

# Frontiers and Open-Challenges

CS330

# Logistics

Final project presentations next week  
Schedule on Piazza.

Final project report  
Due next Friday midnight.

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)

## Meta-learning for addressing distribution shift

Capturing equivariances with meta-learning

Adapting to distribution shift

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

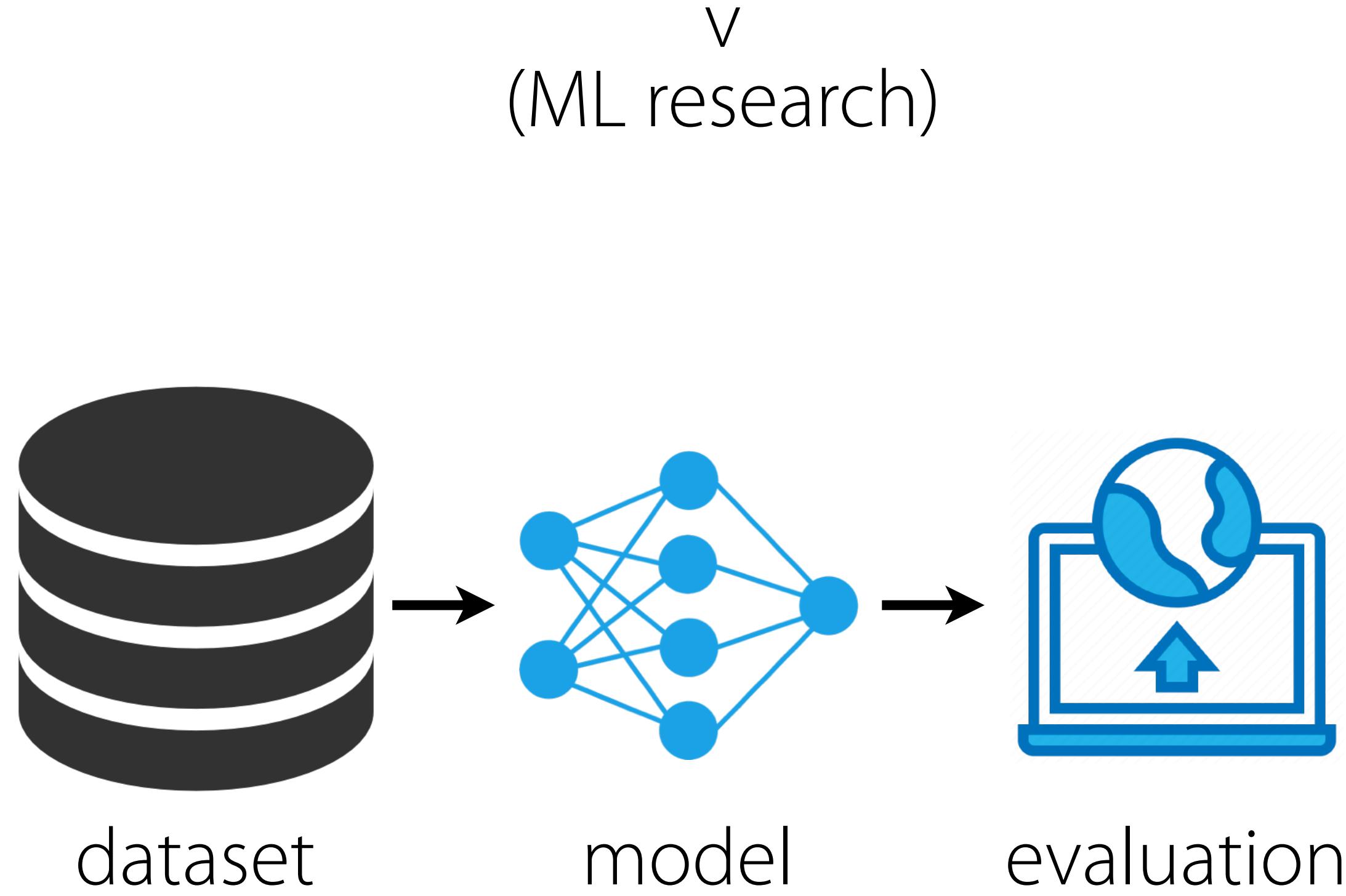
what set of distinct tasks do we train on?

what challenges arise?

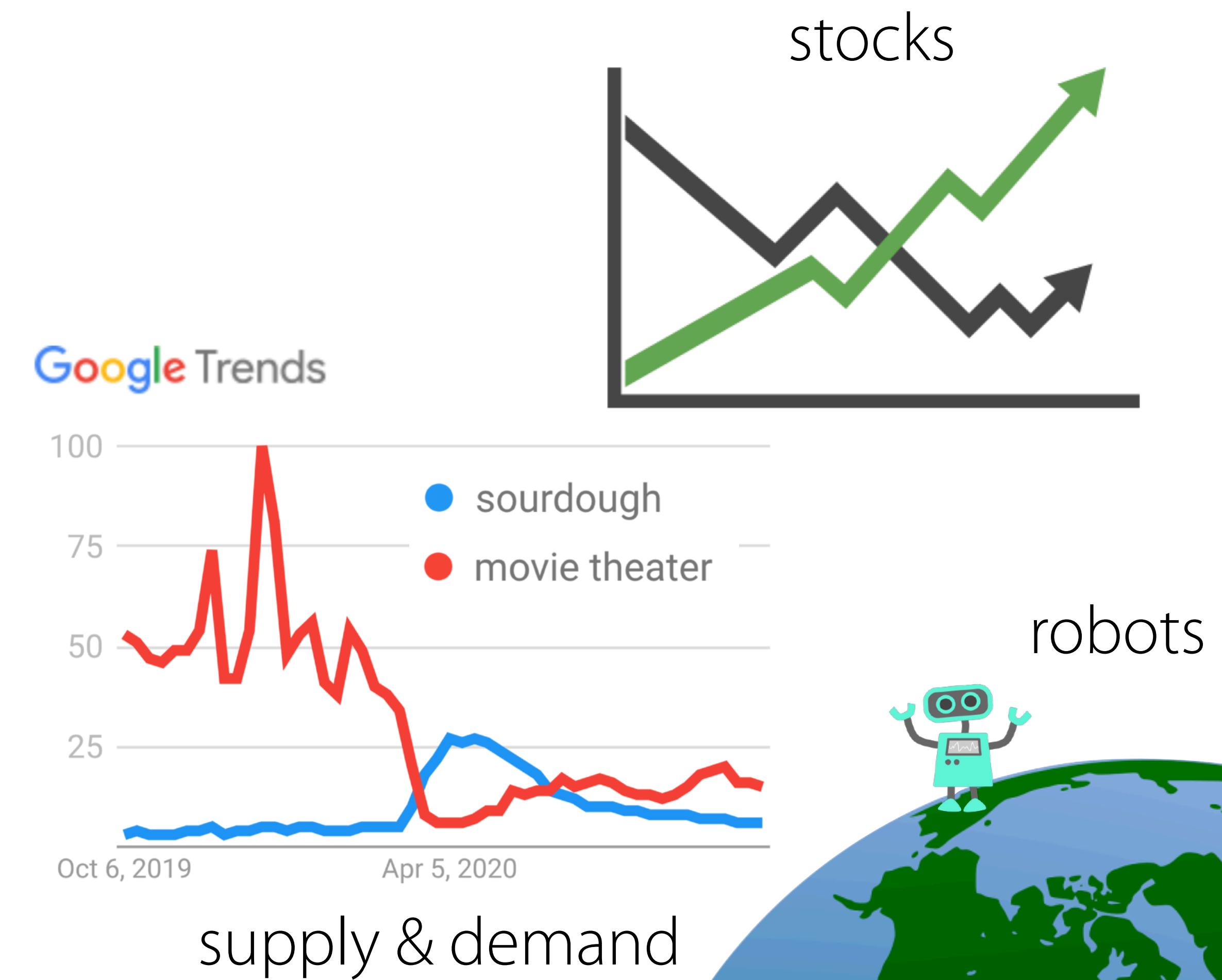
Open Challenges

Why address distribution shift?

# Our current paradigm



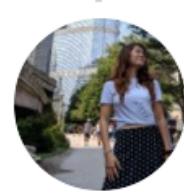
# Our current reality



Can our algorithms handle the **changing** world?

# How does industry cope?

Chip Huyen on misperceptions about ML production:



Chip Huyen  
@chipro

Replies to @chipro

3. If nothing happens, model performance remains the same

ML models perform best right after training. In prod,  
ML systems degrade quickly bc of concept drift.

Tip: train models on data generated 6 months ago &  
test on current data to see how much worse they get.

(4/6)

7:39 AM · Sep 29, 2020 · Twitter Web App



Chip Huyen  
@chipro

Replies to @chipro

4. You won't need to update your models as much

One mindboggling fact about DevOps: Etsy deploys  
50 times/day. Netflix 1000s times/day. AWS every 11.7  
seconds.

MLOps isn't an exemption. For online ML systems, you  
want to update them as fast as humanly possible.

(5/6)

7:40 AM · Sep 29, 2020 · Twitter Web App

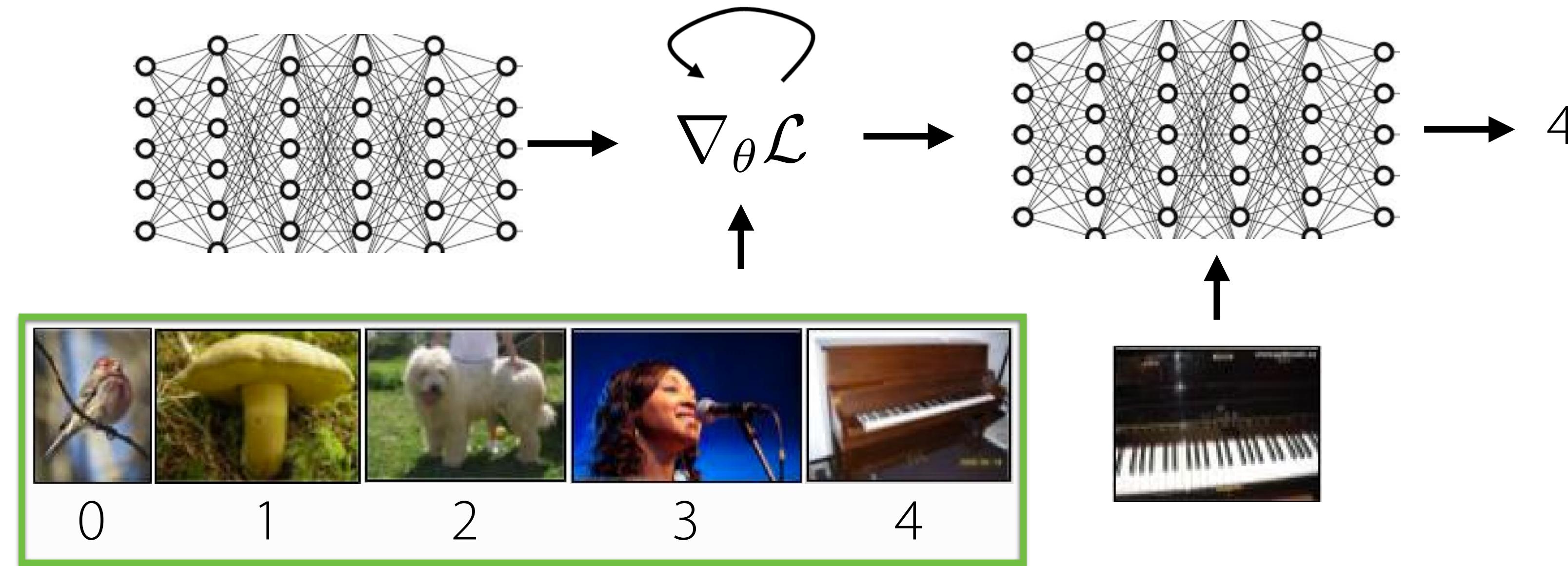
the way our techniques are being *used* != the way we *intend*

**One solution to distribution shift:** build in structure to solve this problem.  
e.g. convolutions

+ Great when we know the structure & how to build it in! — Not great when we don't

Can we discover **equivariant** and **invariant structure** via meta-learning?  
(i.e. **symmetries**)

# Does MAML already do this?



MAML can learn **equivariant** initial features  
but equivariance **may not be preserved** in the gradient update!

