Frontiers and Open-Challenges

CS330

Logistics

The poster session is tomorrow!

Tuesday 12/2, 1:30-3:30 pm

Print your posters well ahead of time.

Final project report

Due Monday 12/16, midnight.

Welcome to submit earlier.

Hard deadline, because grades due shortly afterward

This is the last lecture!

We'll leave time for course evaluations at the end.

Today: What doesn't work very well?

(and how might we fix it)

How do we construct tasks for meta-learning?

memorization problems

can we use data without task boundaries?

can the algorithm come up with the tasks?

What does it take to run multi-task & meta-RL across distinct tasks?

how do we specify the task?

what set of distinct tasks do we train on?

what challenges arise?

Open Challenges

How do we construct distributions of tasks for meta-training?

Given 1 example of 5 classes:

training data $\mathcal{D}_{\text{train}}$

Classify new examples

test set $\mathbf{x}_{\text{test}}$

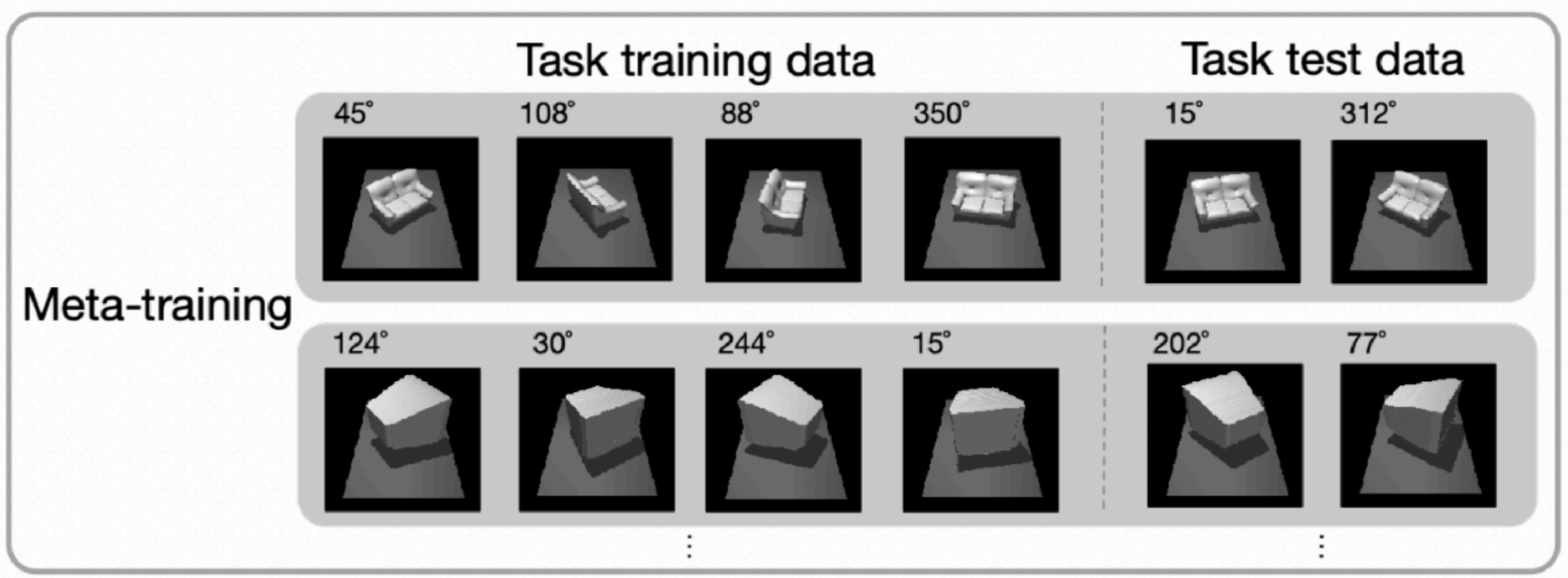
meta-training

training classes

What would happen if we didn't **shuffle the labels**?

How do we construct distributions of tasks for meta-training?

Another example: pose prediction



The learner can ignore the task data $\mathscr{D}_{\text{tr}}$.

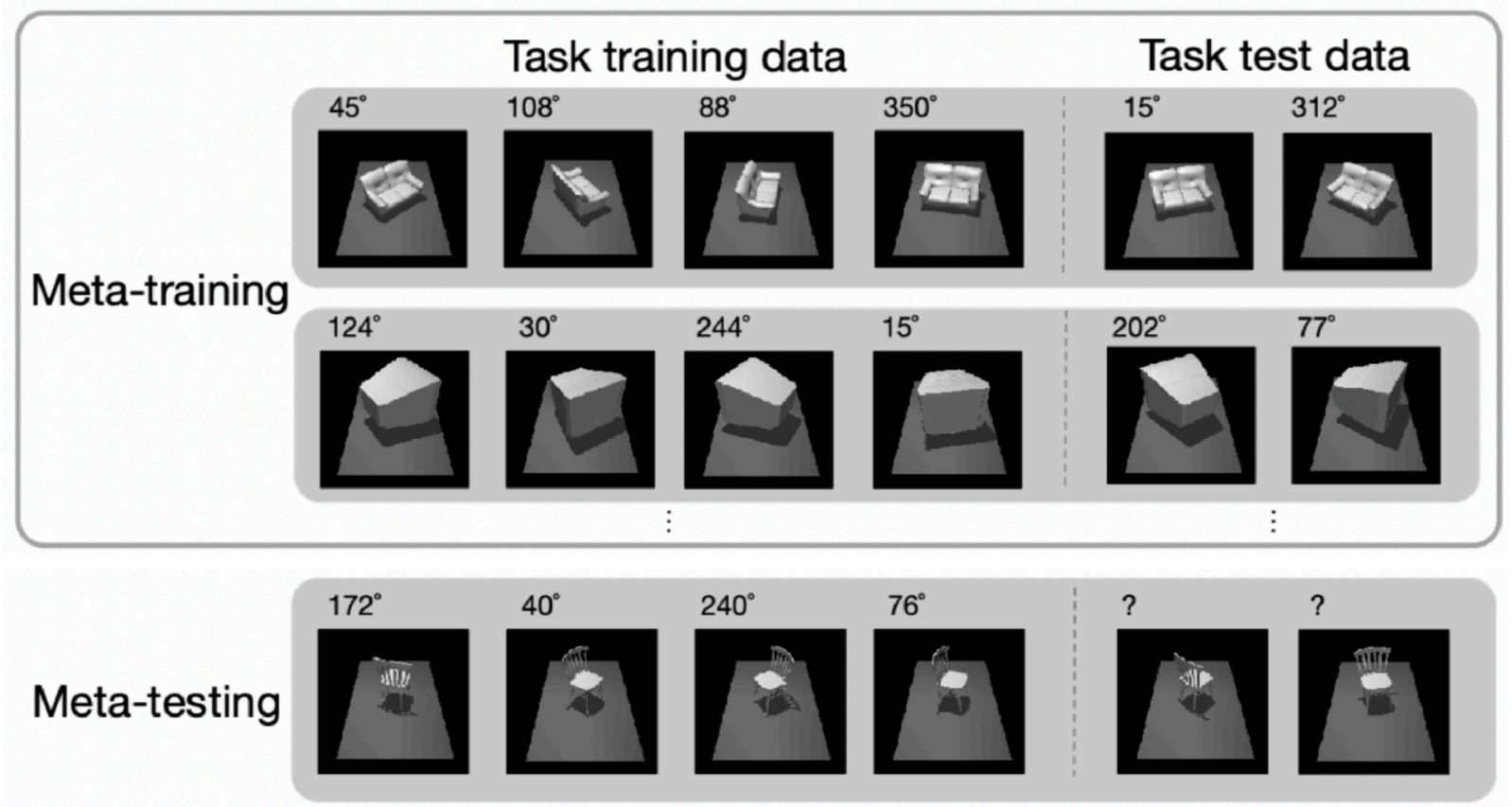
Is this bad?

When is it bad?

Yin, Tucker, Yuan, Levine, Finn. **Meta-Learning without Memorization**. '19

How do we construct distributions of tasks for meta-training?

Another example: pose prediction



Bad when given a unseen object with unseen canonical orientation.

Can we do anything about this?

Yin, Tucker, Yuan, Levine, Finn. **Meta-Learning without Memorization**. '19

If tasks *mutually exclusive*: single function cannot solve all tasks

(i.e. due to label shuffling, hiding information)

If tasks are *non-mutually exclusive*: single function can solve all tasks

multiple solutions to the meta-learning problem

$$y^{\mathrm{ts}} = f_\theta(\mathcal{D}_i^{\mathrm{tr}}, x^{\mathrm{ts}})$$

One solution: memorize canonical pose info in $\boldsymbol{\theta}$ & ignore $\mathcal{D}_i^{\mathrm{tr}}$

Another solution: carry no info about canonical pose in $\boldsymbol{\theta}$, acquire from $\mathcal{D}_i^{\mathrm{tr}}$

An entire spectrum of solutions based on how **information** flows.

Suggests a potential approach: control information flow.

Yin, Tucker, Yuan, Levine, Finn. **Meta-Learning without Memorization**. '19

If tasks are *non-mutually exclusive*: single function can solve all tasks

multiple solutions to the meta-learning problem

$$y^{\text{ts}} = f_\theta(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$

One solution: memorize canonical pose info in θ & ignore $\mathcal{D}_i^{\text{tr}}$

Another solution: carry no info about canonical pose in θ, acquire from $\mathcal{D}_i^{\text{tr}}$

An entire spectrum of solutions based on how **information** flows.

Suggests a potential approach: control information flow.

Meta-regularization (MR): minimize meta-training loss + information in θ

$$\mathscr{L}(\theta, \mathscr{D}_{meta-train}) + \beta D_{KL}(q(\theta; \theta_\mu, \theta_\sigma) \| p(\theta))$$

Places precedence on using information from $\mathcal{D}_i^{\text{tr}}$ over θ.

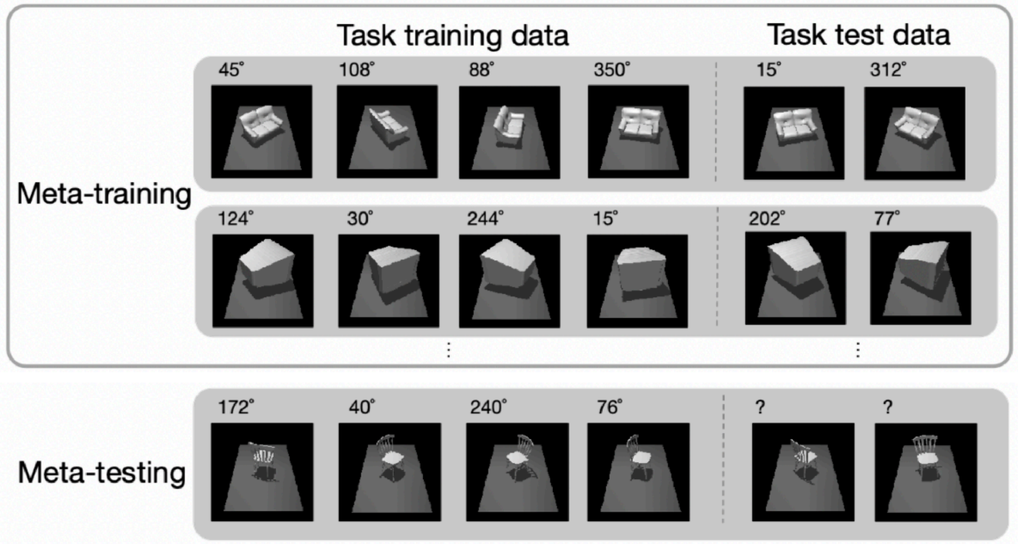
Can combine with your favorite meta-learning algorithm.

