Frontiers and Open-Challenges

CS330
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 $D_{train}$

Classify new examples

test set $X_{test}$

meta-training

$\mathcal{T}_1$

$\mathcal{T}_2$

::

::

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 $\mathcal{D}_{tr}$. Is this bad? When is it bad?
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

\[ y^{ts} = f_{\theta}(D^{tr}_i, x^{ts}) \]

One solution: memorize canonical pose info in \( \theta \) & ignore \( D^{tr}_i \)

Another solution: carry no info about canonical pose in \( \theta \), acquire from \( D^{tr}_i \)

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

\[ y^{ts} = f_\theta(D_i^{tr}, x^{ts}) \]

One solution: memorize canonical pose info in \( \theta \) & ignore \( D_i^{tr} \)

Another solution: carry no info about canonical pose in \( \theta \), acquire from \( D_i^{tr} \)

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 \( \theta \)

\[ \mathcal{L}(\theta, D_{meta-train}) + \beta D_{KL}(q(\theta; \theta_\mu, \theta_\sigma) \| p(\theta)) \]

Places precedence on using information from \( D_i^{tr} \) over \( \theta \).

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

<table>
<thead>
<tr>
<th>Method</th>
<th>NME Omniglot 20-way 1-shot</th>
<th>NME Omniglot 20-way 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td>7.8 (0.2)%</td>
<td>50.7 (22.9)%</td>
</tr>
<tr>
<td>TAML</td>
<td>9.6 (2.3)%</td>
<td>67.9 (2.3)%</td>
</tr>
<tr>
<td>MR-MAML (W) (ours)</td>
<td>83.3 (0.8)%</td>
<td>94.1 (0.1)%</td>
</tr>
</tbody>
</table>

On pose prediction task:

<table>
<thead>
<tr>
<th>Method</th>
<th>MAML</th>
<th>MR-MAML (W) (ours)</th>
<th>CNP</th>
<th>MR-CNP (W) (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>5.39 (1.31)</td>
<td>2.26 (0.09)</td>
<td>8.48 (0.12)</td>
<td>2.89 (0.18)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>CNP</th>
<th>CNP + Weight Decay</th>
<th>CNP + BbB</th>
<th>MR-CNP (W) (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.48 (0.12)</td>
<td>6.86 (0.27)</td>
<td>7.73 (0.82)</td>
<td>2.89 (0.18)</td>
</tr>
</tbody>
</table>

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 $\mathcal{D}_{\text{train}}, \mathbf{x}_{\text{test}}$ → 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

<table>
<thead>
<tr>
<th>method</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML with labels</td>
<td>62.13%</td>
</tr>
<tr>
<td>BiGAN kNN</td>
<td>31.10%</td>
</tr>
<tr>
<td>BiGAN logistic</td>
<td>33.91%</td>
</tr>
<tr>
<td>BiGAN MLP + dropout</td>
<td>29.06%</td>
</tr>
<tr>
<td>BiGAN cluster matching</td>
<td>29.49%</td>
</tr>
<tr>
<td>BiGAN CACTUs MAML</td>
<td>51.28%</td>
</tr>
<tr>
<td>DeepCluster CACTUs MAML</td>
<td><strong>53.97%</strong></td>
</tr>
</tbody>
</table>

**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

- Environment
- Unsupervised Task Acquisition
- Meta-RL
- Meta-learned environment-specific RL algorithm
- Fast Adaptation
- Reward-maximizing policy

Random Task Proposals

- Use randomly initialize discriminators for reward functions
  \[ R(s, z) = \log p_D(z|s) \]
  \( D \rightarrow \) randomly initialized network

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

Random policy – exponential
Random reward – polynomial
Diversity-Driven Proposals

\[ R(s, z) = \log p_D(z|s) \]

- Policy \(\rightarrow\) visit states which are discriminable
- Discriminator \(\rightarrow\) predict skill from state
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.

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*?

\[
\text{meta-train task distribution} = \text{meta-test task distribution}
\]

\[\rightarrow \text{Need broad distribution of tasks for meta-training}\]

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?