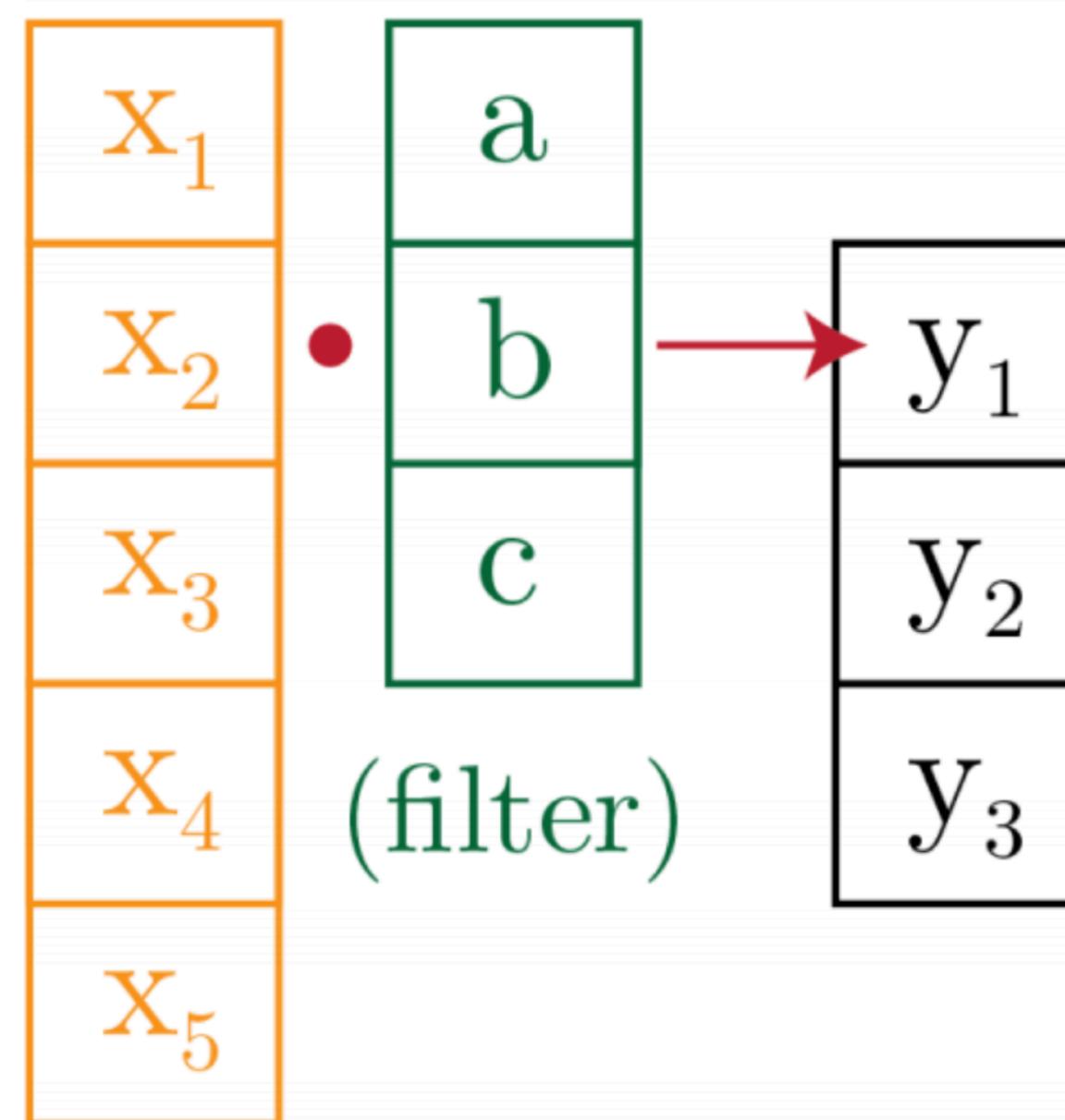
Goal: Can we decompose weights into **equivariant structure** & **corresponding parameters**?

If so: update *only* **parameters** in the inner loop, retaining **equivariance**.

# How are equivariances represented in neural networks?

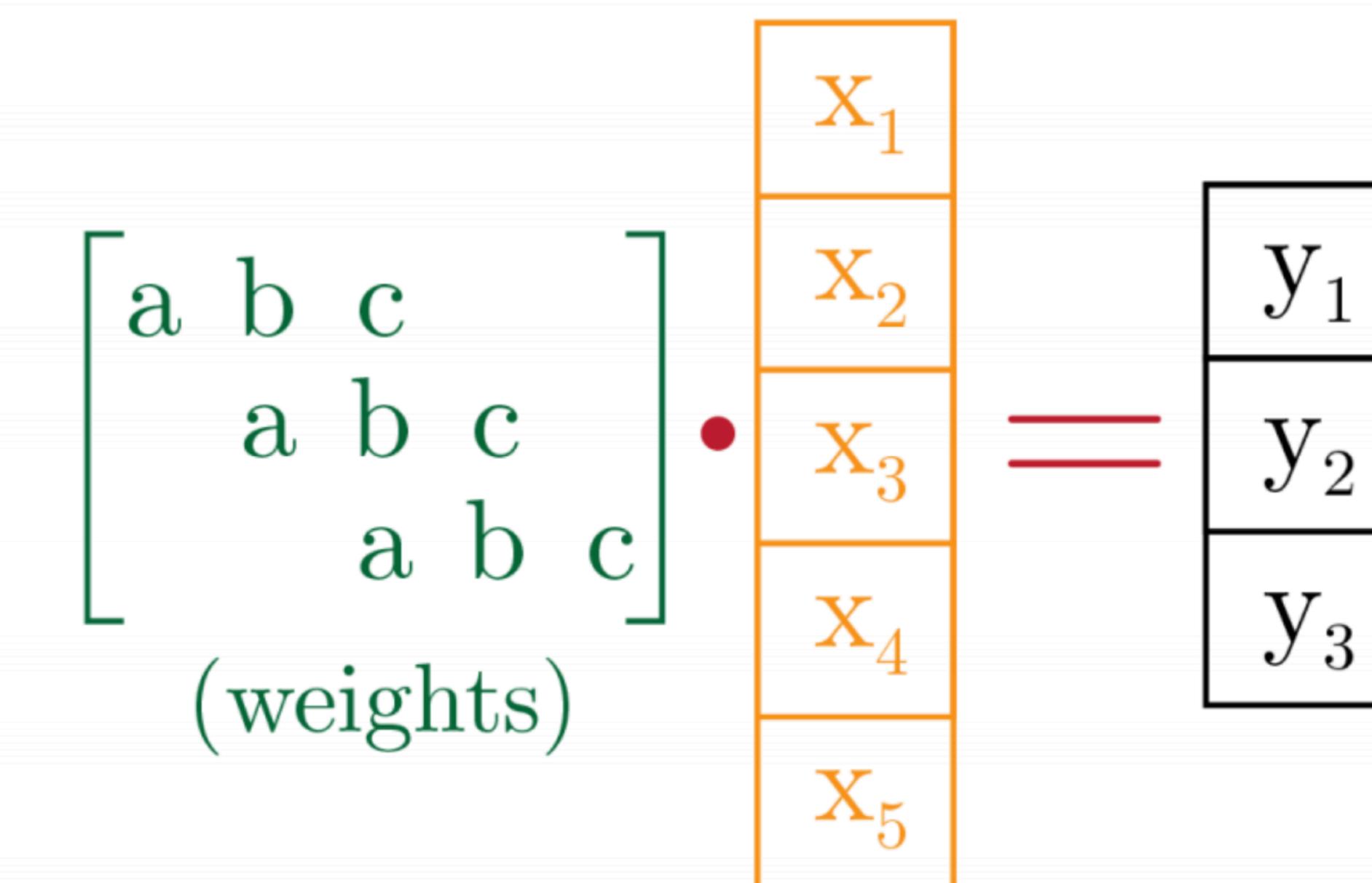
Let's look at an example.

$$\mathbf{x} * \mathbf{w} = \mathbf{y}$$



1D convolution layer

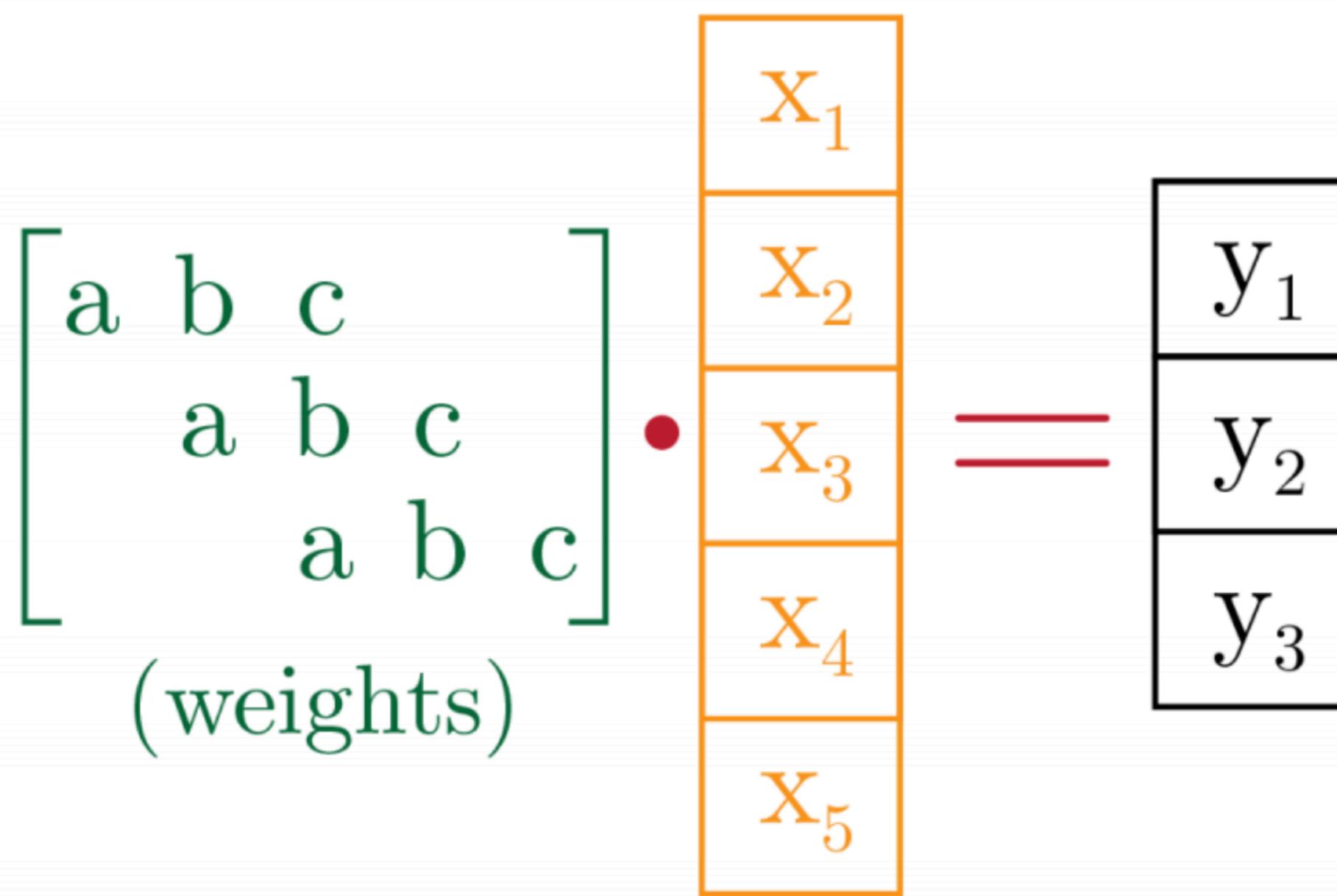
$$\mathbf{W} \bullet \mathbf{x} = \mathbf{y}$$



1D convolution represented as FC layer

# Representing Equivariance by Reparametrization

$$\mathbf{W} \bullet \mathbf{x} = \mathbf{y}$$



1D convolution represented as FC layer

Key idea: reparametrize **weight matrix  $W$**

sharing matrix      underlying filter  
parameters

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ \vdots & & \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \text{vec}(\mathbf{W})$$

$\mathbf{U}$  Captures underlying shared parameters.  
 $\mathbf{v}$  Captures symmetries.