Yin, Tucker, Yuan, Levine, Finn. **Meta-Learning without Memorization**. '19

Omniglot without label shuffling: "non-mutually-exclusive" Omniglot

NME Omniglot	20-way 1-shot	20-way 5-shot
MAML	7.8 (0.2)%	50.7 (22.9)%
TAML	9.6 (2.3)%	67.9 (2.3)%
MR-MAML (W) (ours)	**83.3 (0.8)%**	**94.1 (0.1)%**

On **pose prediction** task:



Method	MAML	MR-MAML(W) (ours)	CNP	MR-CNP(W) (ours)
MSE	5.39 (1.31)	**2.26 (0.09)**	8.48 (0.12)	2.89 (0.18)

(and it's not just as simple as standard regularization)

CNP	CNP + Weight Decay	CNP + BbB	MR-CNP (W) (ours)
8.48 (0.12)	6.86 (0.27)	7.73 (0.82)	**2.89 (0.18)**

TAML: Jamal & Qi. **Task-Agnostic Meta-Learning for Few-Shot Learning**. CVPR '19

Yin, Tucker, Yuan, Levine, Finn. **Meta-Learning without Memorization**. '19

Today: What doesn't work very well?

(and how might we fix it)

How do we construct tasks for meta-learning?

memorization problems

can we use data without task boundaries?

can the algorithm come up with the tasks?

What if we simply have a time series of data?

- predict energy demand
- dynamics of a robot, car
- transportation usage
- stock market
- video analytics
- RL agent

unsegmented
yet, exhibits **temporal structure**

Can we **segment time series** into tasks & **meta-learn** across tasks?

How to segment?

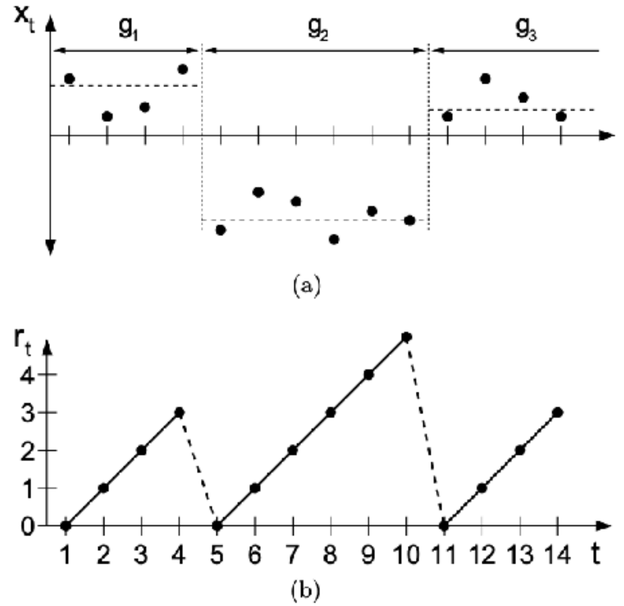
Bayesian online change point detection (BOCPD)

Adams & Mackay '17

Problem: assume task will switch with some probability, at each time t

Maintain **belief over task duration** (run length), **posterior** for each duration

Recursively update belief using **model performance**



BOCPD is differentiable! —> **backprop through update belief update to meta-train model**

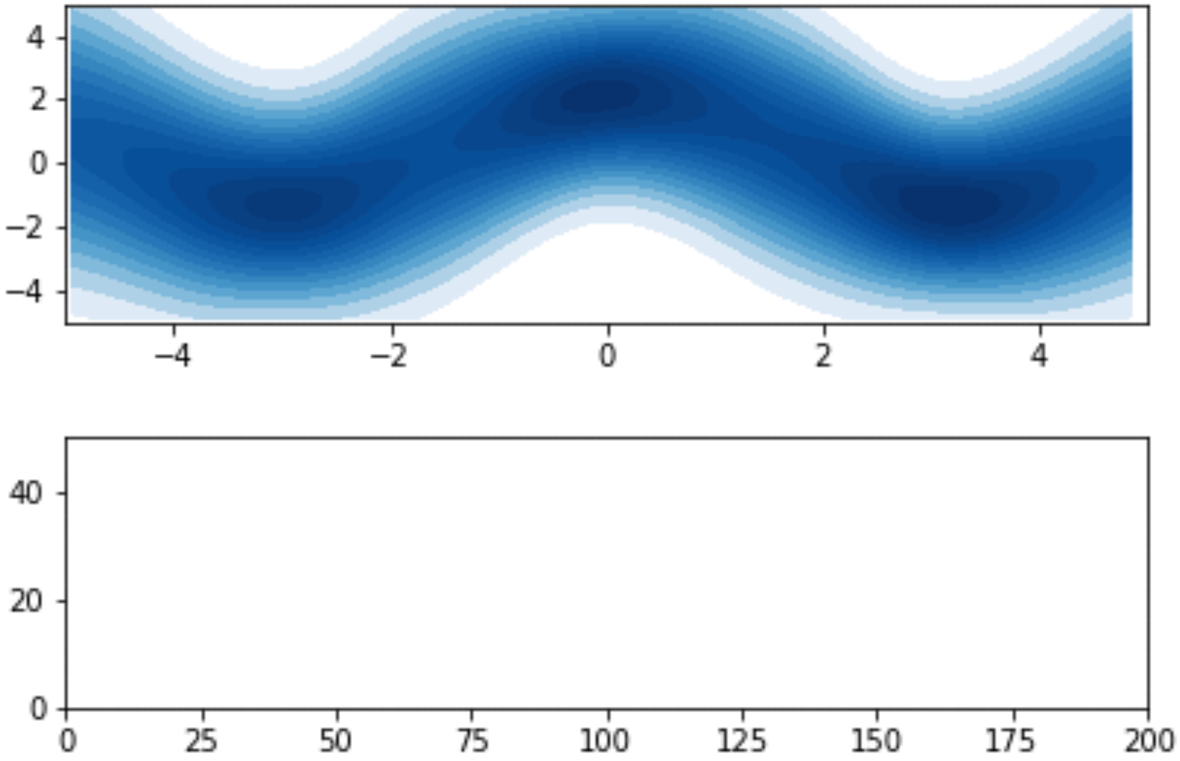
Harrison, Sharma, Finn, Pavone. **Continuous Meta-Learning without Tasks**. '19

Meta-Learning with Online Changepoint Analysis (MOCA)

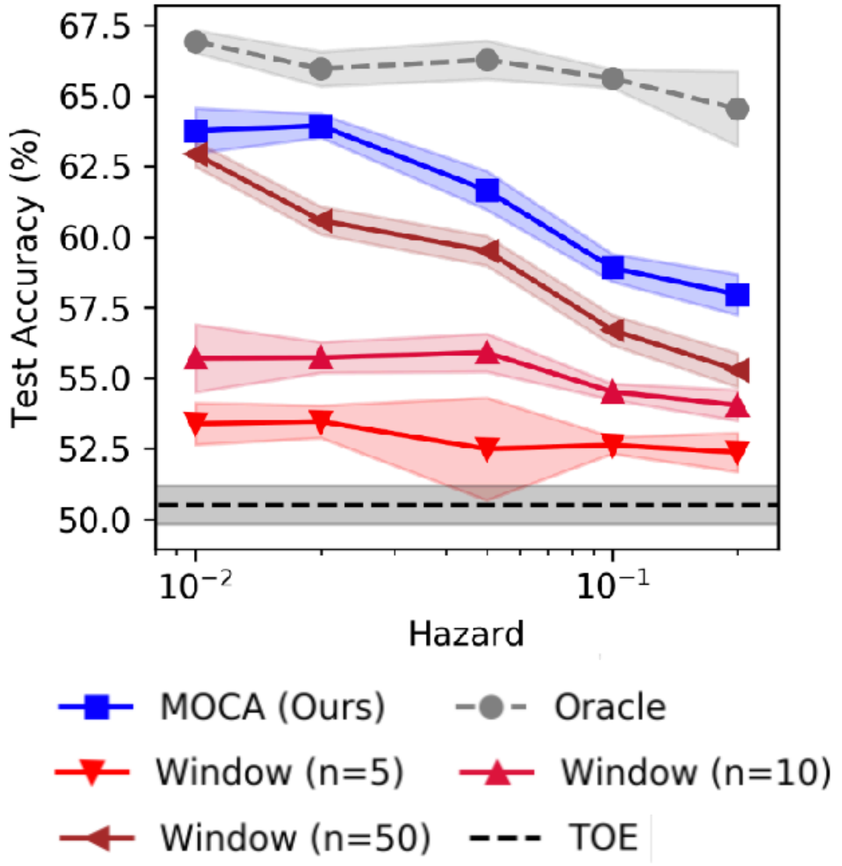
Meta-training phase: given unsegmented time-series of offline data

Meta-test phase: streaming online learning & prediction

Sinusoid regression with discrete shifts



Streaming variant of MiniImagenet.



Today: What doesn't work very well?

(and how might we fix it)

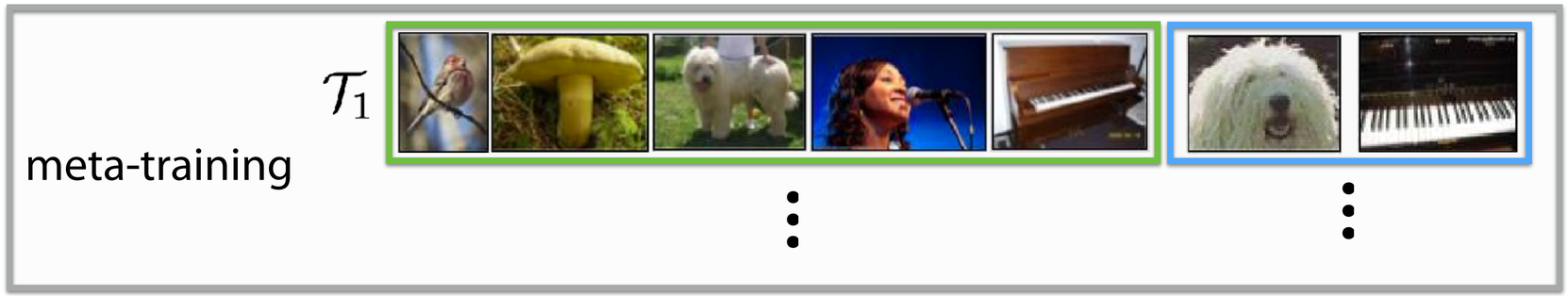
How do we construct tasks for meta-learning?

memorization problems

can we use data without task boundaries?

