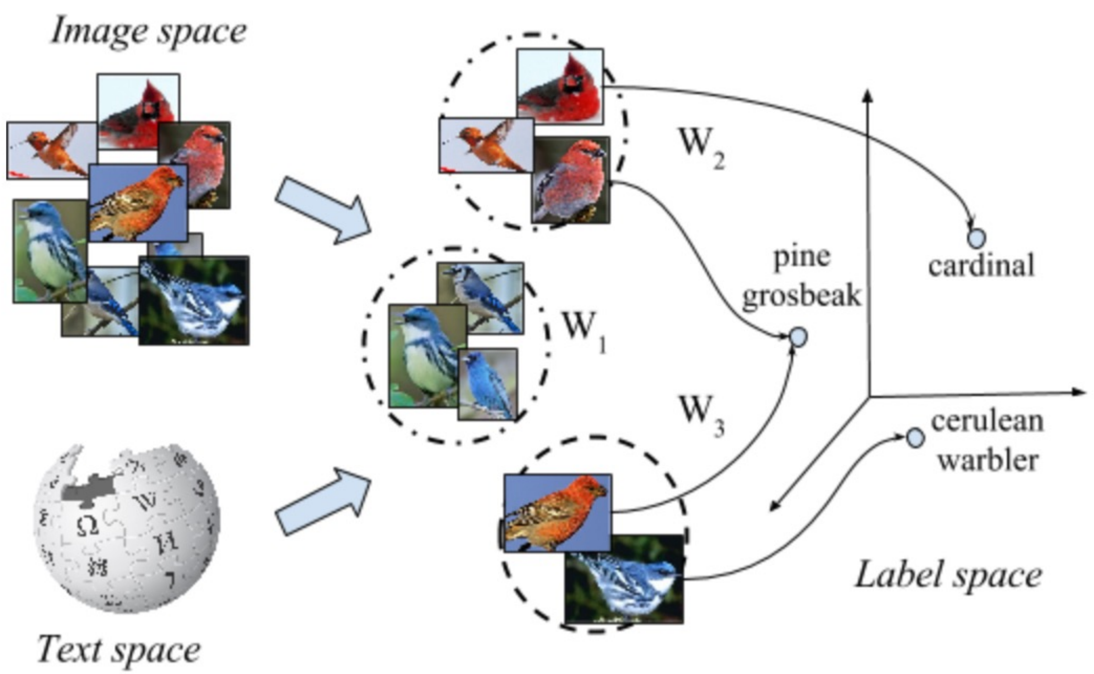
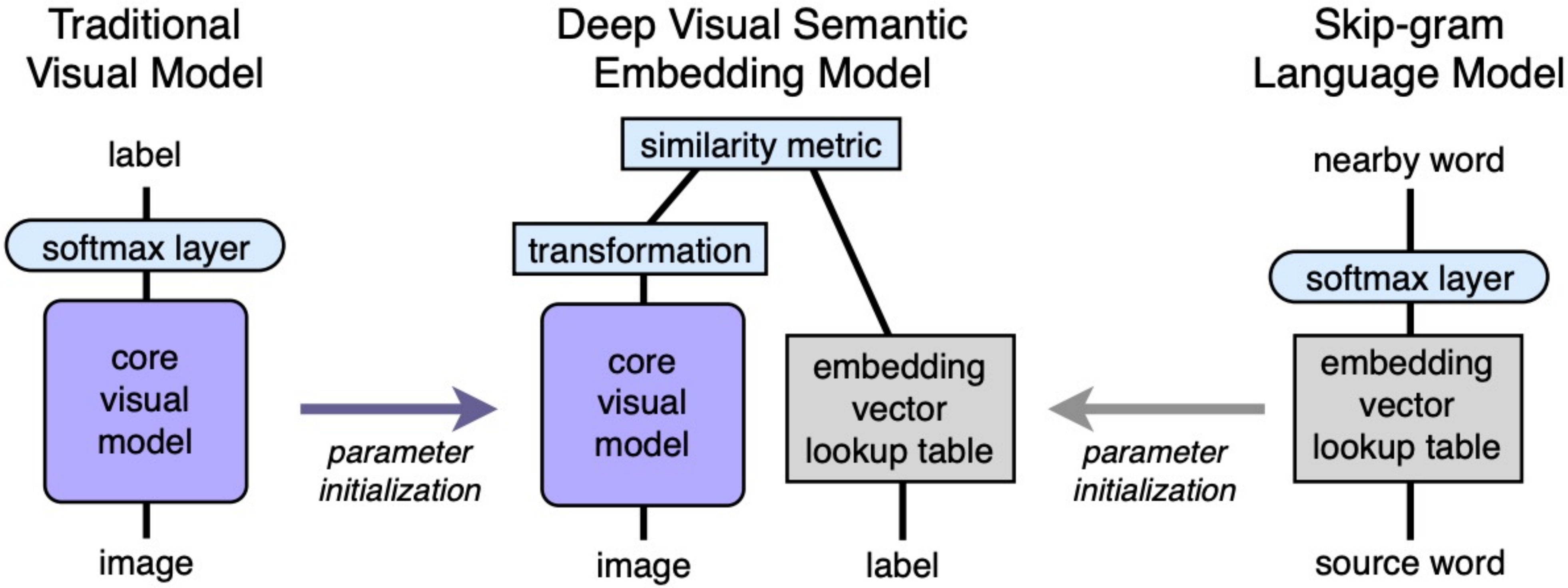