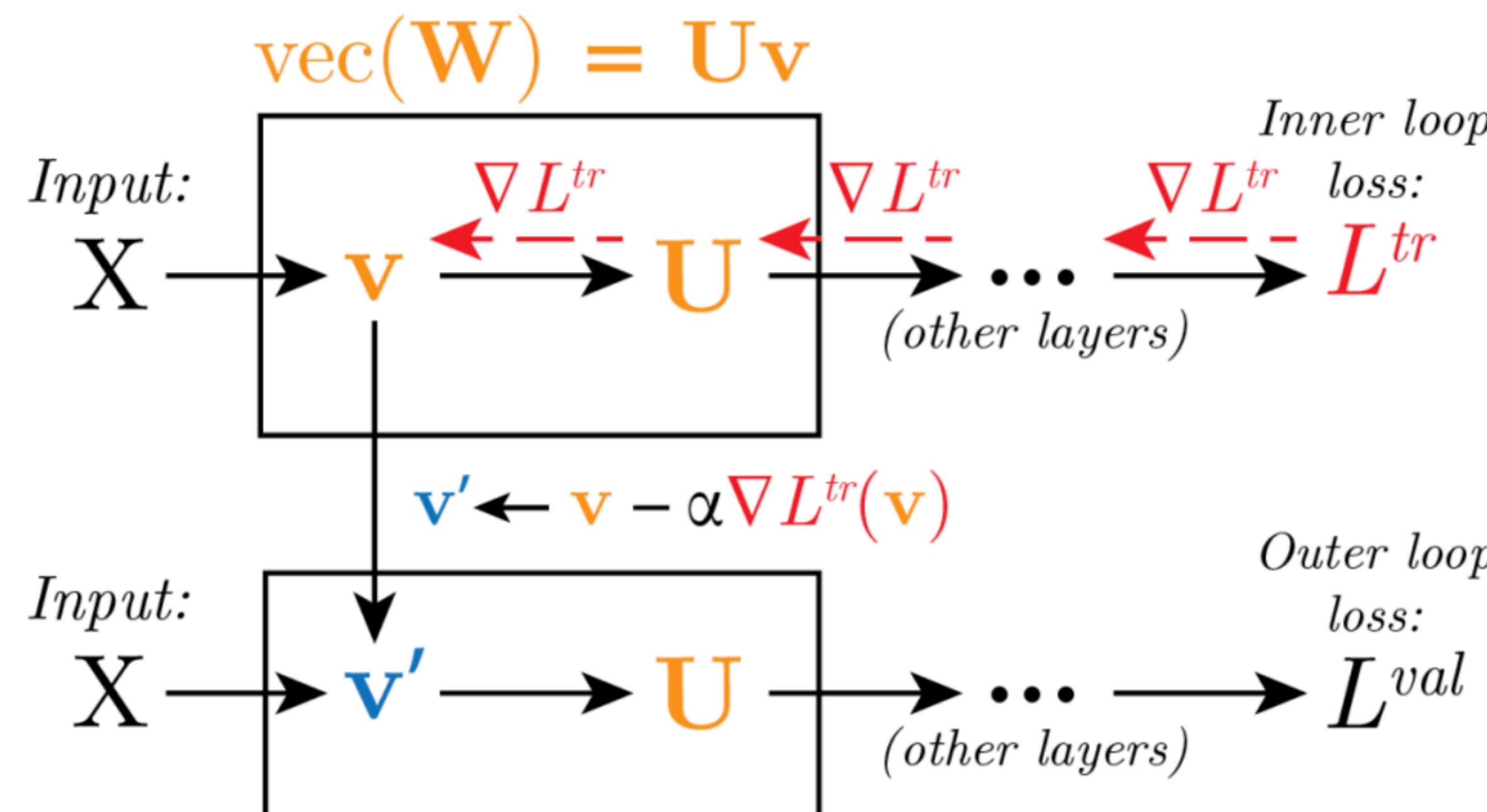
Theoretically, this can directly represent *decoupled equivariant sharing pattern + filter parameters*.

for all G-convolutions with finite group G

# Meta-Learning Equivariance

Inner loop: only update parameters  $v \rightarrow v'$ , keep equivariance  $U$  fixed

Outer loop: learn equivariance  $U$  and initial parameters  $v$



**Important assumption:**  
Some symmetries shared  
by all tasks.

meta-learning symmetries by reparametrization (MSR)

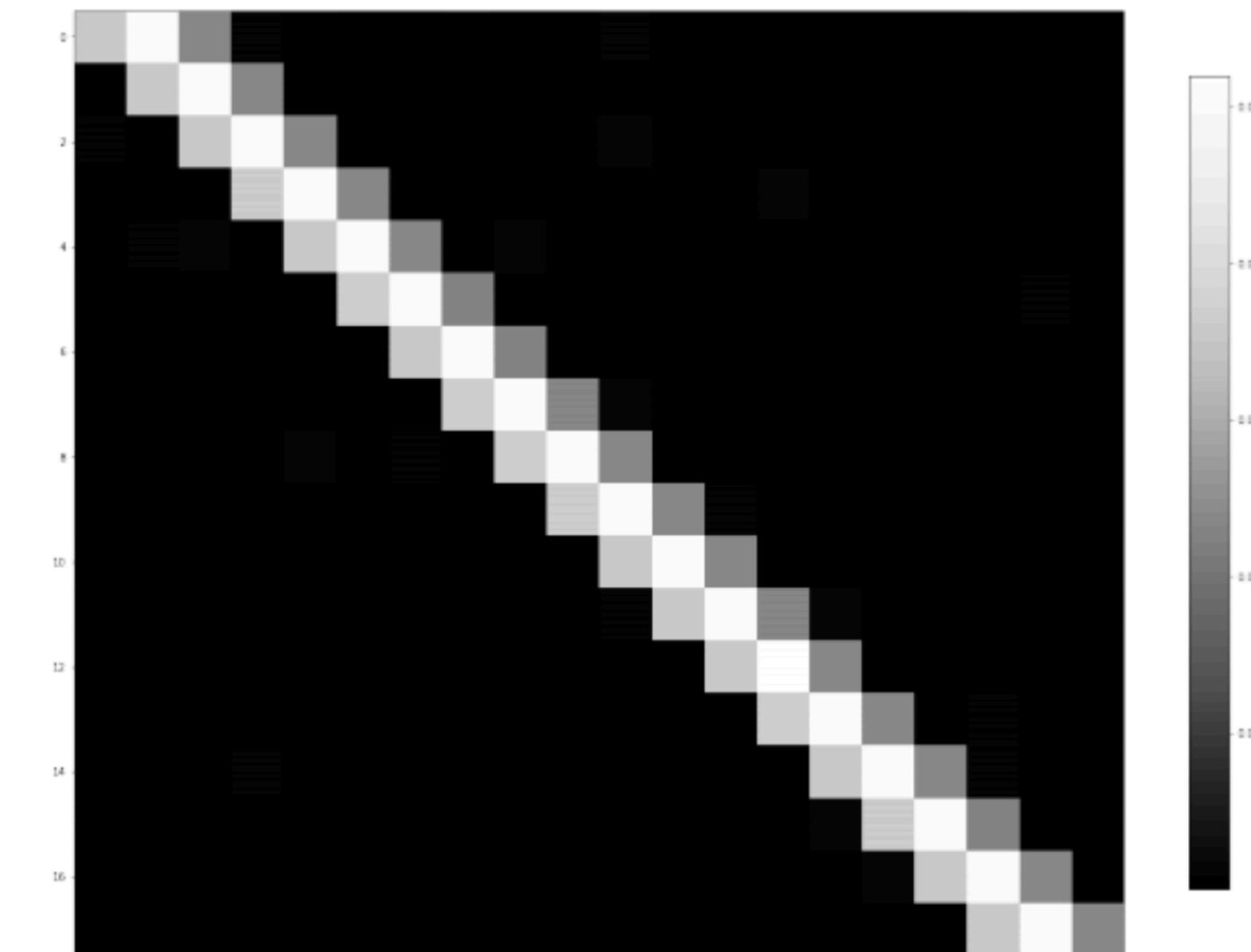
# Can we recover convolutions? from translationally equivariant data

Mean-squared error on held-out test tasks

| Method        | $k = 1$          |
|---------------|------------------|
| MAML-FC       | $3.2 \pm .29$    |
| MAML-LC       | $2.4 \pm .23$    |
| MAML-Conv     | <b>.16 ± .02</b> |
| MSR-FC (Ours) | <b>.18 ± .03</b> |

MAML-X: X corresponds to architecture  
(fully-connected, locally-connected, convolution)

MSR-FC: fully-connected layer weights  $\mathbf{W}$



recovered weight matrix

# Can we recover something better than convolutions?

...from data with *partial* translation symmetry

| Method        | $k = 1$                         | $k = 2$                         | $k = 5$                         |
|---------------|---------------------------------|---------------------------------|---------------------------------|
| MAML-FC       | $3.2 \pm .29$                   | $2.1 \pm .15$                   | $.89 \pm .05$                   |
| MAML-LC       | $2.4 \pm .23$                   | $1.6 \pm .11$                   | $.81 \pm .05$                   |
| MAML-Conv     | <b><math>.16 \pm .02</math></b> | $.52 \pm .05$                   | $.44 \pm .02$                   |
| MSR-FC (Ours) | <b><math>.18 \pm .03</math></b> | <b><math>.21 \pm .02</math></b> | <b><math>.22 \pm .01</math></b> |

$k$ : rank of a locally-connected layer

...from data with translation + rotation + reflection symmetry

| Rotation/Flip Equivariance MSE |             |             |
|--------------------------------|-------------|-------------|
| Method                         | Rot         | Rot+Flip    |
| <b>MSR-Conv (Ours)</b>         | <b>.004</b> | <b>.001</b> |
| MAML-Conv                      | .504        | .507        |

MSR-Conv:  $W$  corresponds to convolution layer weights

# Can we learn symmetries from augmented data?

---

**Algorithm 2:** Augmentation Meta-Training

---

**input**:  $\{\mathcal{T}_i\}_{i=1}^N$ : Meta-training tasks  
**input**: META-TRAIN: Any meta-learner  
**input**: AUGMENT: Data augmenter  
**forall**  $\mathcal{T}_i \in \{\mathcal{T}_i\}_{i=1}^N$  **do**  
     $\{\mathcal{D}_i^{tr}, \mathcal{D}_i^{val}\} \leftarrow \mathcal{T}_i$ ; // task data split  
     $\hat{\mathcal{D}}_i^{val} \leftarrow \text{AUGMENT}(\mathcal{D}^{val})$ ;  
     $\hat{\mathcal{T}}_i \leftarrow \{\mathcal{D}^{tr}, \hat{\mathcal{D}}_i^{val}\}$   
**META-TRAIN** $\left(\{\hat{\mathcal{T}}_i\}_{i=1}^N\right)$