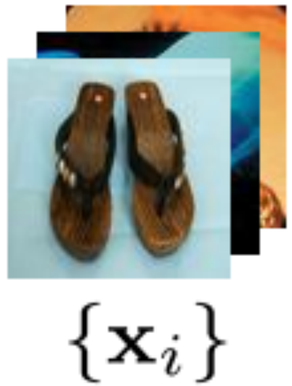
can the algorithm come up with the tasks?

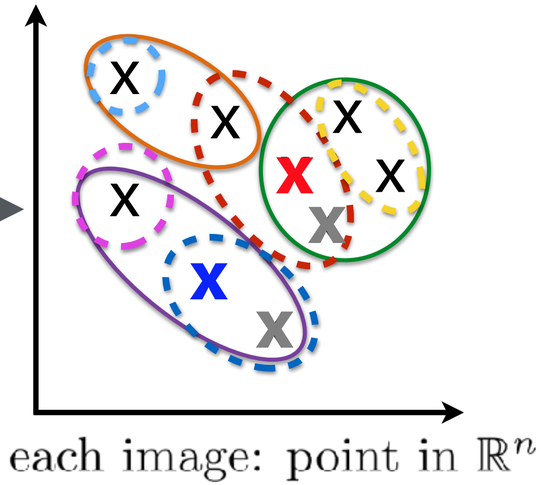
Can we meta-learn with only **unlabeled** images?



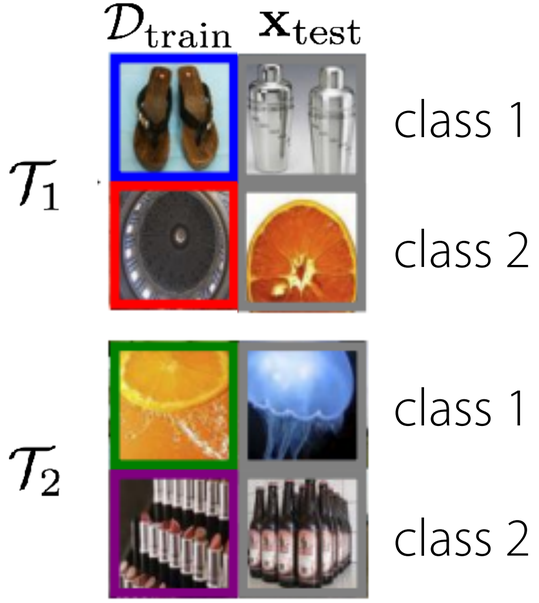
Construct tasks without labeled data?

Unsupervised learning
(to get an embedding space)

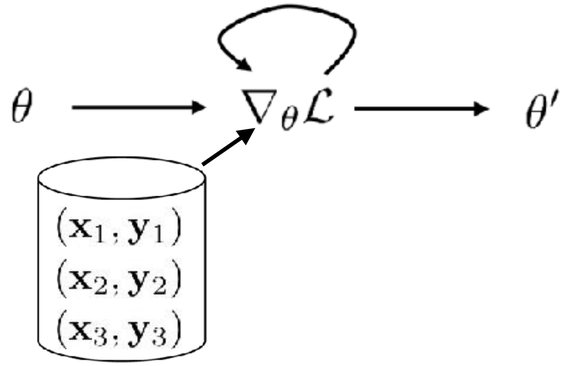




Propose tasks



Run meta-learning



Result: representation suitable for learning downstream tasks

Hsu, Levine, Finn. **Unsupervised Learning via Meta-Learning**. ICLR '18

Propose tasks for meta-learning with only **unlabeled** images?

Unsupervised learning (to get an embedding space) → **Propose tasks** → **Run meta-learning**

A few options:
BiGAN — Donahue et al. '17
DeepCluster — Caron et al. '18

Clustering to Automatically Construct Tasks for Unsupervised Meta-Learning (CACTUs)

MAML — Finn et al. '17
ProtoNets — Snell et al. '17

CACTUs MAML

miniImageNet 5-way 5-shot

method	accuracy
MAML with labels	62.13%
BiGAN kNN	31.10%
BiGAN logistic	33.91%
BiGAN MLP + dropout	29.06%
BiGAN cluster matching	29.49%
BiGAN CACTUs MAML	51.28%
DeepCluster CACTUs MAML	**53.97%**

Same story for:

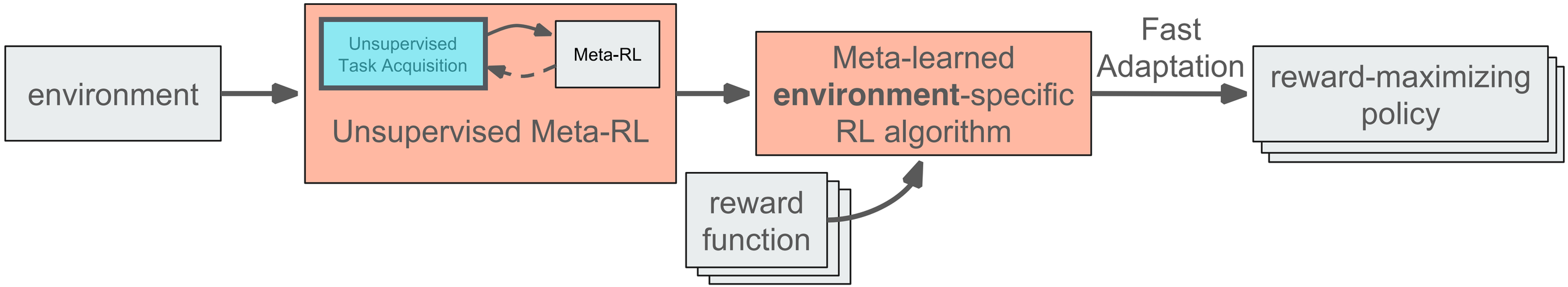
- **4 different embedding methods**
- **4 datasets** (Omniglot, CelebA, miniImageNet, MNIST)
- **2 meta-learning methods** (*)
- Test tasks with **larger datasets**

*ProtoNets underperforms in some cases.

Hsu, Levine, Finn. Unsupervised Learning via Meta-Learning. ICLR'19

What about Unsupervised Meta-RL?

General Recipe



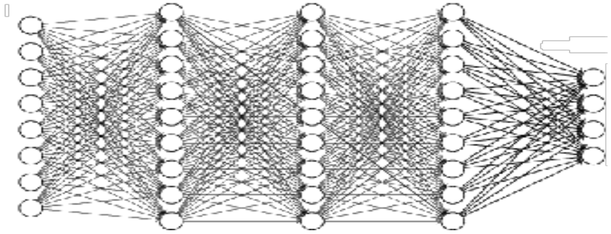
Gupta, Eysenbach, Finn, Levine. Unsupervised Meta-Learning for Reinforcement Learning.