Illustration from [here]()

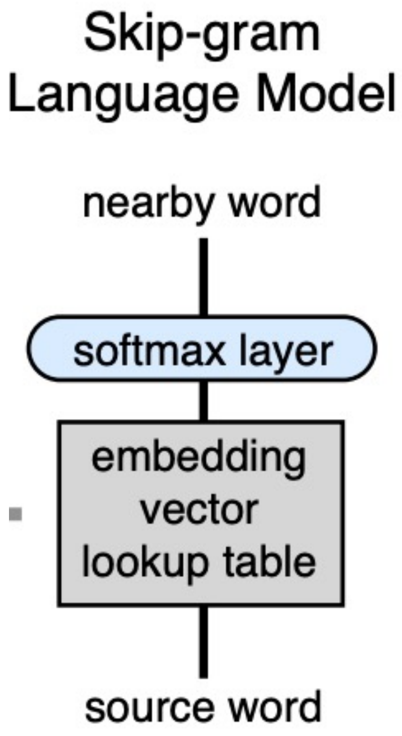
ZSL Image Classification



[Frome et. al, 2013]

Recap

- How does skip gram work?

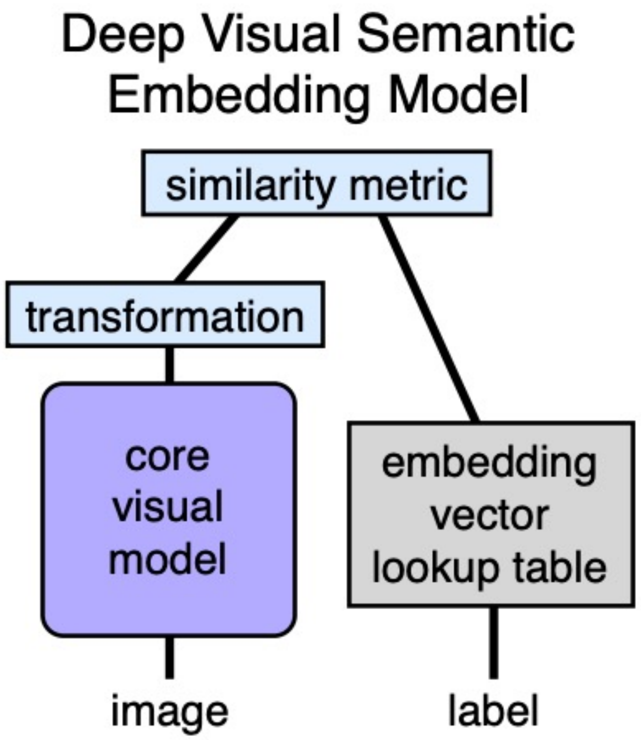


Metric Learning Objective

- The usual way to measure two vector's similarity

$$x^T y = \sum_i x_i y_i$$

- More generally, we may want to
  - weigh the dimensions
  - Consider cross dimensions
- That's $\sum_{i,j} m_{i,j} x_i y_j = x^T M y$



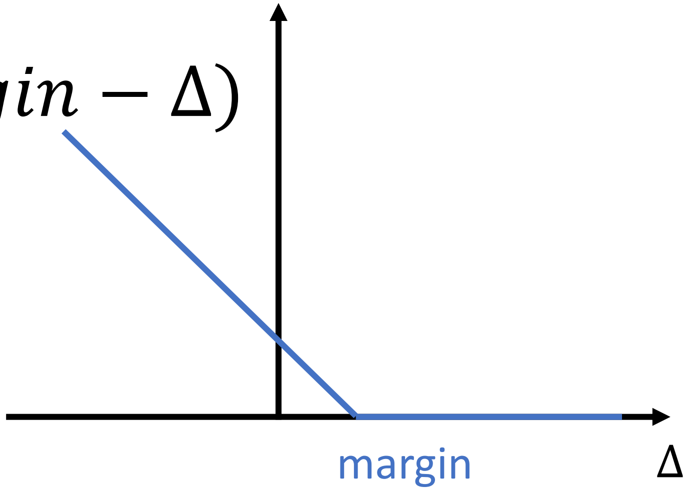
Frome et. al, 2013

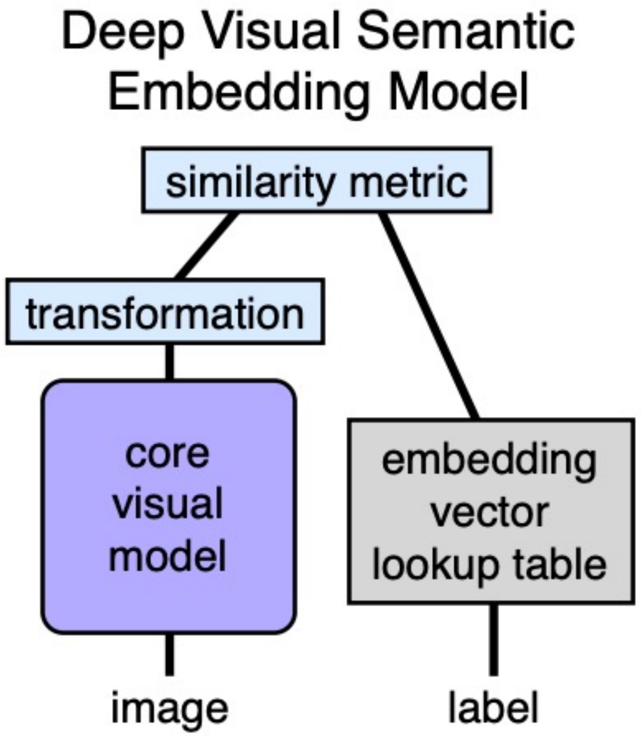
Metric Learning Objective

- Image vector should be close to its text label
- But far away from a wrong text label
- Require the distance differ by some “margin”

$$loss(image, label) = \sum_{j \neq label} \max[0, margin - \vec{t}_{label} M \vec{v}(image) + \vec{t}_j M \vec{v}(image)]$$

- Hinge loss = $\max(0, margin - \Delta)$





Testing Phase

- Classify a new image by nearest neighbor search
- But with distance metric M

$$\min_{text} \vec{v}(text) M \vec{v}(image)$$

A

ZSL	Softmax over ImageNet 1K
eyepiece, ocular	typewriter keyboard
Polaroid	tape player
compound lens	reflex camera
telephoto lens, zoom lens	CD player
rangefinder, range finder	space bar

B

ZSL	Softmax over ImageNet 1K
oboe, hautboy, hautbois	reel
bassoon	punching bag, punch bag, ...
English horn, cor anglais	whistle
hook and eye	bassoon
hand	letter opener, paper knife, ...