---

| Method     | Aug-Omniglot                     |                                  |                                  |                                  | Aug-MiniImagenet                 |                                  |
|------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
|            | 5 way                            |                                  | 1-shot                           | 20 way                           | 5-shot                           | 1-shot                           |
| 1-shot     | 5-shot                           | 1-shot                           | 5-shot                           | 1-shot                           | 5-shot                           |                                  |
| MAML       | 87.3 $\pm$ 0.5                   | 93.6 $\pm$ 0.3                   | 67.0 $\pm$ 0.4                   | 79.9 $\pm$ 0.3                   | 42.5 $\pm$ 1.1                   | 61.5 $\pm$ 1.0                   |
| MAML (Big) | 89.3 $\pm$ 0.4                   | 94.8 $\pm$ 0.3                   | 69.6 $\pm$ 0.4                   | 83.2 $\pm$ 0.3                   | 37.2 $\pm$ 1.1                   | 63.2 $\pm$ 1.0                   |
| ANIL       | 86.4 $\pm$ 0.5                   | 93.2 $\pm$ 0.3                   | 67.5 $\pm$ 3.5                   | 79.8 $\pm$ 0.3                   | 43.0 $\pm$ 1.1                   | 62.3 $\pm$ 1.0                   |
| ProtoNets  | 92.9 $\pm$ 0.4                   | <b>97.4 <math>\pm</math> 0.2</b> | <b>85.1 <math>\pm</math> 0.3</b> | <b>94.3 <math>\pm</math> 0.2</b> | 34.6 $\pm$ 0.5                   | 54.5 $\pm$ 0.6                   |
| MSR (Ours) | <b>95.3 <math>\pm</math> 0.3</b> | <b>97.7 <math>\pm</math> 0.2</b> | 84.3 $\pm$ 0.2                   | 92.6 $\pm$ 0.2                   | <b>45.5 <math>\pm</math> 1.1</b> | <b>65.2 <math>\pm</math> 1.0</b> |

MSR provides a framework for understanding the  
interplay of **features** & **structure** in meta-learning

# Today: What doesn't work very well?

(and how might we fix it)

## Meta-learning for addressing distribution shift

Capturing equivariances with meta-learning

Adapting to distribution shift

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

what set of distinct tasks do we train on?

what challenges arise?

Open Challenges

# What kind of distribution shift to adapt to?

We'll now focus on: *group shift*

**categorical group variable  $z$**   
e.g. user, location, time of day  
(can be derived from meta-data)

Training data from  $p(x, y | z)p_{\text{tr}}(z)$

Test data from  $p(x, y | z)p_{\text{ts}}(z)$

can capture **label shift**, most **covariate shift**  
captures problems like **federated learning**

**Group DRO** (*distributionally robust optimization*):

(Ben-Tal et al. '13, Duchi et al '16)

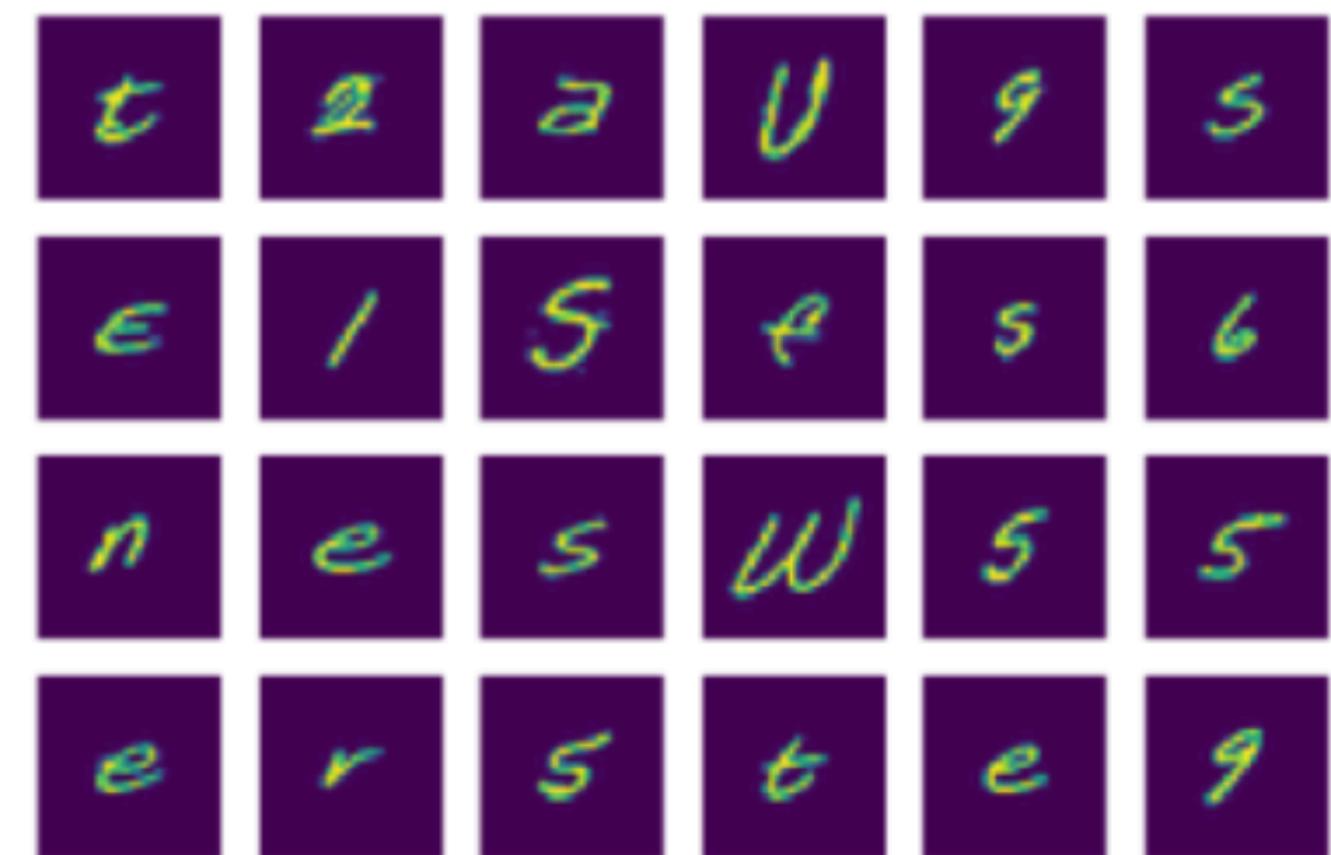
Form adversarial distribution  $q(z)$ :  $\min_{\theta} \sup_{q \in \mathcal{Q}} \mathbb{E}_{q_z} [\mathbb{E}_{p_{xy|z}} [\ell(g(x; \theta), y)]]$

+ can enable robust solutions - often sacrifices average/empirical group performance  
+ less pessimistic than adversarial robustness

# Can we aim to *adapt* instead of aiming for robustness?

## Test time

unlabeled data from test sub-distribution  
(e.g. new user, different time-of-day, new place)



adapt model & infer labels



**Assumption:** test inputs from one group available in a batch or streaming.

## Adaptive risk minimization (ARM)

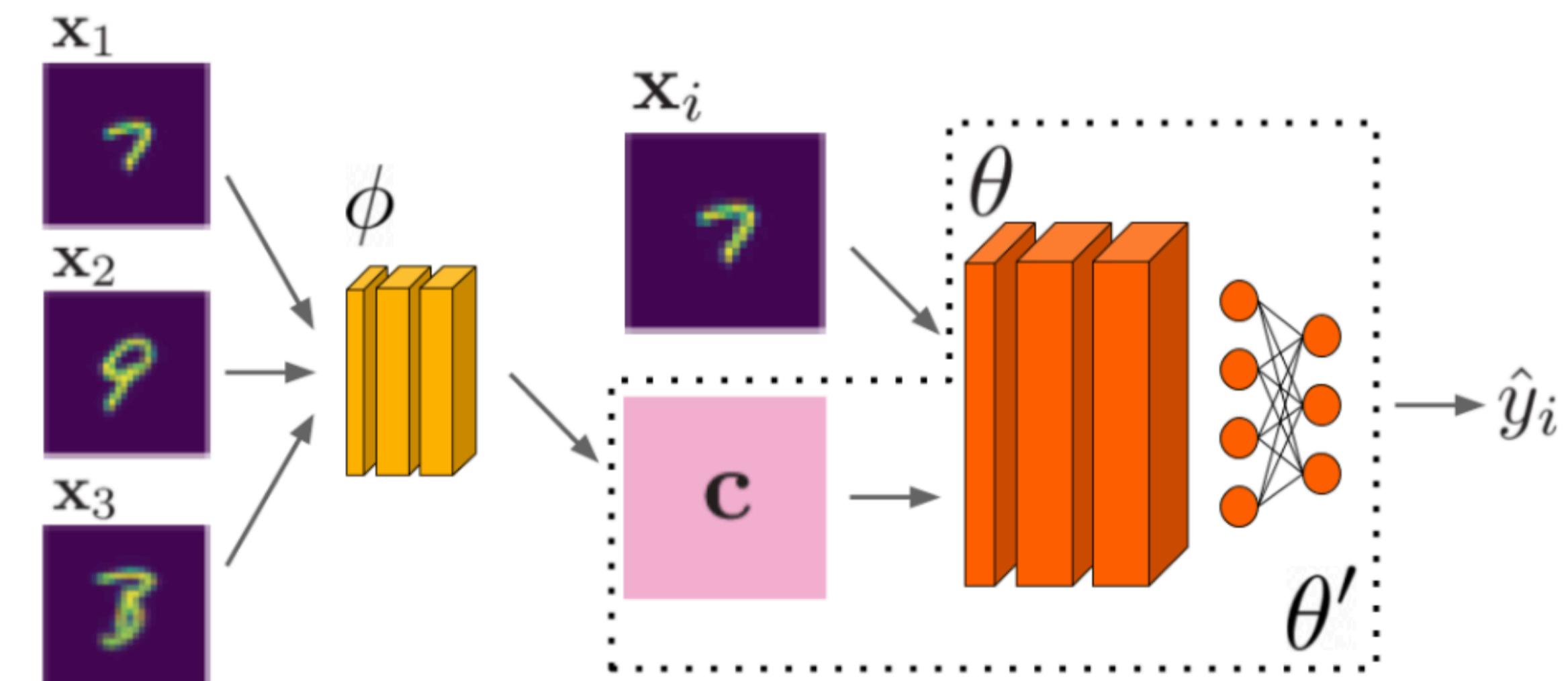
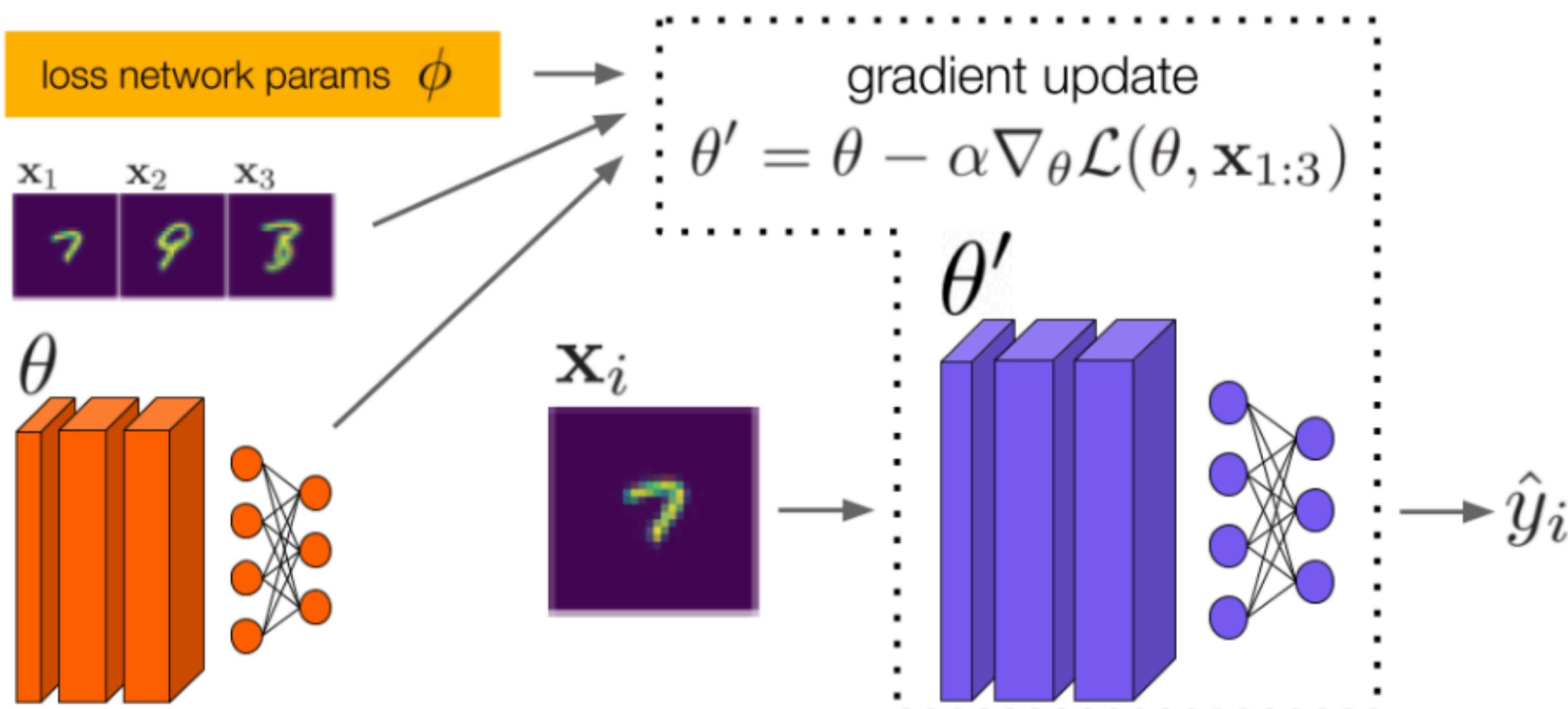
## Adaptive risk minimization (ARM)