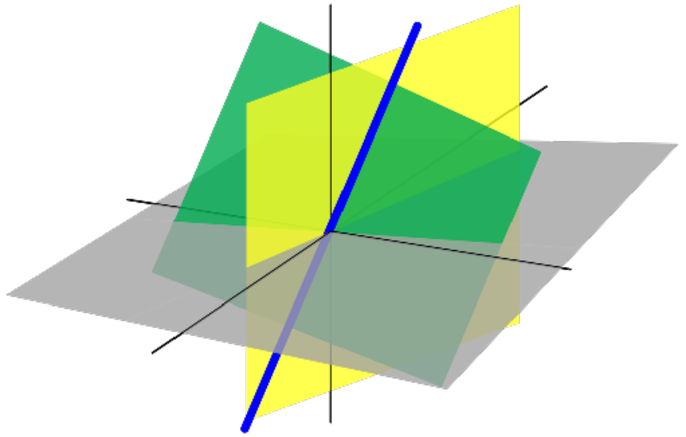
Random Task Proposals

- Use randomly initialize discriminators for reward functions

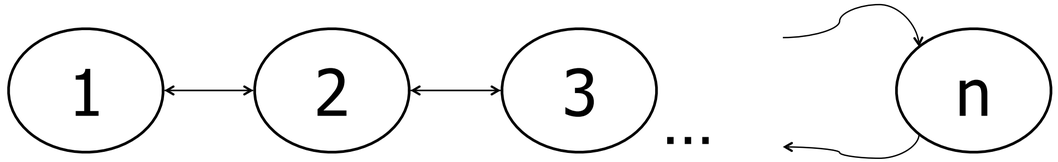
$$R(s, z) = \log p_D(z|s)$$

D → **randomly initialized network**



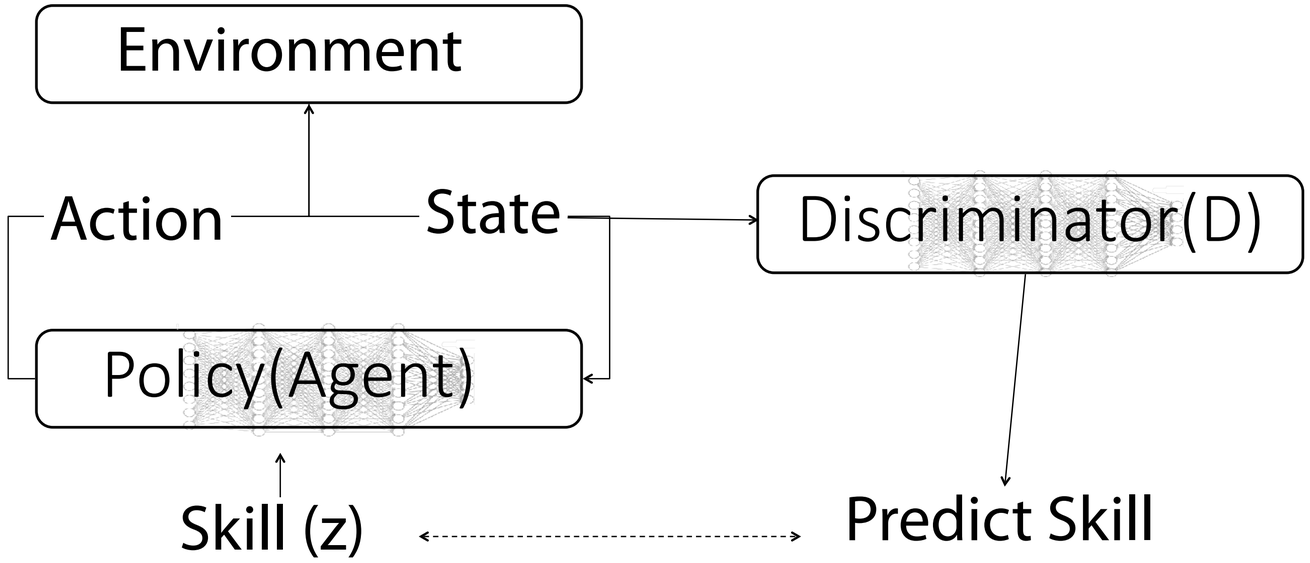


- Important: Random functions over state space, not random policies



Random policy – exponential
Random reward – polynomial

Diversity-Driven Proposals

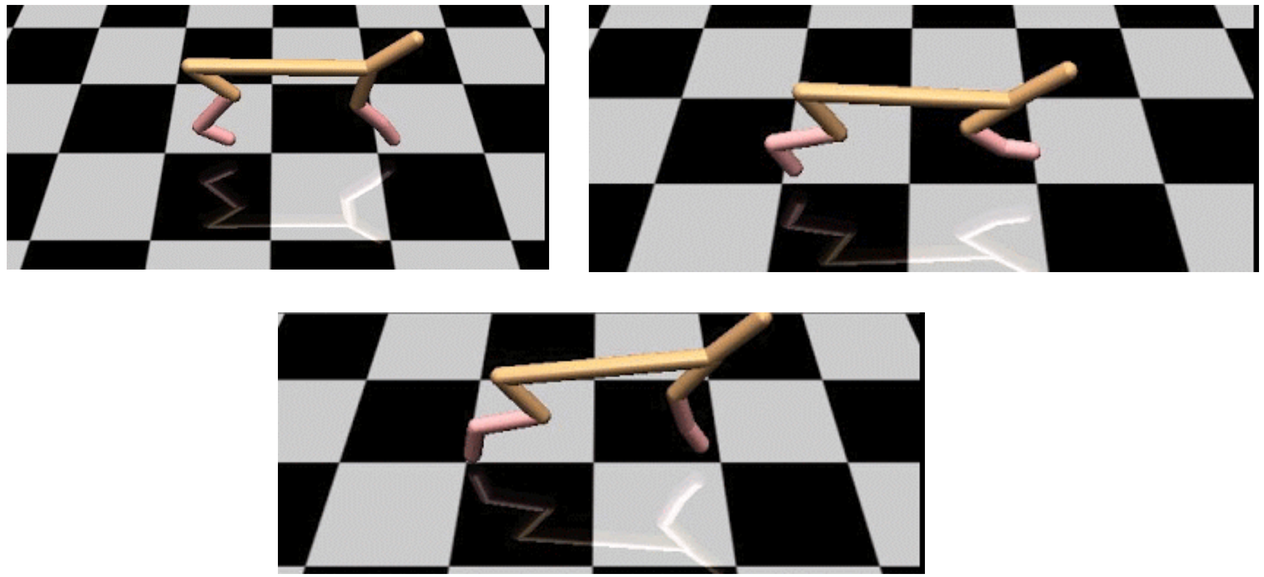


- Policy → visit states which are discriminable
- Discriminator → predict skill from state

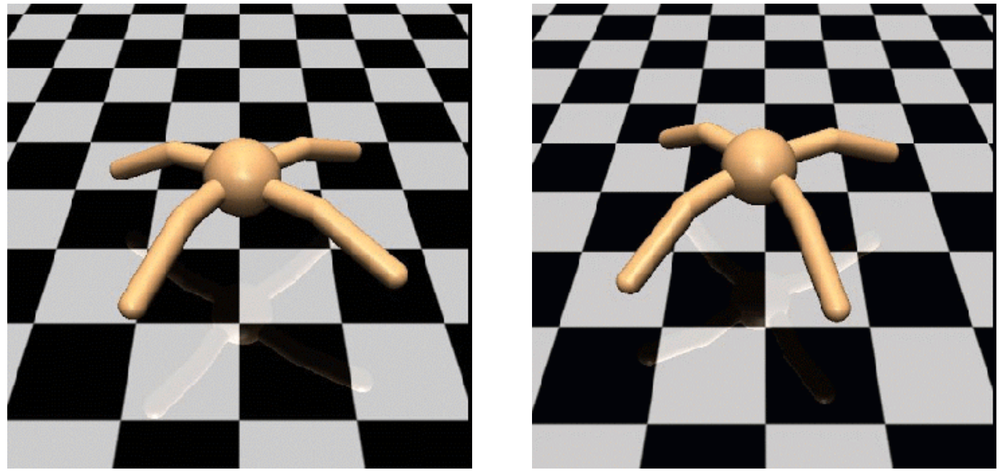
Task Reward for UML: $R(s, z) = \log p_D(z|s)$

Eysenbach, Gupta, Ibarz, Levine. Diversity is All You Need.

Examples of Acquired Tasks



Cheetah

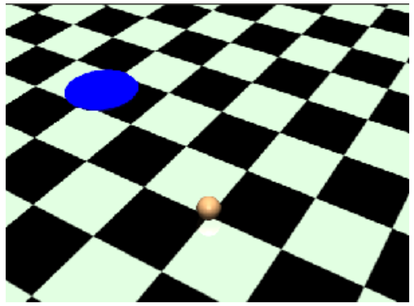


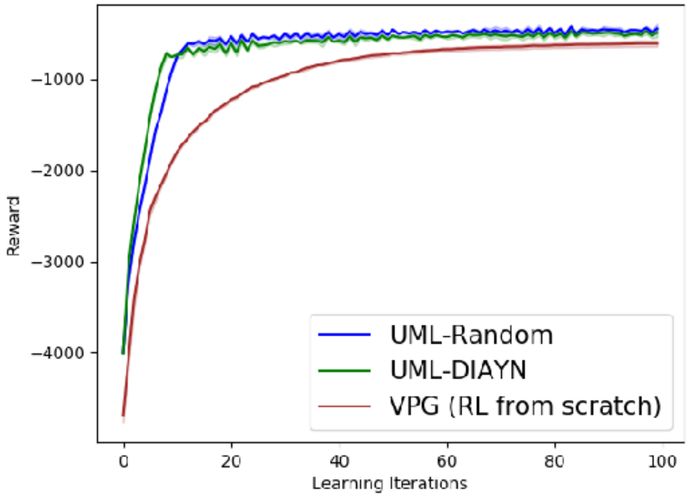
Ant

Eysenbach, Gupta, Ibarz, Levine. Diversity is All You Need.

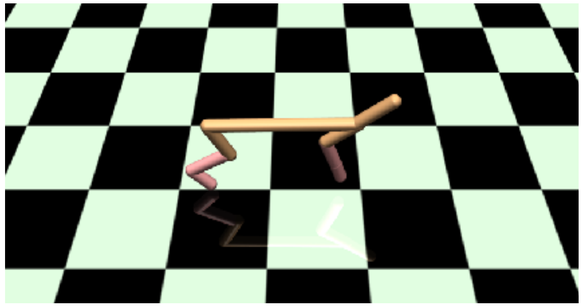
Does it work?

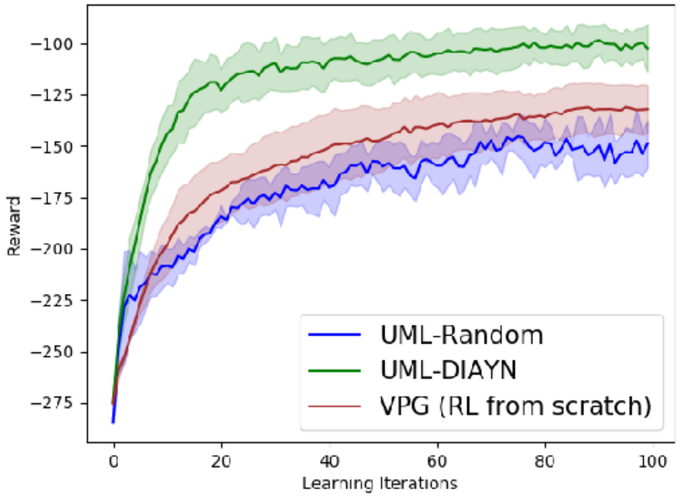
2D Navigation



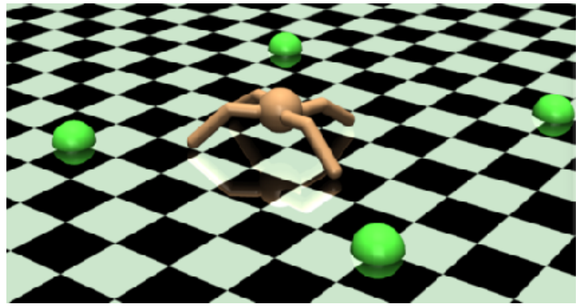


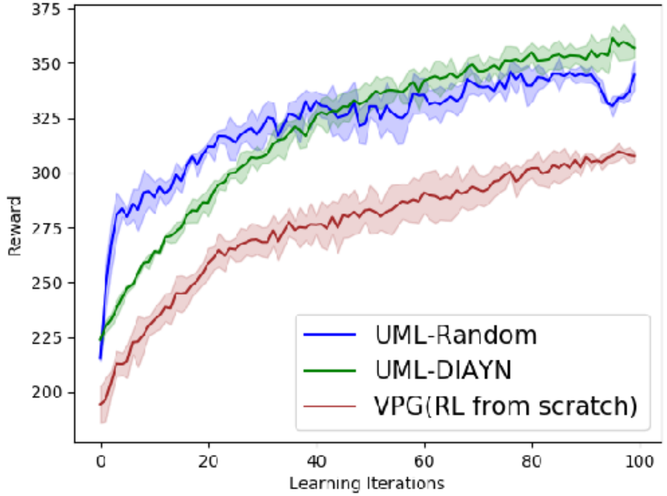
Cheetah





Ant





Meta-test performance with rewards

Takeaway: Relatively **simple** mechanisms for proposing tasks work surprisingly well.

Gupta, Eysenbach, Finn, Levine. **Unsupervised Meta-Learning for Reinforcement Learning.**

Today: What doesn't work very well?

(and how might we fix it)

How do we construct tasks for meta-learning?

memorization problems

can we use data without task boundaries?

can the algorithm come up with the tasks?

Takeaways:

Can learn priors for few-shot adaptation using:

- non-mutually exclusive tasks — through meta-regularization
- from unsegmented time series — via end-to-end changepoint detection
- from unlabeled data and experience. — using clustering

Should make it *significantly easier* to deploy meta-learning algorithms!

Today: What doesn't work very well?

(and how might we fix it)

How do we construct tasks for meta-learning?

memorization problems

can we use data without task boundaries?

can the algorithm come up with the tasks?

What does it take to run multi-task & meta-RL across distinct tasks?

how do we specify the task?

what set of distinct tasks do we train on?

what challenges arise?

Open Challenges

Have MAML, PEARL accomplished our goal of making policy adaptation fast?