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. $D_{train}$ : 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!
Has meta-RL accomplished our goal of making policy adaptation fast? Can we adapt to entirely new tasks?

\[
\text{meta-train task distribution} = \text{meta-test task distribution} \quad \rightarrow \quad \text{Need broad distribution of tasks for meta-training}
\]

\[\wedge\not\text{sparse}\]

\text{WTL trains across task families.}

A few options:

- 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
Results: Meta-learning algorithms seem to struggle…

<table>
<thead>
<tr>
<th>Methods</th>
<th>ML45</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>meta-train</td>
<td>meta-test</td>
</tr>
<tr>
<td>MAML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEARL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

…even on the 45 meta-training tasks!

Multi-task RL algorithms also struggle…

<table>
<thead>
<tr>
<th>Methods</th>
<th>MT50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-task PPO</td>
<td>8.98%</td>
</tr>
<tr>
<td>Multi-task TRPO</td>
<td>22.86%</td>
</tr>
<tr>
<td>Task embeddings</td>
<td>15.31%</td>
</tr>
<tr>
<td>Multi-task SAC</td>
<td>28.83%</td>
</tr>
<tr>
<td>Multi-task multi-head SAC</td>
<td><strong>35.85%</strong></td>
</tr>
</tbody>
</table>

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

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:

MT10

MT50

Success Rates

Number of thousand env steps

Success Rates

Number of thousand env steps

SAC+PA

Multi-head SAC+PA

Independent

SAC+PCGrad+PA (ours)
### Multi-Task CIFAR-100

<table>
<thead>
<tr>
<th>Task</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>task specific-1-fc (Rosenbaum et al., 2018)</td>
<td>42</td>
</tr>
<tr>
<td>task specific-all-fc (Rosenbaum et al., 2018)</td>
<td>49</td>
</tr>
<tr>
<td>cross stitch-all-fc (Misra et al., 2016b)</td>
<td>53</td>
</tr>
<tr>
<td>routing-all-fc + WPL (Rosenbaum et al., 2019)</td>
<td>74.7</td>
</tr>
<tr>
<td>independent</td>
<td>67.7</td>
</tr>
<tr>
<td>PCGrad (ours)</td>
<td>71</td>
</tr>
<tr>
<td>routing-all-fc + WPL + PCGrad (ours)</td>
<td>77.5</td>
</tr>
</tbody>
</table>

### Multi-Task NYUv2

<table>
<thead>
<tr>
<th>#P.</th>
<th>Architecture</th>
<th>Weighting</th>
<th>Segmentation</th>
<th>Depth</th>
<th>Surface Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Higher Better)</td>
<td>(Lower Better)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mIoU</td>
<td>Pix Acc</td>
<td>Angle Distance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Abs Err</td>
<td>Rel Err</td>
<td>(Lower Better) Mean</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Median</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Within $t^\circ$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Higher Better)</td>
</tr>
<tr>
<td>$\approx 3$</td>
<td>Cross-Stitch‡</td>
<td>Equal Weights</td>
<td>14.71, 50.23</td>
<td>0.6481, 0.2871</td>
<td>33.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uncert. Weights*</td>
<td>15.69, 52.60</td>
<td>0.6277, 0.2702</td>
<td>32.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DWA†, $T=2$</td>
<td>16.11, 53.19</td>
<td>0.5922, 0.2611</td>
<td>32.34</td>
</tr>
<tr>
<td>1.77</td>
<td>MTAN†</td>
<td>Equal Weights</td>
<td>17.72, 55.32</td>
<td>0.5906, 0.2577</td>
<td>31.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uncert. Weights*</td>
<td>17.67, 55.61</td>
<td>0.5927, 0.2592</td>
<td>31.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DWA†, $T=2$</td>
<td>17.15, 54.97</td>
<td>0.5956, 0.2569</td>
<td>31.60</td>
</tr>
<tr>
<td>1.77</td>
<td>MTAN† + PCGrad (ours)</td>
<td>Uncert. Weights*</td>
<td>20.17, 56.65</td>
<td>0.5904, 0.2467</td>
<td>30.01</td>
</tr>
</tbody>
</table>

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.

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
- ILSVRC
- Omniglot
- Aircraft
- Birds
- Textures
- Quick Draw
- Fungi
- VGG Flower
- Traffic Signs
- MSCOCO

Meta-World Benchmark
- Triantafillou et al. ‘19
- 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!
Machines are specialists.
Humans are *generalists*.

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.