1. Construct sub-distributions of training data
2. Train for adaptation to sub-distributions.

How to adapt with unlabeled data?

MAML with learned loss

or meta-learning with context variable



Simplest setting: context = BN statistics

# Experimental Comparisons

**ERM** - standard deep network training

**DRNN** - distributional robustness  
(Sagawa, Koh et al. ICLR '20)

**UW** - ERM but upweight groups to  
the uniform distribution

**ARM** - adaptive risk minimization

**ARM-CML** - adapt with context variable

**ARM-BN** - adapt using batch norm stats

**ARM-LL** - adapt with learned loss

# Experiment 1. Federated Extended MNIST (Cohen et al. 2017, Caldas et al. 2019)

**Distribution shift:** adapt to new users with only unlabeled data

| FEMNIST               |                                  |                                  |
|-----------------------|----------------------------------|----------------------------------|
| Method                | WC                               | Avg                              |
| ERM                   | $62.9 \pm 1.9$                   | $80.1 \pm 0.9$                   |
| UW*                   | $61.8 \pm 0.9$                   | $80.1 \pm 0.3$                   |
| DRNN                  | $58.1 \pm 0.7$                   | $74.4 \pm 0.8$                   |
| <i>q</i> -FedAvg [37] | $58.2 \pm 1.0$                   | $80.8 \pm 0.3$                   |
| ARM-CML               | $67.8 \pm 1.3$                   | <b><math>85.7 \pm 0.3</math></b> |
| ARM-BN                | <b><math>72.6 \pm 0.3</math></b> | <b><math>85.7 \pm 0.1</math></b> |
| ARM-LL                | $69.6 \pm 2.1$                   | <b><math>85.6 \pm 0.5</math></b> |

+ 5% improvement in average accuracy  
+ 10% improvement in worst-case accuracy

**ARM** - adaptive risk minimization

**DRNN** - distributional robustness  
(Sagawa, Koh et al. ICLR '20)

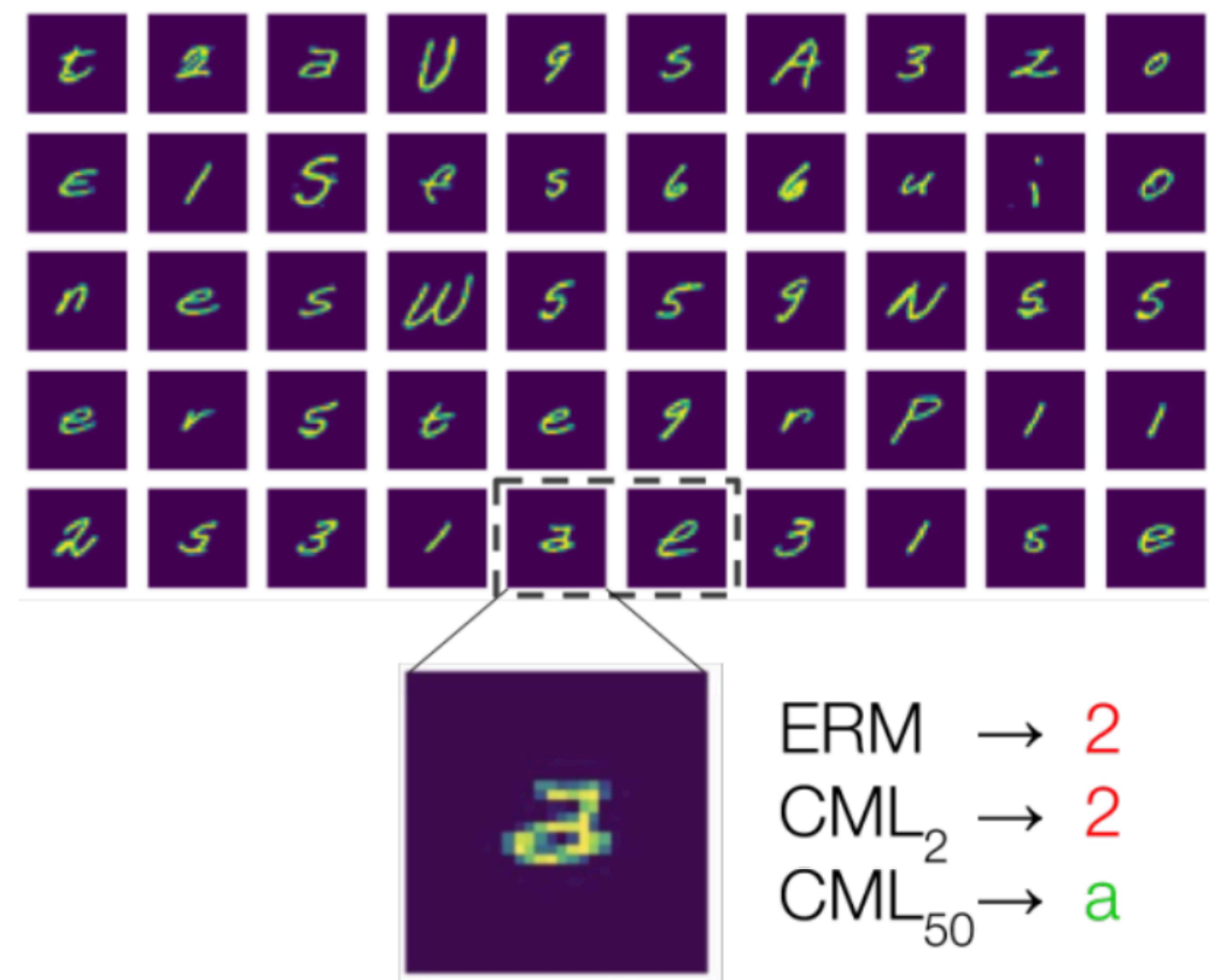
***q*-FedAvg** (Li et al. 2020) - federated learning method

**ERM** - standard deep network training

**UW** - ERM but upweight groups to the uniform distribution

# Experiment 1. Federated Extended MNIST (Cohen et al. 2017, Caldas et al. 2019)

**Distribution shift:** adapt to new users with only unlabeled data



## Experiment 2. CIFAR-C, TinyImageNet-C (Hendrycks & Dietterich, 2019)

**Distribution shift:** adapt to new image corruptions

(train using 56 corruptions, test using 22 disjoint corruptions)

| <b>Method</b> | <b>CIFAR-10-C</b> |                   | <b>Tiny ImageNet-C</b> |                   |
|---------------|-------------------|-------------------|------------------------|-------------------|
|               | <b>WC</b>         | <b>Avg</b>        | <b>WC</b>              | <b>Avg</b>        |
| ERM           | 49.6 ± 0.1        | 69.8 ± 0.4        | 19.3 ± 0.5             | 41.4 ± 0.2        |
| UW*           | —                 | —                 | —                      | —                 |
| DRNN          | 44.5 ± 0.5        | 70.7 ± 0.6        | 19.9 ± 0.3             | 41.6 ± 0.2        |
| ARM-CML       | 67.7 ± 0.5        | 79.2 ± 0.3        | 21.4 ± 0.2             | 43.3 ± 0.4        |
| ARM-BN        | <b>71.1 ± 0.1</b> | <b>80.9 ± 0.2</b> | <b>27.7 ± 0.2</b>      | <b>44.9 ± 0.2</b> |
| ARM-LL        | 66.9 ± 0.2        | 75.7 ± 0.3        | 27.1 ± 0.3             | 44.2 ± 0.4        |

+ 3-10% improvement in average accuracy

+ 8-21% improvement in worst-case accuracy

**ARM** - adaptive risk minimization

**DRNN** - distributional robustness  
(Sagawa, Koh et al. ICLR '20)

**ERM** - standard deep network training

**UW** - ERM but upweight groups to the uniform distribution

# Today: What doesn't work very well?

(and how might we fix it)

Meta-learning for addressing distribution shift

Capturing equivariances with meta-learning

Adapting to distribution shift

## *Takeaways*

Preliminary evidence that meta-learning can capture equivariances

via reparametrized weight matrices

Allow adaptation / fine-tuning without labeled target data

via adaptive risk minimization

# Today: What doesn't work very well?

(and how might we fix it)

Meta-learning for addressing distribution shift

Capturing equivariances with meta-learning

Adapting to distribution shift

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

what set of distinct tasks do we train on?

what challenges arise?

Open Challenges

Have MAML, RL<sup>2</sup>, PEARL, DREAM accomplished our goal of making policy adaptation fast?

Sort of...