Sort of…

Can we adapt to *entirely new tasks*?

meta-train task distribution

=

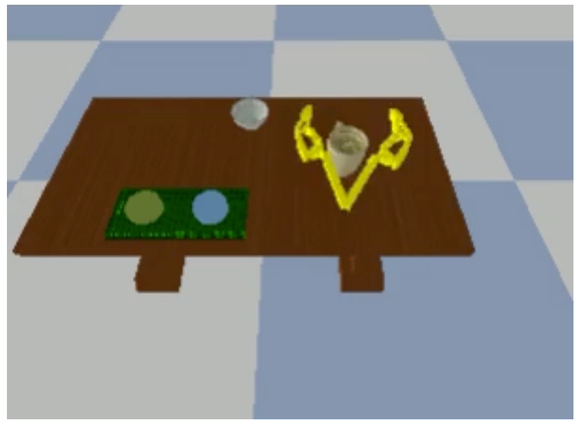
meta-test task distribution

—> Need **broad distribution of tasks** for meta-training

Can we perform meta-training *across task families*?

Can we meta-learn *across task families*?

Space of manipulation tasks



- grasping objects
- pressing buttons
- sliding objects
- stacking two objects

Goal: Learn a new variation of one of these task families with a **small number of trials** & **sparse rewards**

Problem: Robot will have to explore **every possible task**.

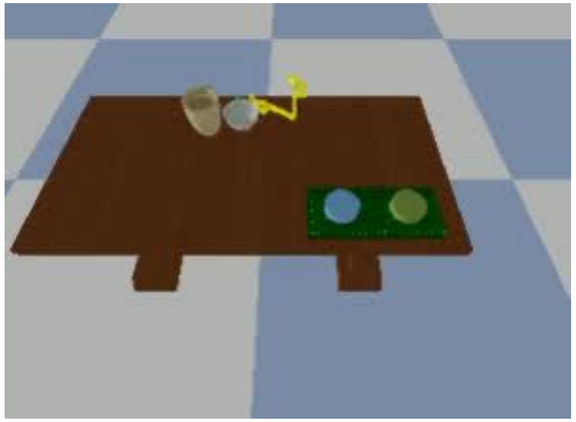
This work: Can we learn from **one demonstration** & **a few trials**?

(to convey the task) (to figure out how to solve it)

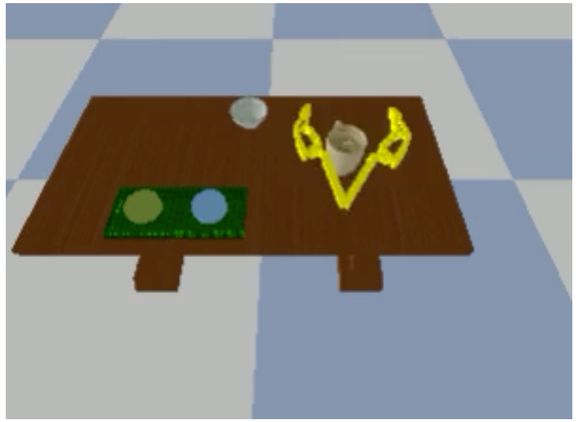
Zhao, Jang, Kappler, Herzog, Khansari, Bai, Kalakrishnan, Levine, Finn. **Watch-Try-Learn**. '19

Can we learn from one demonstration & a few trials?

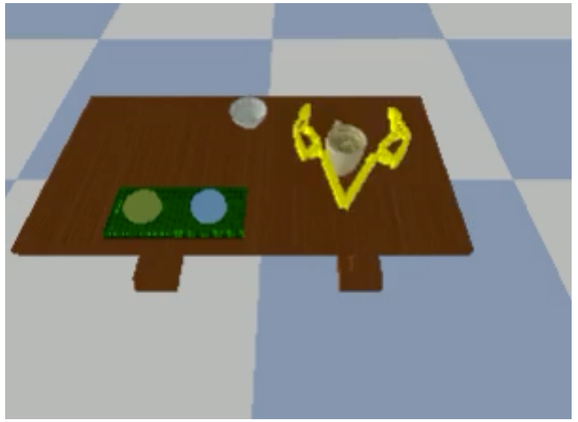
Watch one task demonstration



Try task in new situation



Learn from demo & trial to solve task



How can we train for this in a scalable way?

1. Collect a **few** demonstrations for **many** different tasks
2. Train a **one-shot imitation learning** policy.
3. Collect trials for each task by running one-shot imitation policy.
   [batch off-policy collection]
4. Train **"re-trial" policy** through imitation objective. $\mathcal{D}_{\text{train}}$: demo + trial(s)

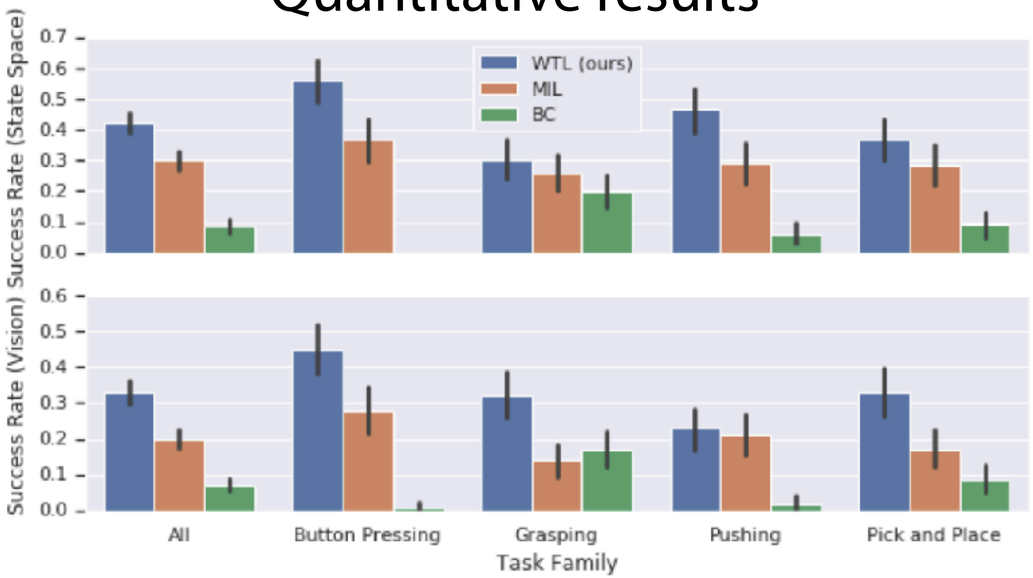
Zhao, Jang, Kappler, Herzog, Khansari, Bai, Kalakrishnan, Levine, Finn. **Watch-Try-Learn**. '19

Experiments

Compare:

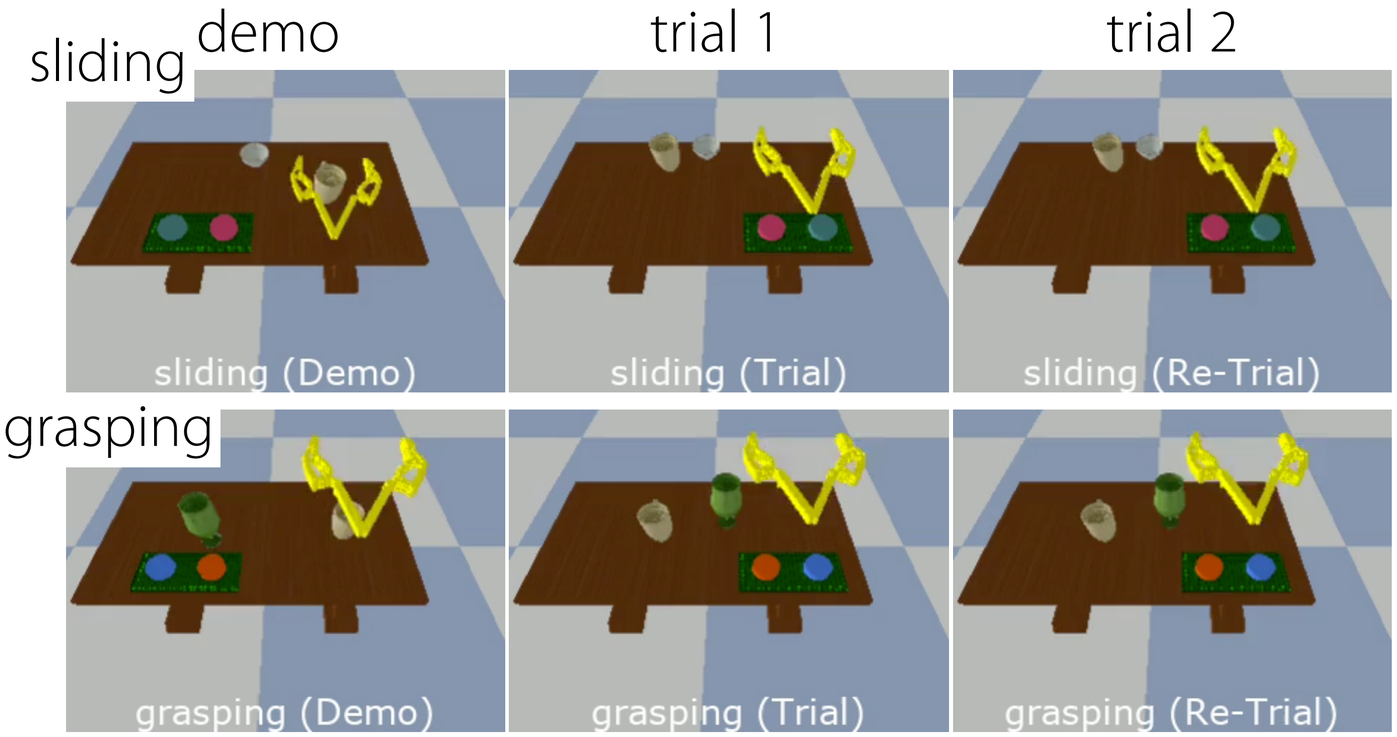
- watch-try-learn (one trial + one demo)
- meta-reinforcement learning (only use trials)
- meta imitation learning (only use demonstration)
- behavior cloning across all tasks (no meta-learning)

Quantitative results



- WTL learns across **4 distinct task families**
- significantly outperforms using **only trials** or only demos

Qualitative examples



Reinforcement learning from BC initialization requires **900 trials** to match performance of WTL.

A side note on memorization.

WTL set-up: Demonstration **partially specifies the task**, hence **requiring trials** to identify the task.

What if the demo fully specifies the task?

will resolve to one-shot imitation learning

will *ignore the trials* during meta-training

🙂 Fine if good at meta-test tasks.

☹️ But not, if robot fails and can't adapt quickly.

This is a variant on the memorization problem!

Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine **One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning**. RSS '18

Has meta-RL accomplished our goal of making policy adaptation fast?

Can we adapt to *entirely new tasks*?

meta-train task distribution

=

meta-test task distribution

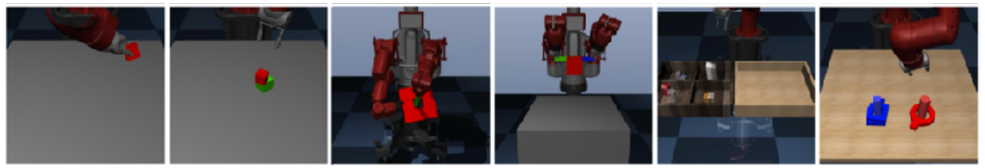
—> Need **broad distribution of tasks** *& not sparse* for meta-training

WTL trains across task families.