C

ZSL	Softmax over ImageNet 1K
barbet	patas, hussar monkey, ...
patas, hussar monkey, ...	proboscis monkey, Nasalis ...
babbler, cackler	macaque
titmouse, tit	titi, titi monkey
bowerbird, catbird	guenon, guenon monkey

D

ZSL	Softmax over ImageNet 1K
fruit	pineapple, ananas
pineapple	coral fungus
pineapple plant, Ananas ...	artichoke, globe artichoke
sweet orange	sea anemone, anemone
sweet orange tree, ...	cardoon

E

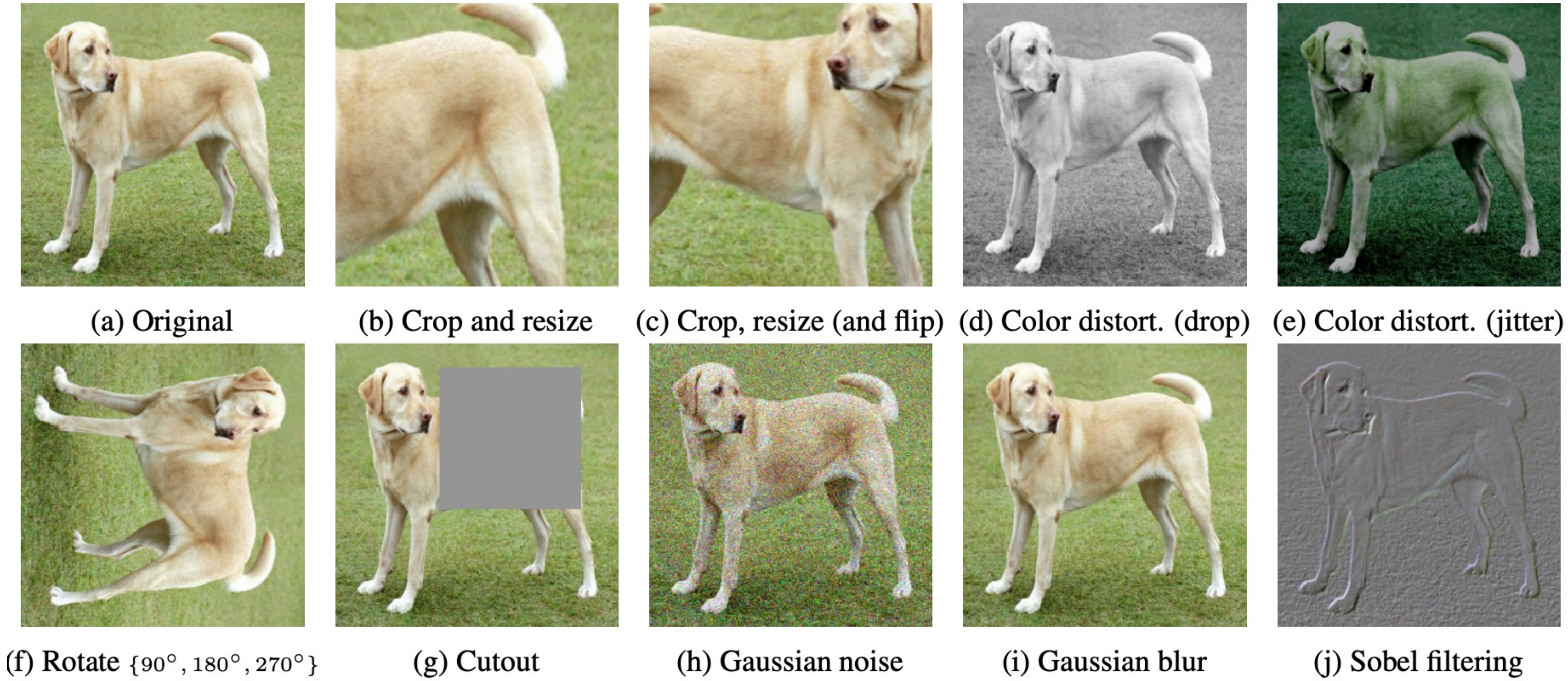
ZSL	Softmax over ImageNet 1K
comestible, edible, ...	pot, flowerpot
dressing, salad dressing	cauliflower
Sicilian pizza	guacamole
vegetable, veggie, veg	cucumber, cuke
fruit	broccoli