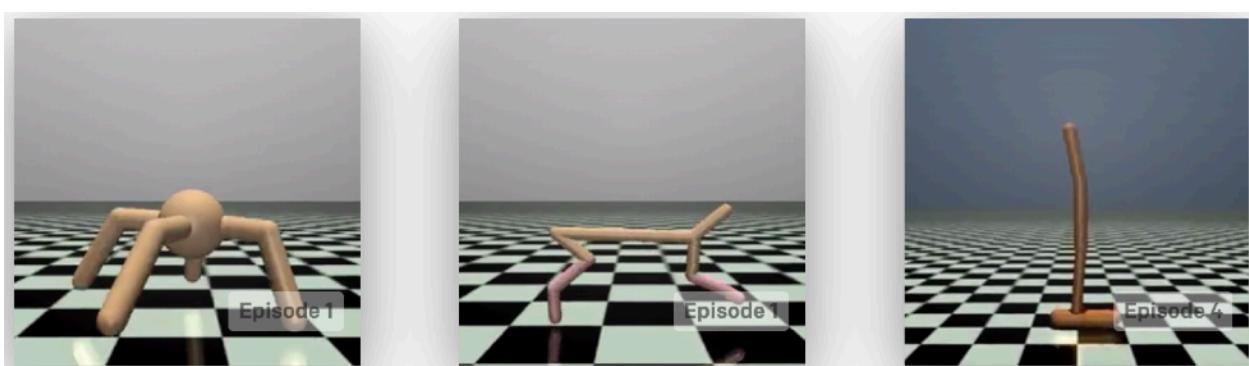
Can we adapt to *entirely new tasks*?

meta-train task  
distribution

=  
meta-test task  
distribution

& not sparse  
↗  
→ Need broad distribution of tasks  
for meta-training

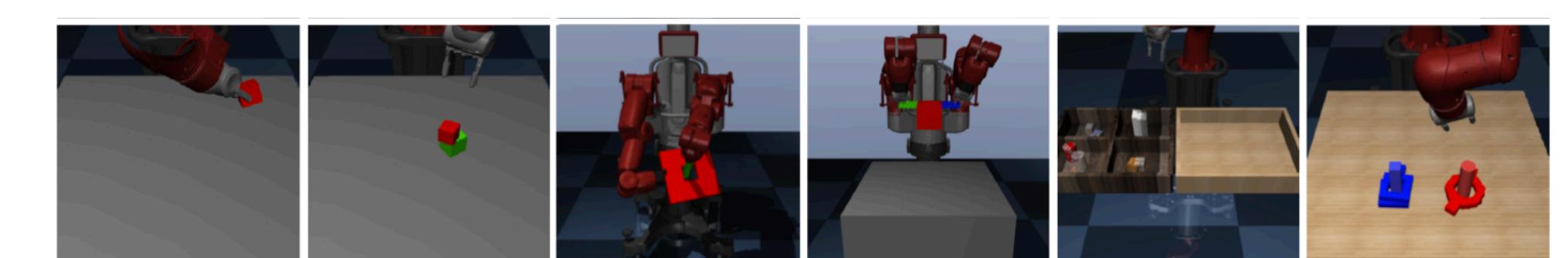
A few options:



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

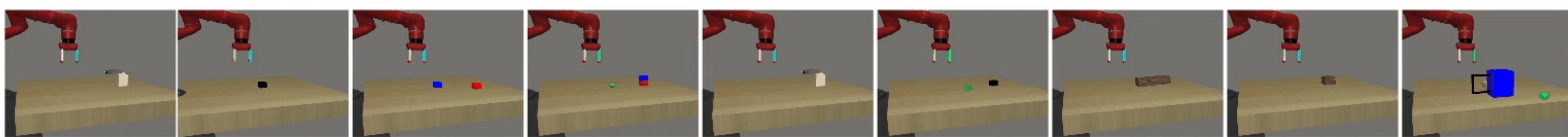
50+ qualitatively distinct tasks

shaped reward function &  
success metrics

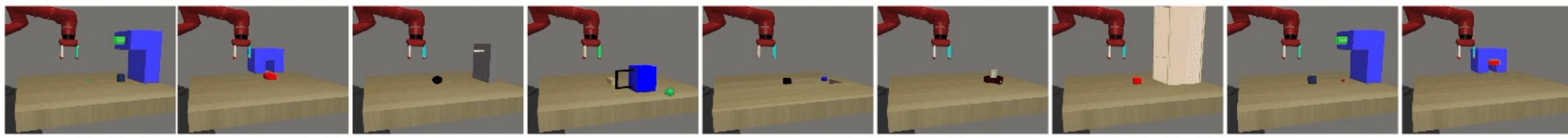
All tasks individually solvable  
(to allow us to focus on multi-  
task / meta-RL component)

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

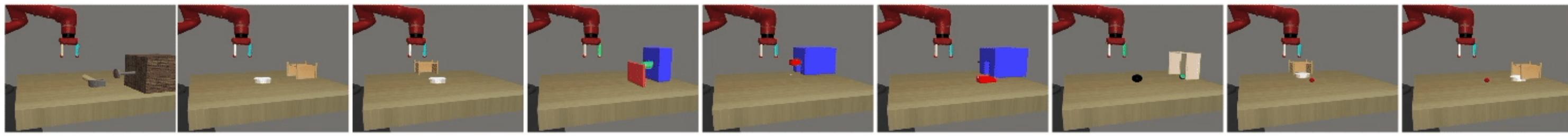
Train tasks



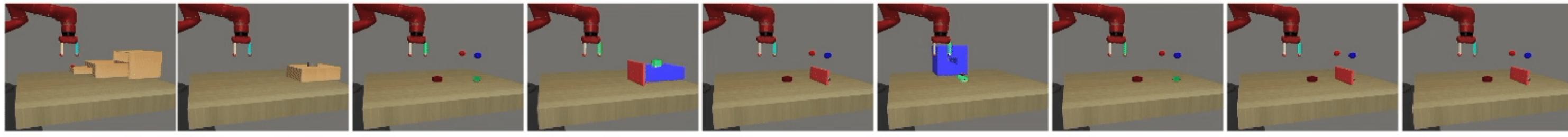
1. turn on faucet    2. sweep    3. stack    4. unstack    5. turn off faucet    6. push back    7. pull lever    8. turn dial    9. push with stick



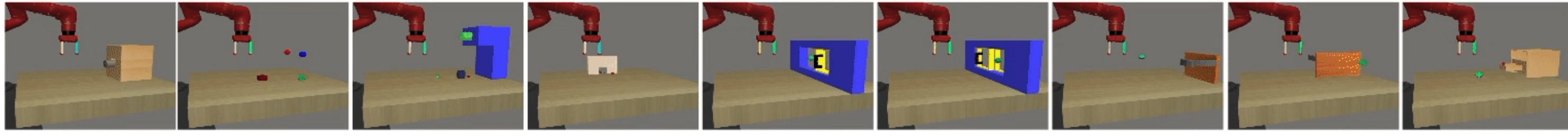
10. get coffee    11. pull handle side    12. basketball    13. pull with stick    14. sweep into hole    15. disassemble nut    16. place onto shelf    17. push mug    18. press handle side



19. hammer    20. slide plate    21. slide plate side    22. press button wall    23. press handle    24. pull handle    25. soccer    26. retrieve plate side    27. retrieve plate

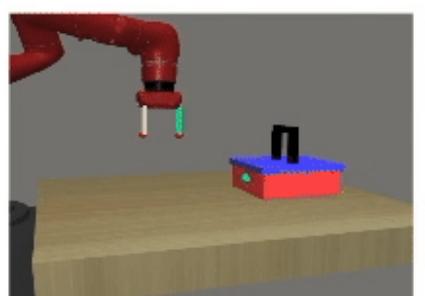


28. close drawer    29. press button top    30. reach    31. press button top w/ wall    32. reach with wall    33. insert peg side    34. push    35. push with wall    36. pick & place w/ wall

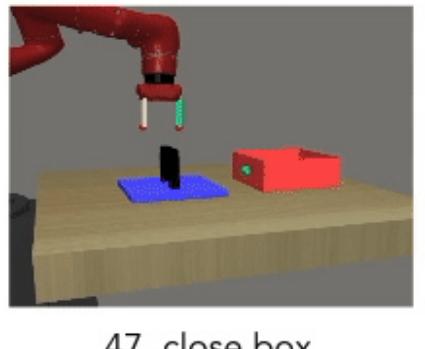


37. press button    38. pick & place    39. pull mug    40. unplug peg    41. close window    42. open window    43. open door    44. close door    45. open drawer

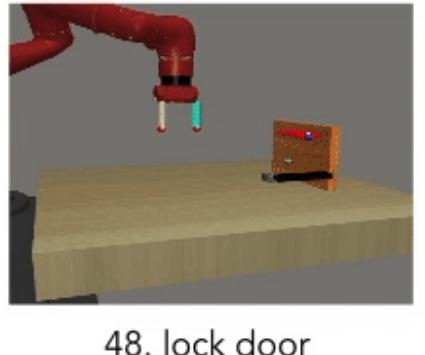
Test tasks



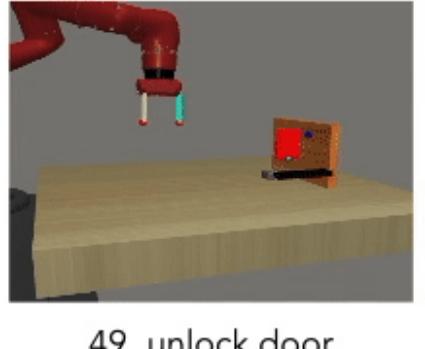
46. open box



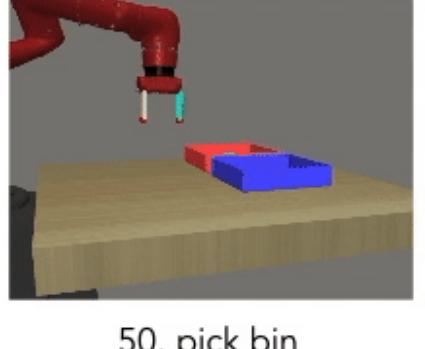
47. close box



48. lock door



49. unlock door



50. pick bin

## Meta-World Benchmark

Results: Meta-learning algorithms seem to struggle...

| Methods         | ML45       |           |
|-----------------|------------|-----------|
|                 | meta-train | meta-test |
| MAML            |            |           |
| RL <sup>2</sup> |            |           |
| 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 | <b>35.85%</b> |

# Why the poor results?

Exploration challenge?

All tasks individually solvable.

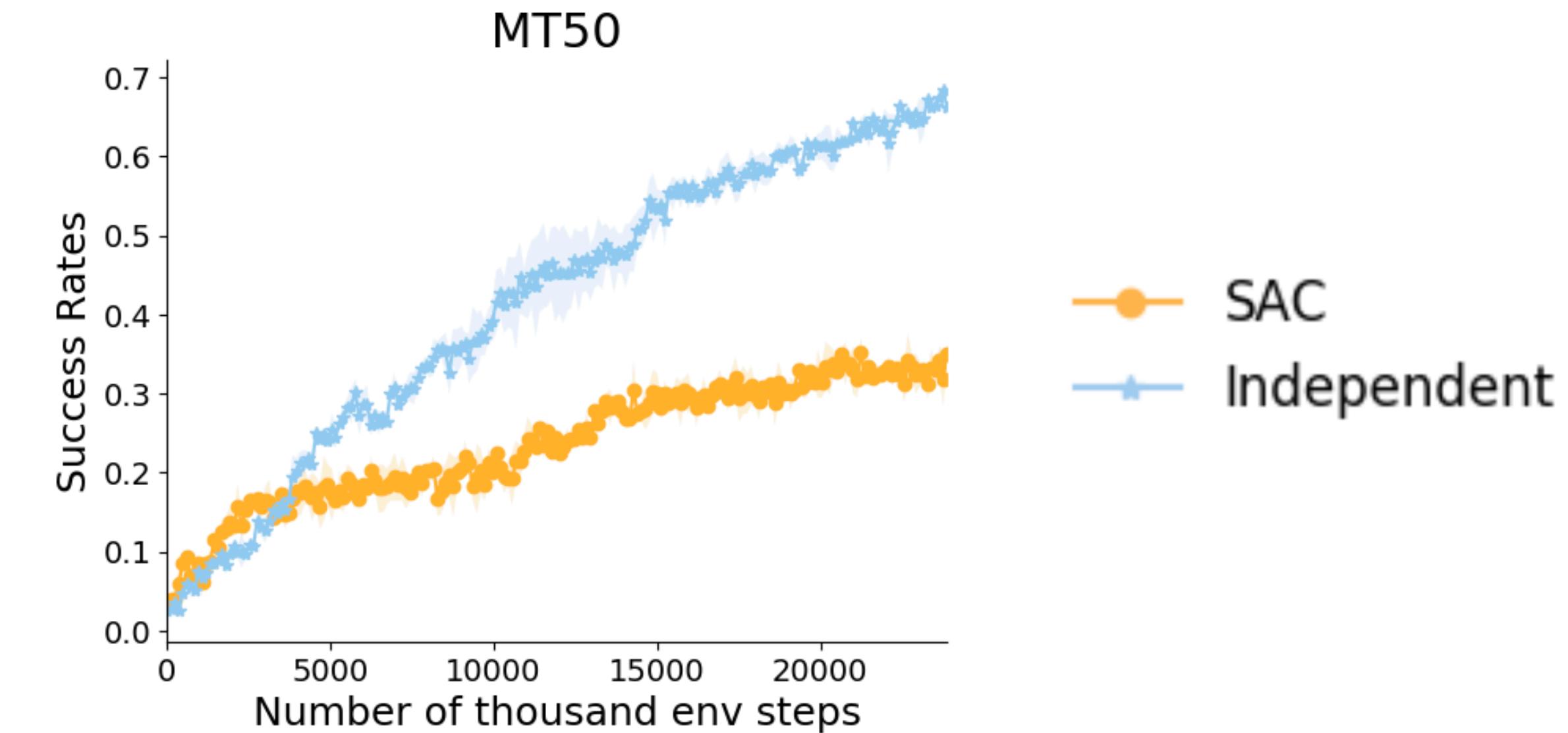
Data scarcity?