A few options:

Brockman et al. *OpenAI Gym*. 2016

Bellemare et al. *Atari Learning Environment*. 2016



Fan et al. *SURREAL: Open-Source Reinforcement Learning Framework and Robot Manipulation Benchmark*. CoRL 2018

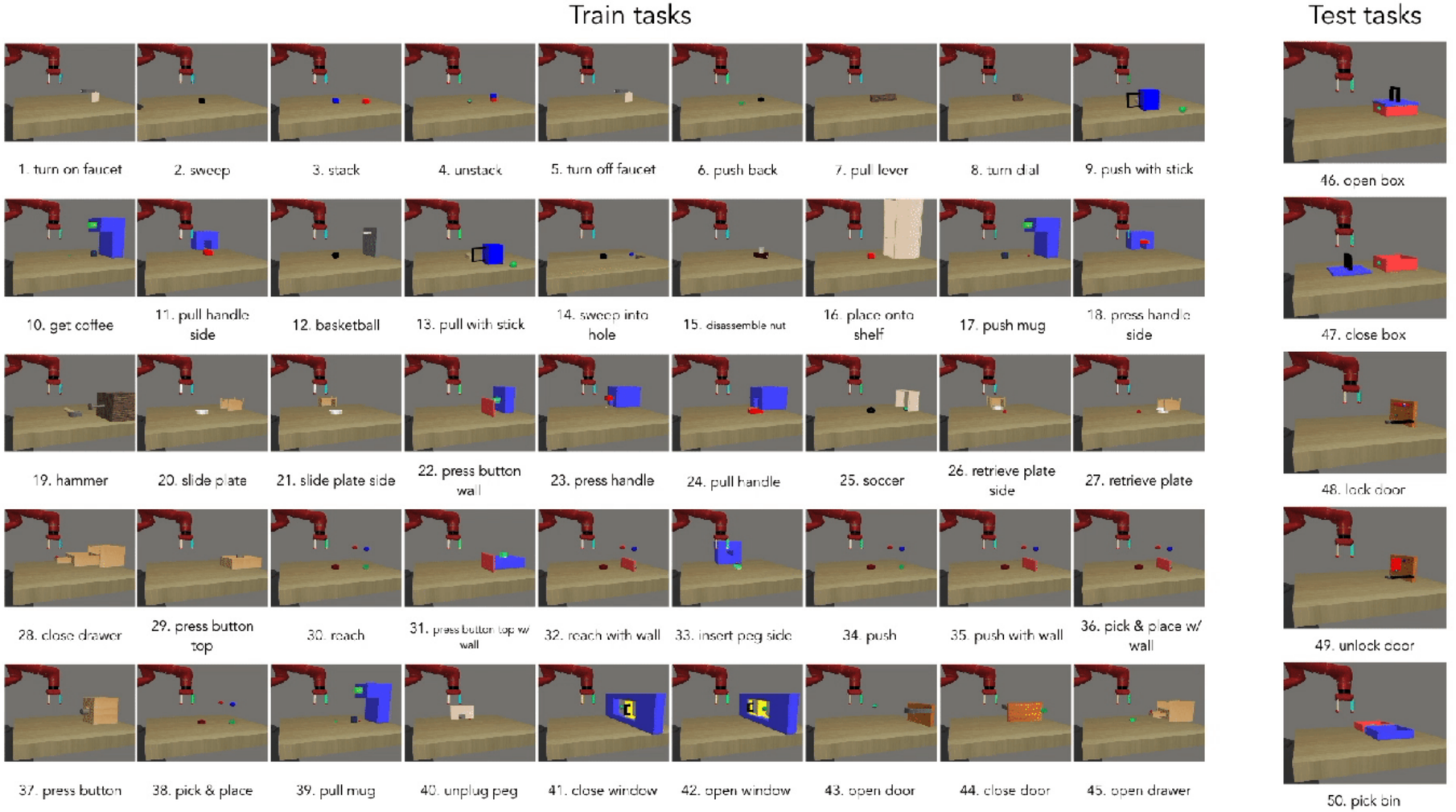
Our desiderata

50+ **qualitatively distinct** tasks

shaped **reward function** & **success metrics**

All tasks individually solvable (to allow us to focus on multi-task / meta-RL component)

Unified state & action space, environment (to facilitate transfer)



Meta-World Benchmark

Current results: signs of life, but significant room for improvement

T Yu, D Quillen, Z He, R Julian, K Hausman, C Finn, S Levine. ***Meta-World***. CoRL '19

Results: Meta-learning algorithms seem to struggle…

Methods	ML45	
	meta-train	meta-test
MAML		
RL^2		
PEARL		

…even on the 45 **meta-training** tasks!

Multi-task RL algorithms also struggle…

Methods	MT50
Multi-task PPO	8.98%
Multi-task TRPO	22.86%
Task embeddings	15.31%
Multi-task SAC	28.83%
Multi-task multi-head SAC	**35.85%**

T Yu, D Quillen, Z He, R Julian, K Hausman, C Finn, S Levine. ***Meta-World***. CoRL '19

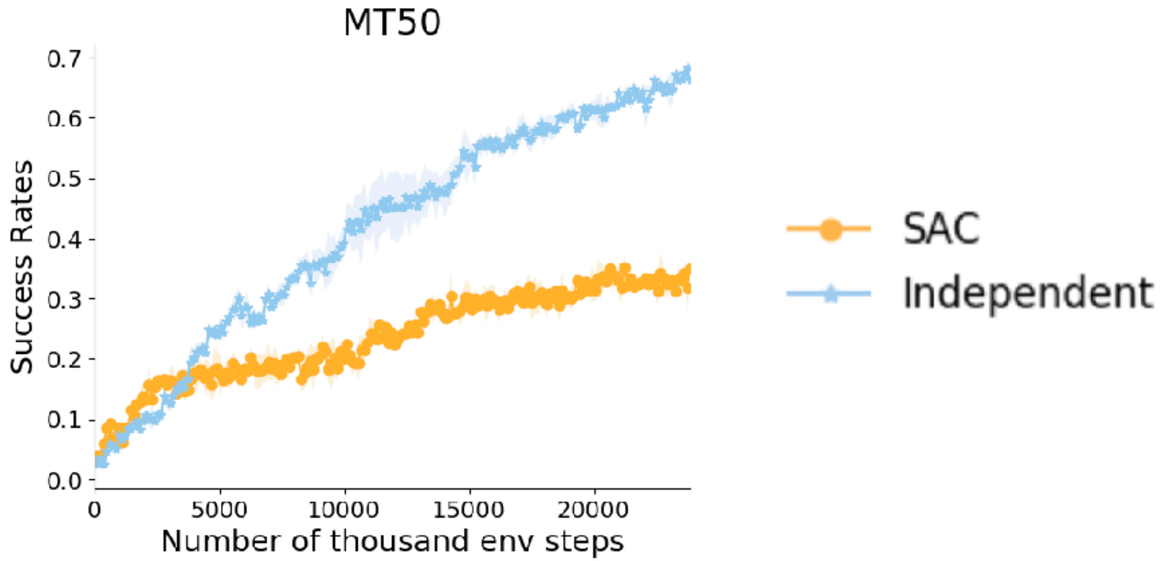
Why the poor results?

Exploration challenge? All tasks individually solvable.

Data scarcity? All methods given budget with plenty of samples.

Limited model capacity? All methods plenty of capacity.

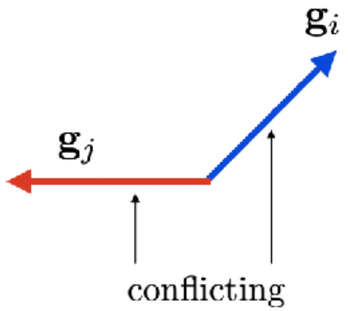
Training models *independently* performs the best.



Our conclusion: must be an *optimization* challenge.

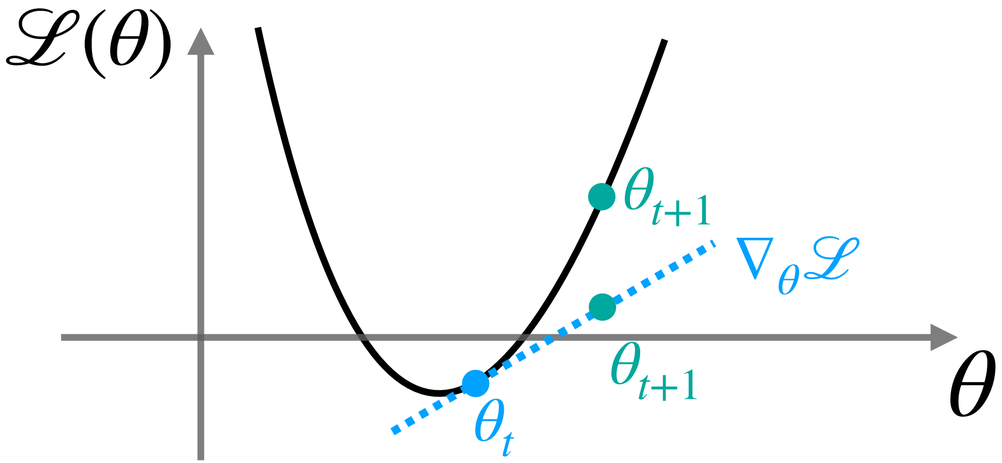
Hypothesis 1: Gradients from different tasks often conflict

If so: would see **negative inner product** of gradients.



Hypothesis 2: When they do conflict, they cause more damage than expected.

i.e. due to high curvature

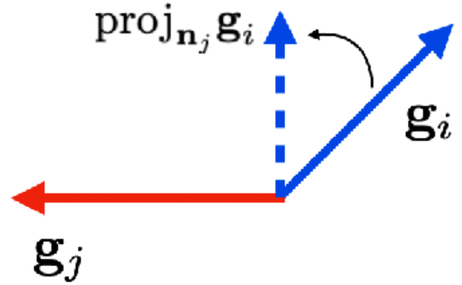


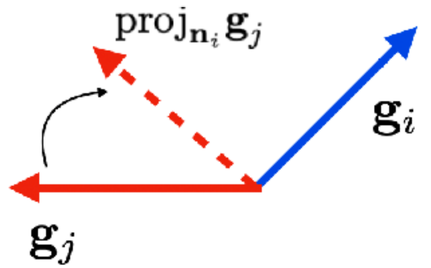
T Yu, S Kumar, A Gupta, S Levine, K Hausman, C Finn. ***Gradient Surgery for Multi-Task Learning***. '19

Our solution: try to avoid making other tasks worse, when taking gradient step.