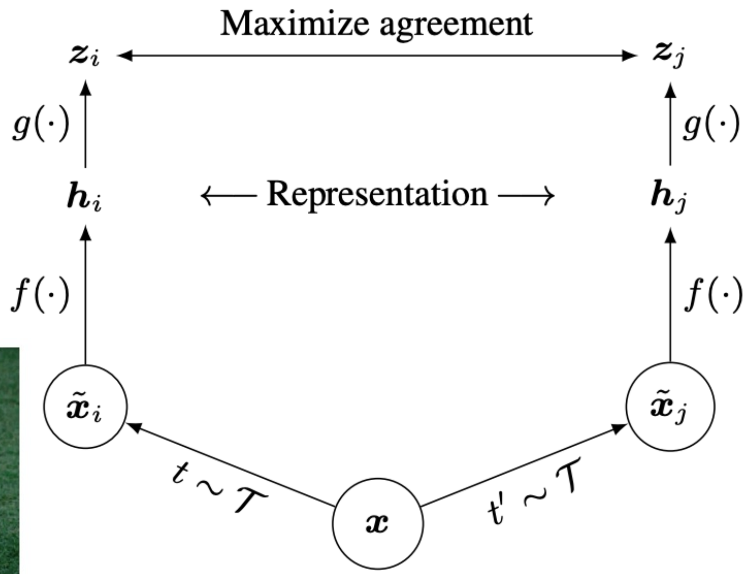
F

ZSL	Softmax over ImageNet 1K
dune buggy, beach buggy	warplane, military plane
searcher beetle, ...	missile
seeker, searcher, quester	projectile, missile
Tragelaphus eurycerus, ...	sports car, sport car
bongo, bongo drum	submarine, pigboat, sub, ...

More on Learning by Contrast

- Self-supervised pretraining of vision models
- $\mathcal{T}$: set of augmentations