All methods given budget with plenty of samples.

Limited model capacity?

All methods plenty of capacity.

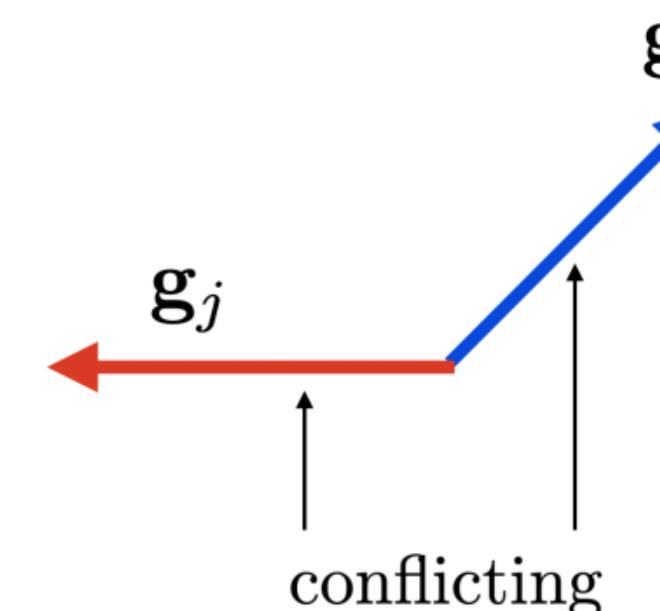
Training models *independently* performs the best.



Our conclusion: must be a multi-task *optimization* challenge.

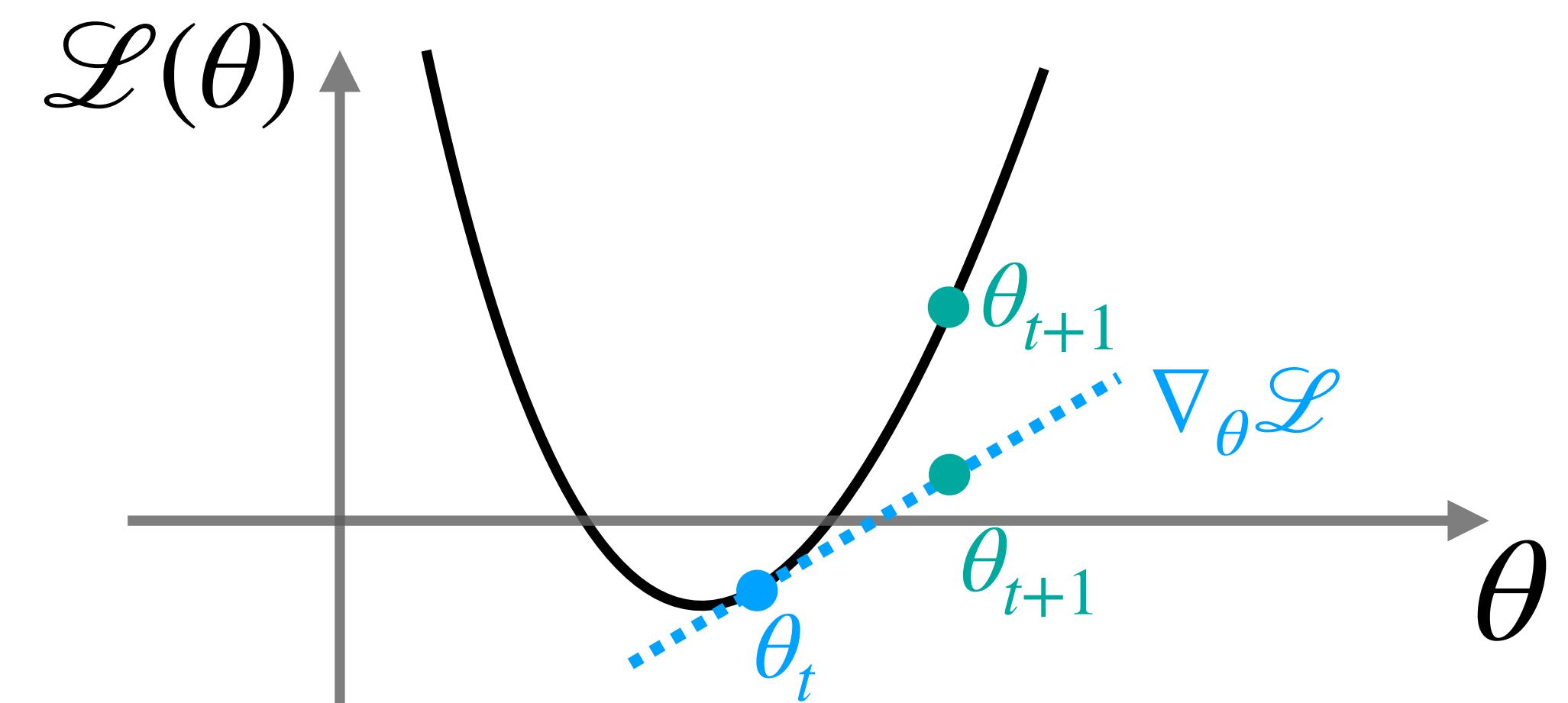
## Hypothesis 1: Gradients from different tasks often conflict

If so: would see negative inner product of gradients.



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

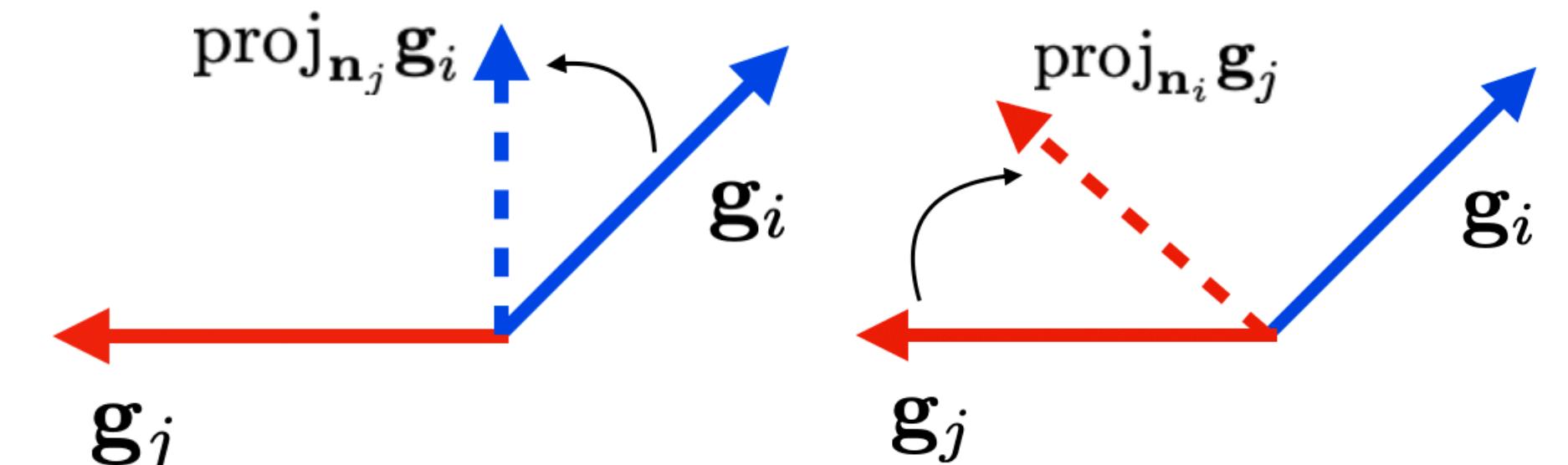
i.e. due to high curvature



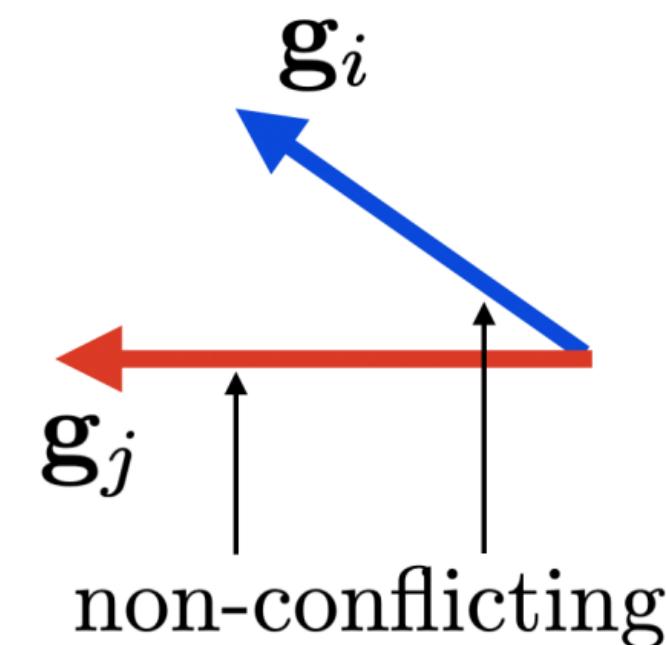
Our solution: try to avoid making other tasks worse, when taking gradient step.

## Algorithm:

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

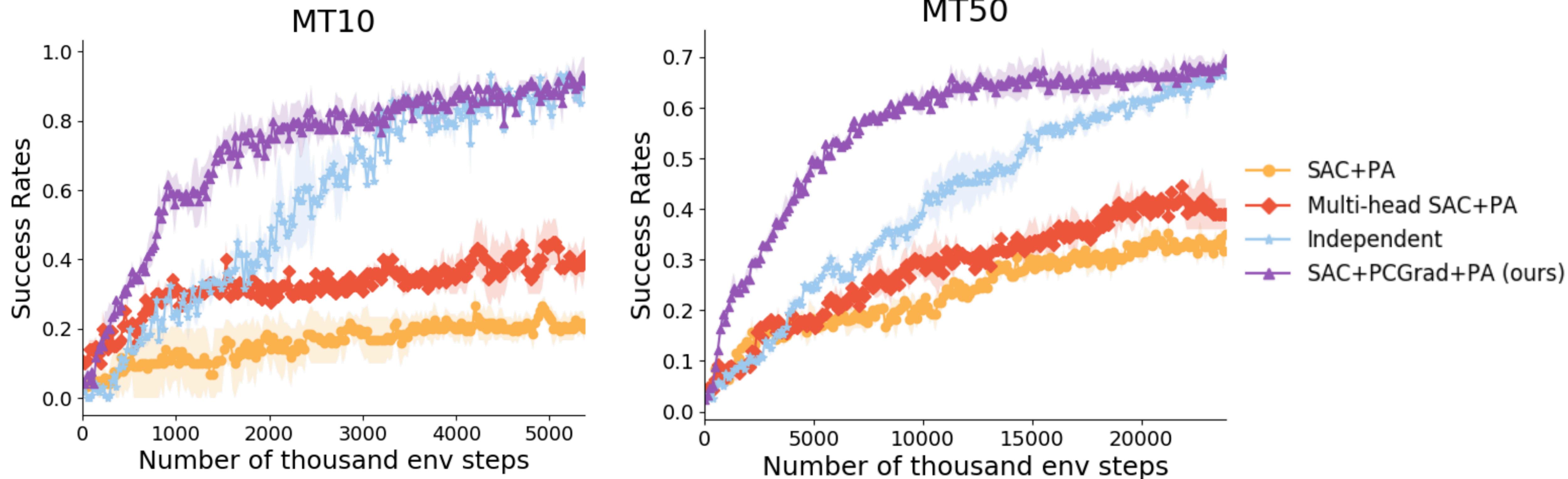


Else:  
leave them alone



i.e. project conflicting gradients  
“PCGrad”