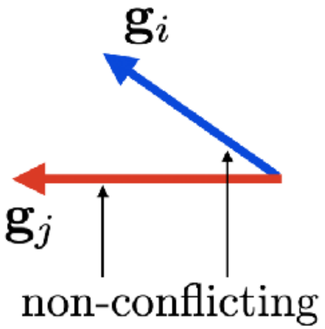
Algorithm:

If two gradients *conflict*:
project each onto the normal plane of the other





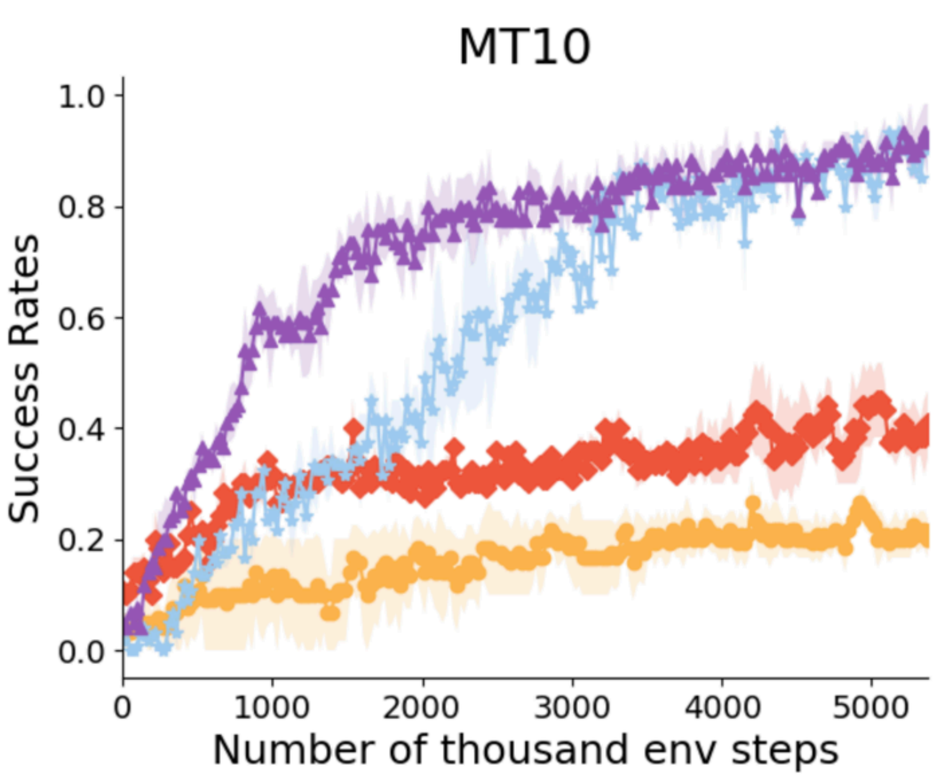
Else:
leave them alone

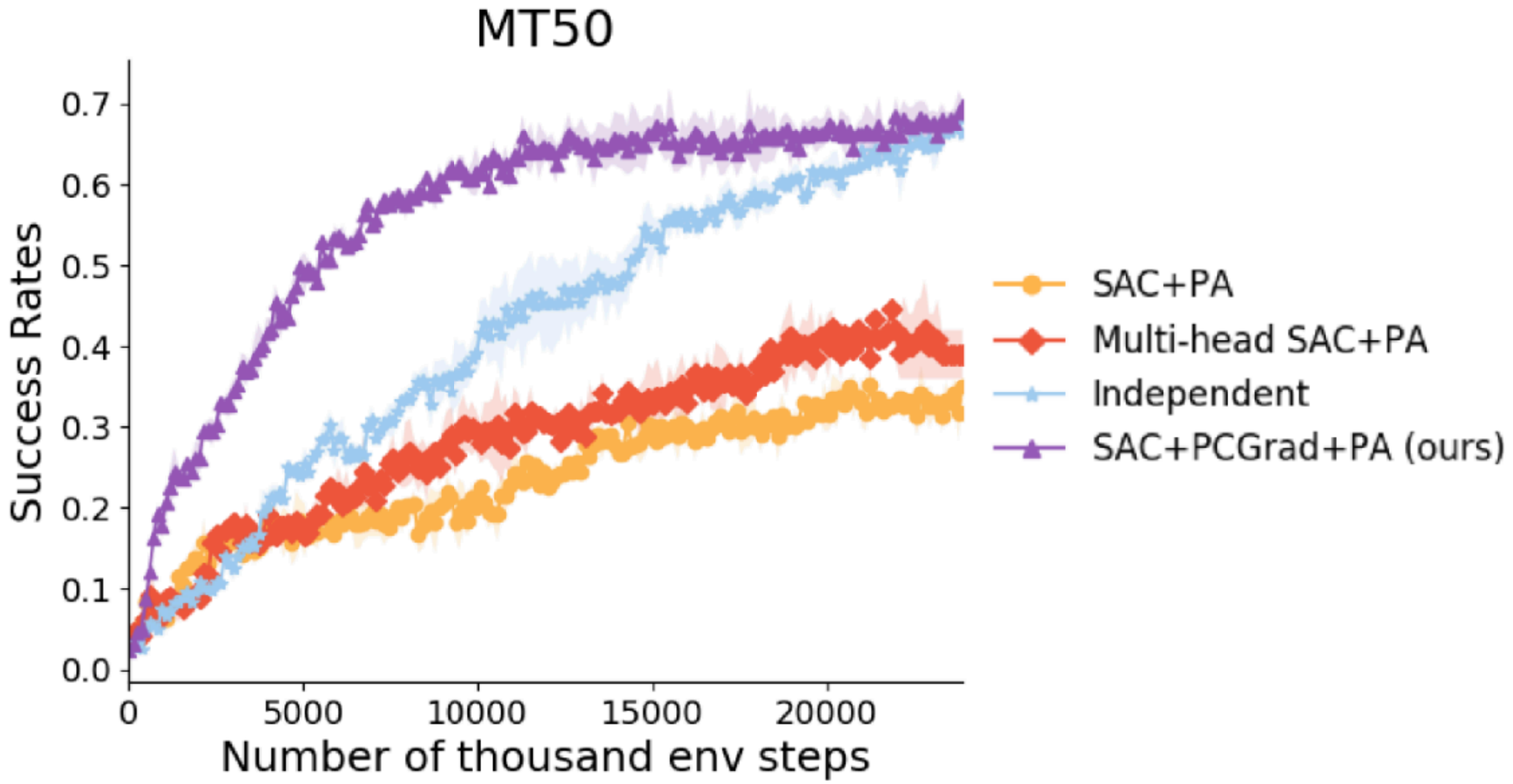


i.e. project conflicting gradients
"PCGrad"

T Yu, S Kumar, A Gupta, S Levine, K Hausman, C Finn. ***Gradient Surgery for Multi-Task Learning***. '19

Multi-Task RL on Meta-World:





Multi-Task CIFAR-100

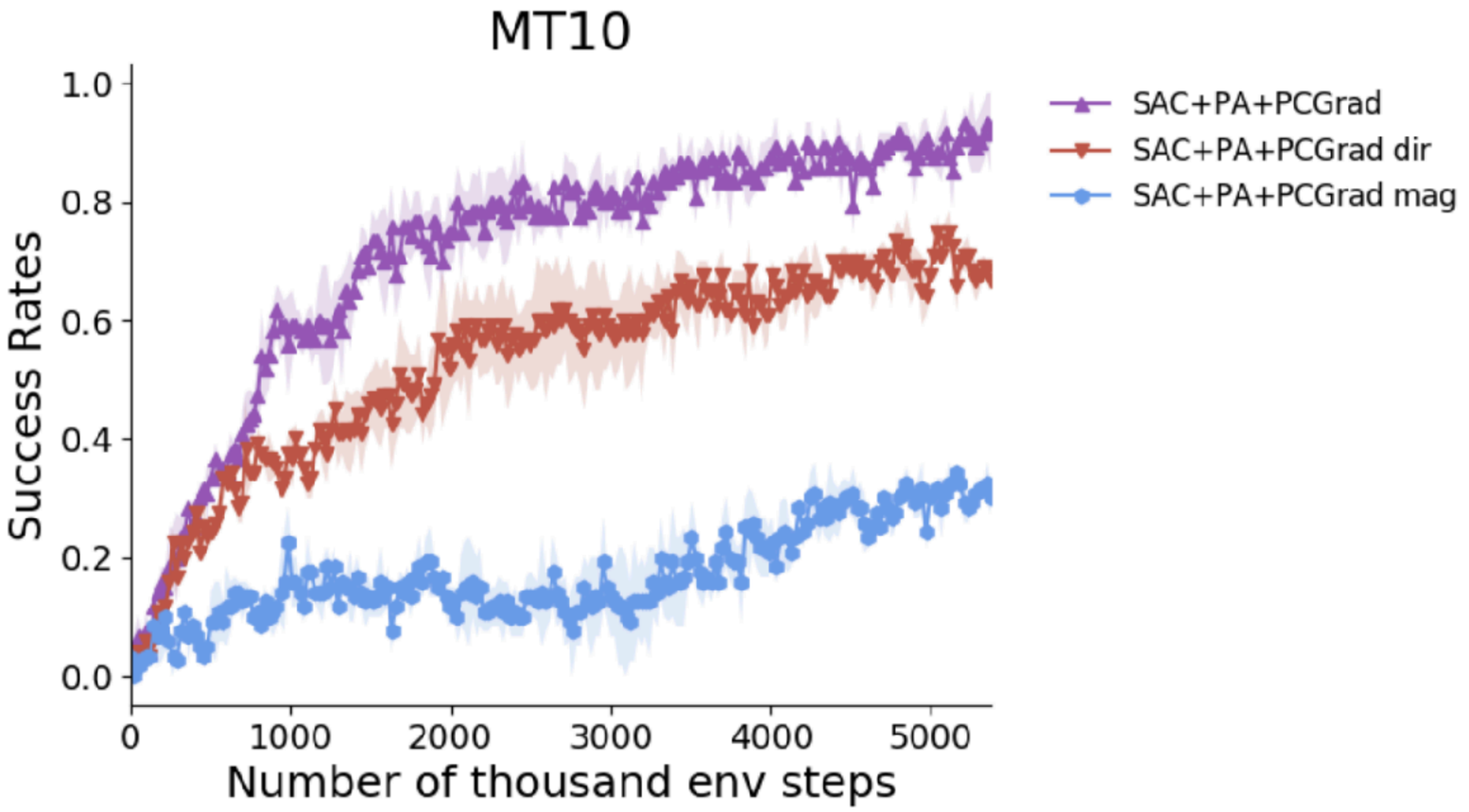
	% accuracy
task specific-1-fc (Rosenbaum et al., 2018)	42
task specific-all-fc (Rosenbaum et al., 2018)	49
cross stitch-all-fc (Misra et al., 2016b)	53
routing-all-fc + WPL (Rosenbaum et al., 2019)	74.7
independent	67.7
PCGrad (ours)	71
routing-all-fc + WPL + PCGrad (ours)	**77.5**

Multi-Task NYUv2

#P.	Architecture	Weighting	Segmentation (Higher Better)		Depth (Lower Better)		Surface Normal: Angle Distance (Lower Better)		Surface Normal: Within t° (Higher Better)		
			mIoU	Pix Acc	Abs Err	Rel Err	Mean	Median	11.25	22.5	30
≈3	Cross-Stitch‡	Equal Weights	14.71	50.23	0.6481	0.2871	33.56	28.58	20.08	40.54	51.97
		Uncert. Weights*	15.69	52.60	0.6277	0.2702	32.69	27.26	21.63	42.84	54.45
		DWA†, $T = 2$	**16.11**	**53.19**	**0.5922**	**0.2611**	**32.34**	**26.91**	**21.81**	**43.14**	**54.92**
1.77	MTAN†	Equal Weights	**17.72**	55.32	**0.5906**	0.2577	31.44	**25.37**	**23.17**	45.65	57.48
		Uncert. Weights*	17.67	**55.61**	0.5927	0.2592	**31.25**	25.57	22.99	**45.83**	**57.67**
		DWA†, $T = 2$	17.15	54.97	0.5956	**0.2569**	31.60	25.46	22.48	44.86	57.24
1.77	MTAN† + PCGrad (ours)	Uncert. Weights*	**20.17**	**56.65**	**0.5904**	**0.2467**	**30.01**	**24.83**	22.28	**46.12**	**58.77**

also helps multi-task **supervised** learning, complementary to **multi-task architectures**

Why does it work so well?