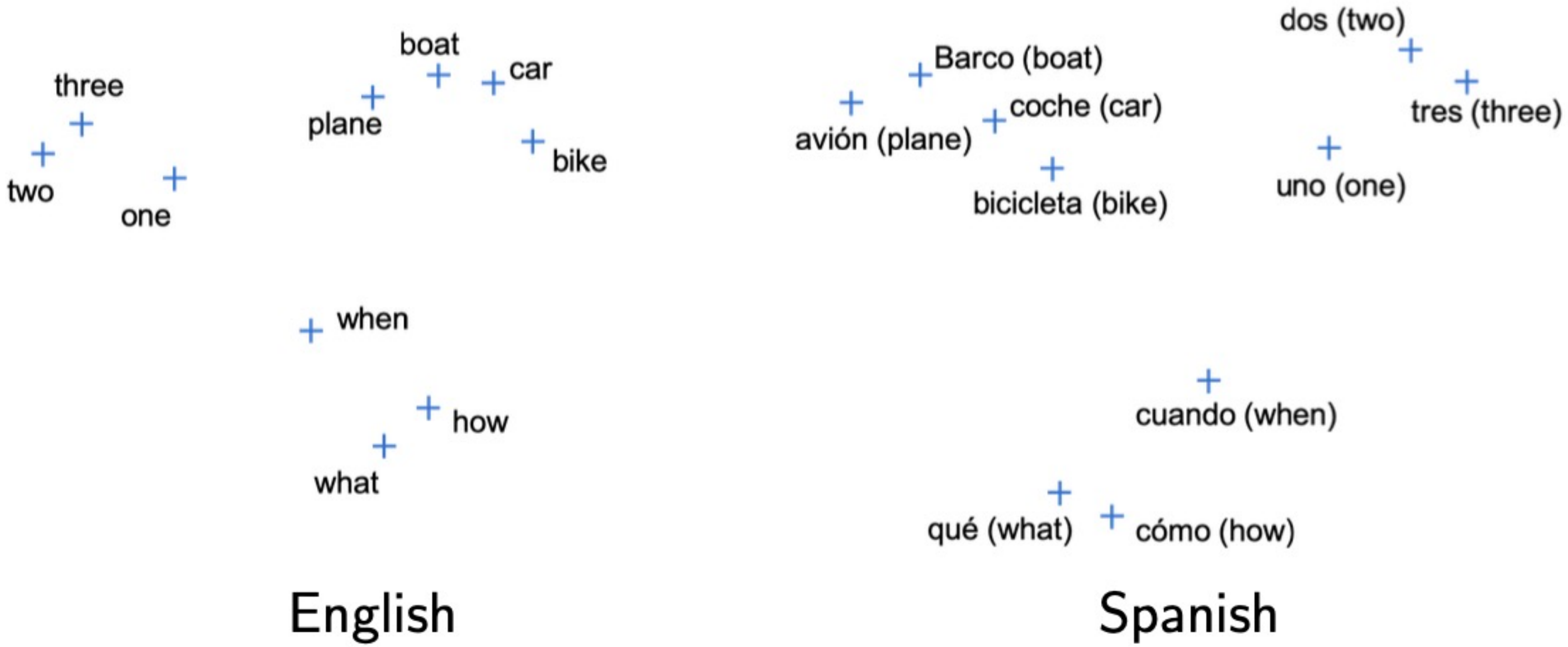




[Chen et. al, 2020]

Another ZSL: Bilingual Lexicon Induction (BLI)

- Generate word-to-word translation from very few “seeding” pairs
- Again take advantage of word embeddings

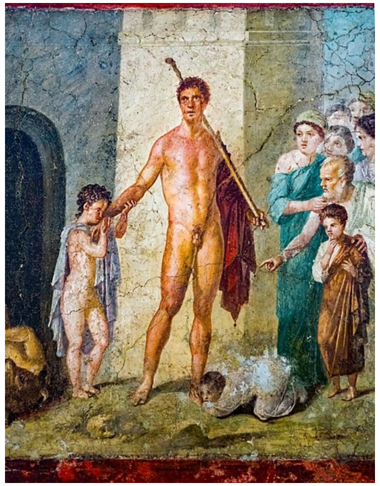


BLI

- Source embedding $X \in \mathbb{R}^{n \times d}$, target embedding $Y \in \mathbb{R}^{n \times d}$
- Learn a rotation matrix $R \in \mathbb{R}^{d \times d}, RR^T = I$
- Procrustes problem

$$\min_{R:RR^T=I} \|XR - Y\|^2$$

Procrustes



Theseus

Images from https://michelinewalker.com/2011/08/15/the-procrustean-bed-2 and https://en.wikipedia.org/wiki/Theseus

Solving the Procrustes Problem

$$\min_{R:RR^T=I} \|XR - Y\|^2$$

- We can show it's equivalent to solving

$$\max_{R:RR^T=I} \langle R, Y^T X \rangle$$

- Let the SVD of $Y^T X = U\Lambda V^T$, then optimum $R^* = UV^T$
- Why impose $RR^T = I$?
  - Prior knowledge: languages should share some fundamentals
  - A regularization