Today: What doesn't work very well?

(and how might we fix it)

What does it take to run multi-task & meta-RL across distinct tasks?

how do we specify the task?

what set of distinct tasks do we train on?

what challenges arise?

Takeaways:

Scaling to broad task distributions is hard,
can't be taken for granted:

- Convey task information beyond reward (e.g. a demo)
- Train on broad, dense task distributions like Meta-World
- Avoid conflicting gradients

Open Challenges in Multi-Task and Meta Learning

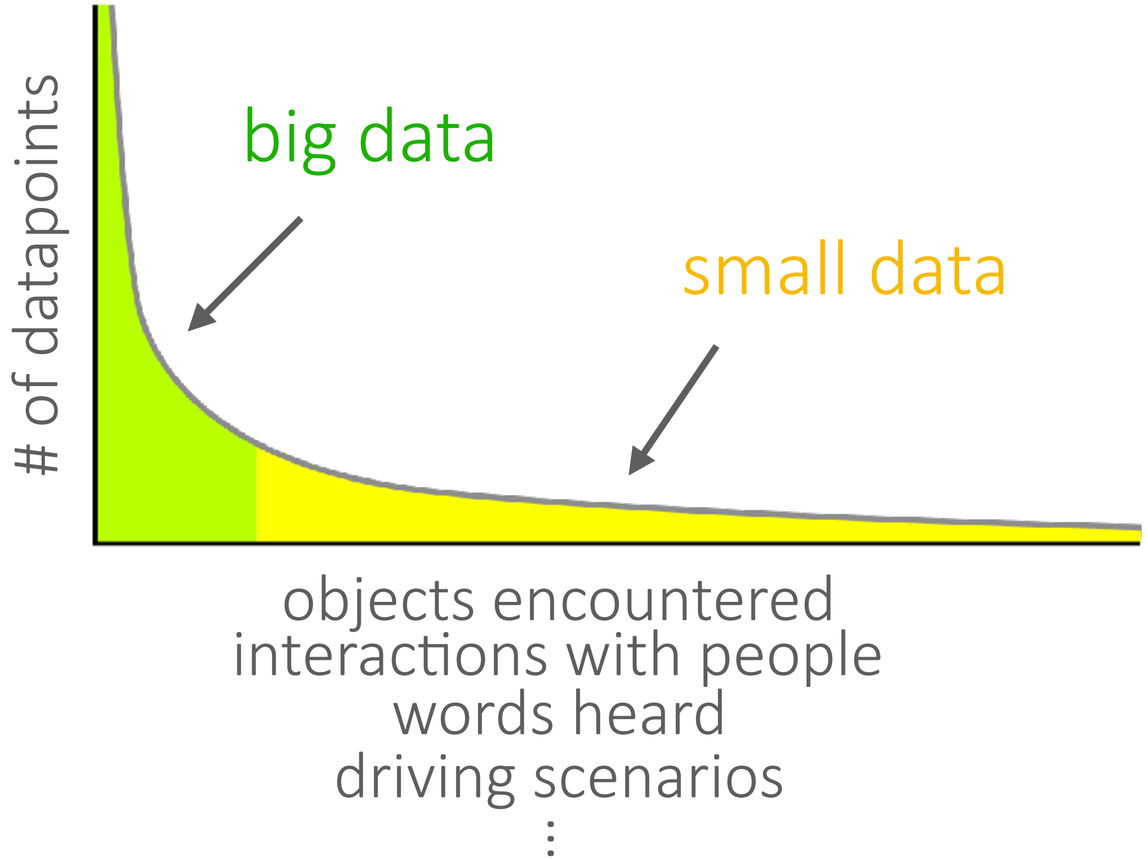
(that we haven't previously covered)

Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions

- **Generalization**: Out-of-distribution tasks, long-tailed task distributions

The problem with long-tailed distributions.



We learned how to do few-shot learning

…but these few-shot tasks are from a different distribution

Some hints might come from **domain adaptation**, **robustness** literature.

Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions

- **Generalization**: Out-of-distribution tasks, long-tailed task distributions
- **Multimodality**: Can you learn priors from multiple modalities of data?

Rich sources of prior experiences.

visual imagery

tactile feedback

language

social cues

Can we learn priors across multiple data modalities?

Varying dimensionalities, units

Carry different, complementary forms of information

Some hints might come from multimodal learning literature.

Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions

- **Generalization**: Out-of-distribution tasks, long-tailed task distributions
- **Multimodality**: Can you learn priors from multiple modalities of data?
- **Algorithm, Model Selection:** When will multi-task learning help you?

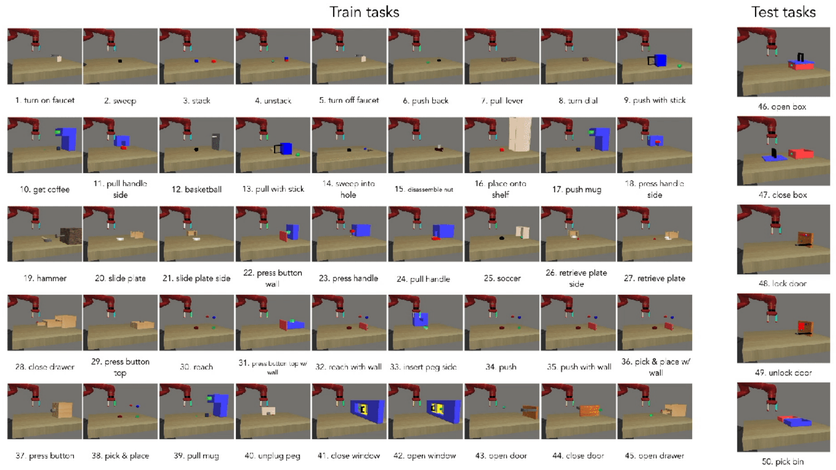
Benchmarks

- **Breadth**: That challenge current algorithms to find common structure
- **Realistic**: That reflect real-world problems

Some steps towards good benchmarks

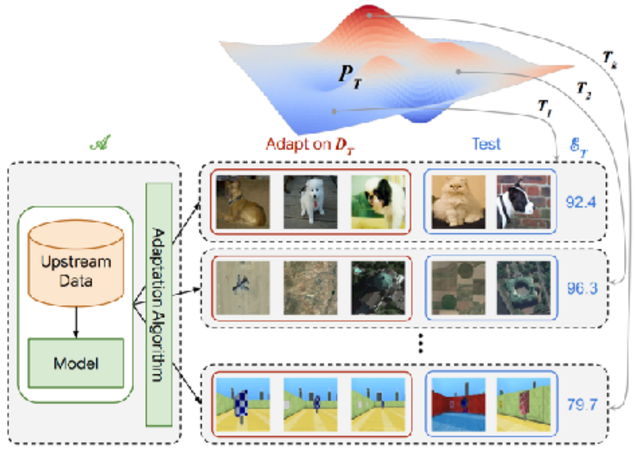
Meta-Dataset

Triantafillou et al.'19



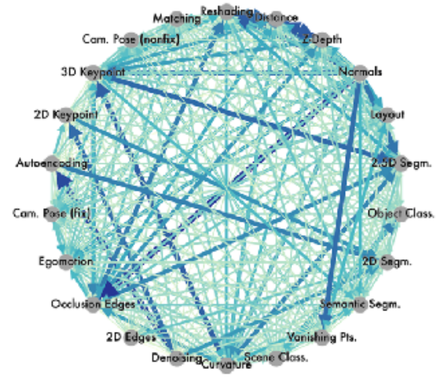
Meta-World Benchmark

Yu et al.'19



Visual Task Adaptation Benchmark

Zhai et al.'19



Taskonomy Dataset

Zamir et al.'18

Goal: reflection of real world problems + appropriate level of difficulty + ease of use

Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions

- **Generalization**: Out-of-distribution tasks, long-tailed task distributions
- **Multimodality**: Can you learn priors from multiple modalities of data?
- **Algorithm, Model Selection:** When will multi-task learning help you?

Benchmarks

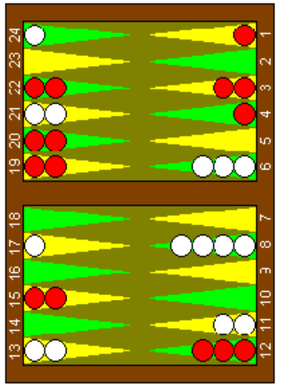
- **Breadth**: That challenge current algorithms to find common structure
- **Realistic**: That reflect real-world problems

Improving core algorithms

- **Computation & Memory**: Making large-scale bi-level optimization practical
- **Theory**: Develop a theoretical understanding of the performance of these algorithms
- **Multi-Step Problems**: Performing tasks in sequence presents challenges.

+ the challenges you discovered in your homework & final projects!

The Bigger Picture



TD Gammon

Watson

helicopter acrobatics

machine translation

DQN

Machines are ***specialists***.

Source: https://youtu.be/8vNxjwt2AqY

A Step Towards Generalists

Some of what we covered in CS330:

- learn **multiple tasks** (multi-task learning)
- leverage **prior experience** when learning new things (meta-learning)
- learn **general-purpose models** (model-based RL)
- **prepare for tasks** before you know what they are (exploration, unsupervised learning)
- perform tasks **in sequence** (hierarchical RL)
- learn **continuously** (lifelong learning)

What's missing?

Logistics

The poster session is tomorrow!

Tuesday 12/2, 1:30-3:30 pm

Print your posters well ahead of time.

Final project report

Due Monday 12/16, midnight.

Welcome to submit earlier.

Hard deadline, because grades due shortly afterward

This is the last lecture!

We'll leave time for course evaluations at the end.