## 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)          | <b>77.5</b> |

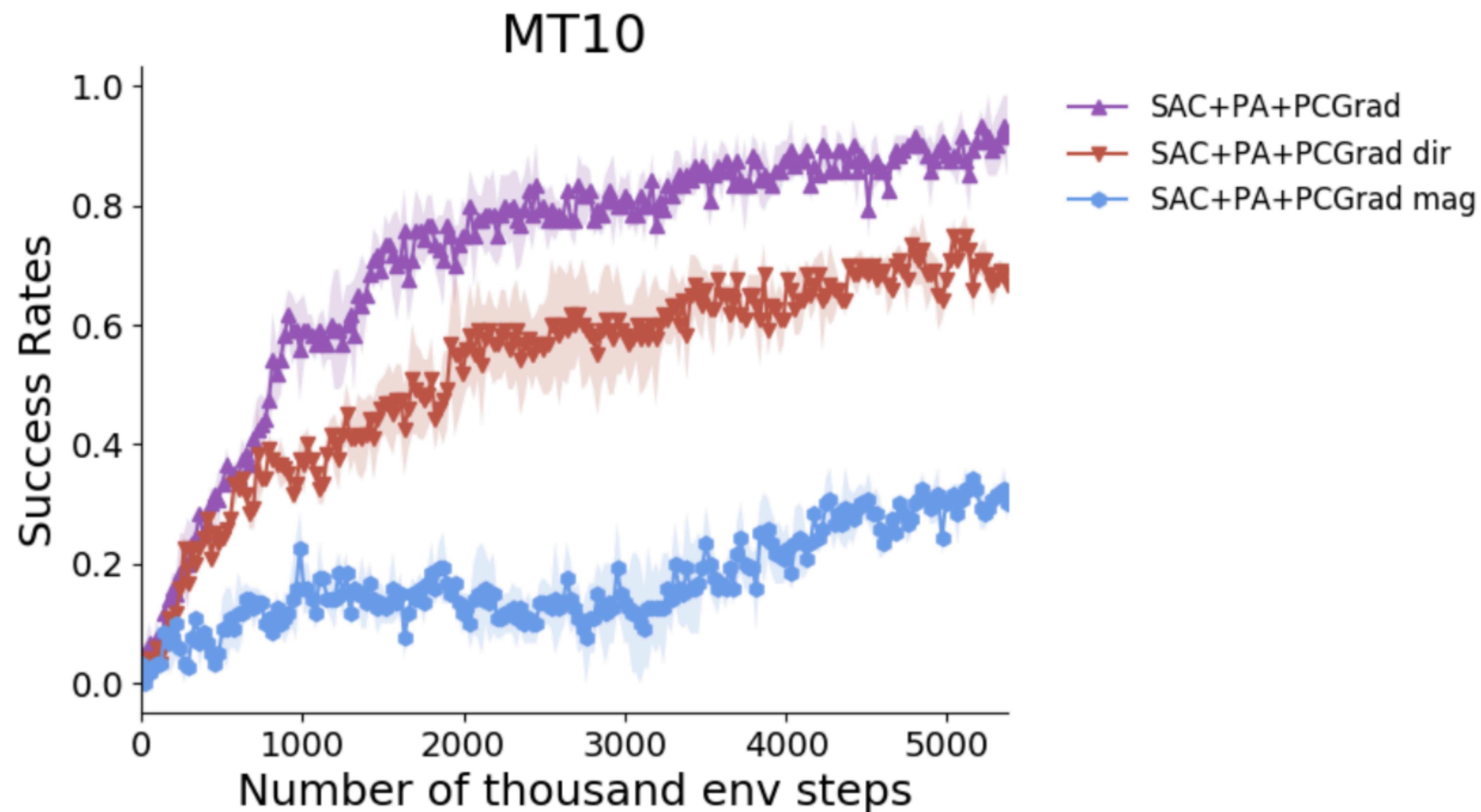
# Multi-Task NYUv2

| #P.         | Architecture                      | Weighting                  | Segmentation    |              | Depth          |               | Surface Normal                |              |                                  |              |              |  |
|-------------|-----------------------------------|----------------------------|-----------------|--------------|----------------|---------------|-------------------------------|--------------|----------------------------------|--------------|--------------|--|
|             |                                   |                            | (Higher Better) |              | (Lower Better) |               | Angle Distance (Lower Better) |              | Within $t^\circ$ (Higher Better) |              |              |  |
|             |                                   |                            | mIoU            | Pix Acc      | Abs Err        | Rel Err       | Mean                          | Median       | 11.25                            | 22.5         | 30           |  |
| $\approx 3$ | Cross-Stitch <sup>†</sup>         | 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 <sup>†</sup> , $T = 2$ | <b>16.11</b>    | <b>53.19</b> | <b>0.5922</b>  | <b>0.2611</b> | <b>32.34</b>                  | <b>26.91</b> | <b>21.81</b>                     | <b>43.14</b> | <b>54.92</b> |  |
| 1.77        | MTAN <sup>†</sup>                 | Equal Weights              | <b>17.72</b>    | 55.32        | <b>0.5906</b>  | 0.2577        | 31.44                         | <b>25.37</b> | <b>23.17</b>                     | 45.65        | 57.48        |  |
|             |                                   | Uncert. Weights*           | 17.67           | <b>55.61</b> | 0.5927         | 0.2592        | <b>31.25</b>                  | 25.57        | 22.99                            | <b>45.83</b> | <b>57.67</b> |  |
|             |                                   | DWA <sup>†</sup> , $T = 2$ | 17.15           | 54.97        | 0.5956         | <b>0.2569</b> | 31.60                         | 25.46        | 22.48                            | 44.86        | 57.24        |  |
| 1.77        | MTAN <sup>†</sup> + PCGrad (ours) | Uncert. Weights*           | <b>20.17</b>    | <b>56.65</b> | <b>0.5904</b>  | <b>0.2467</b> | <b>30.01</b>                  | <b>24.83</b> | 22.28                            | <b>46.12</b> | <b>58.77</b> |  |

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

# Why does it work?

## (Part 1)

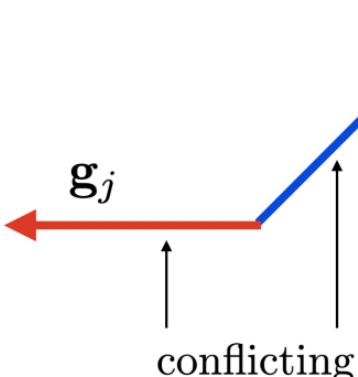


# Why does it work?

## (Part 2)

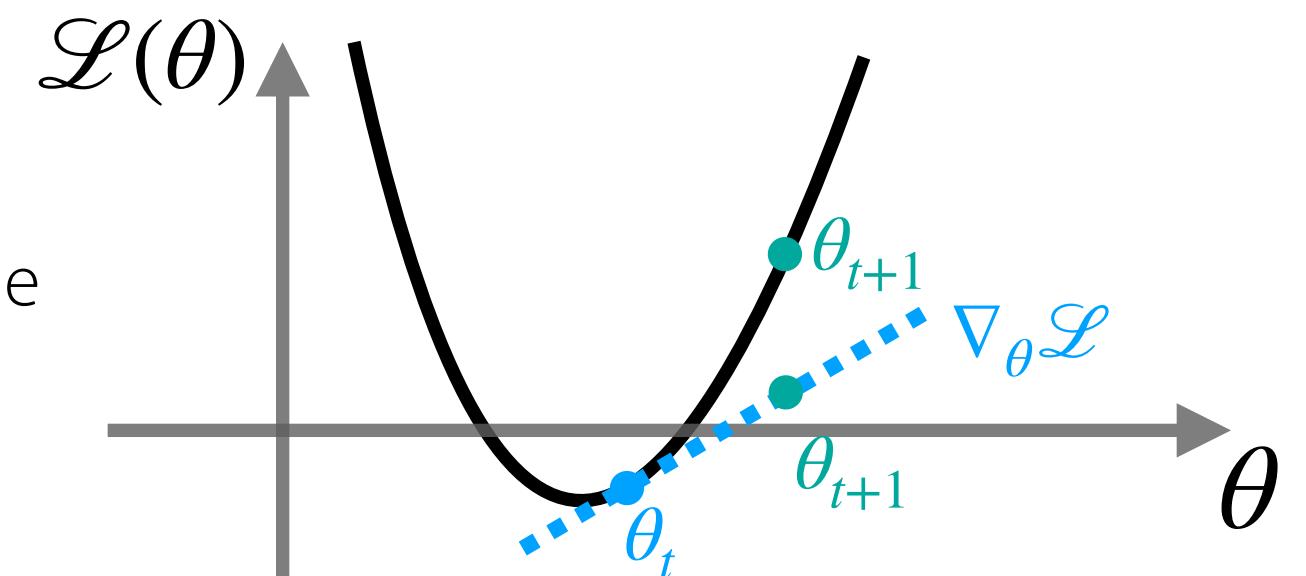
Hypothesis 1: Gradients from different tasks often conflict

If so: would see negative inner product of gradients



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

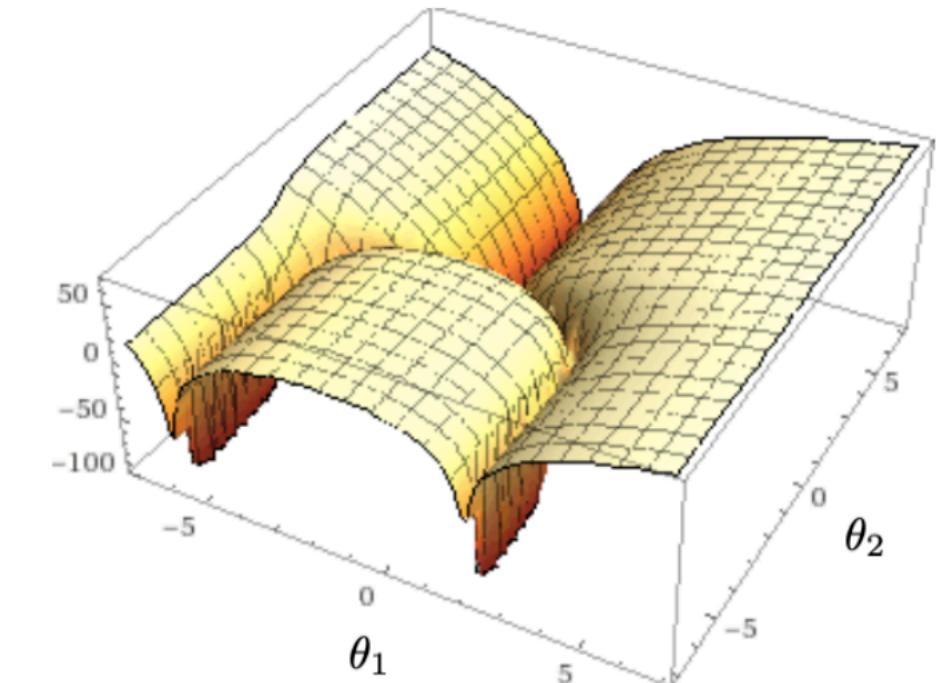
i.e. due to high curvature & difference in grad magnitude



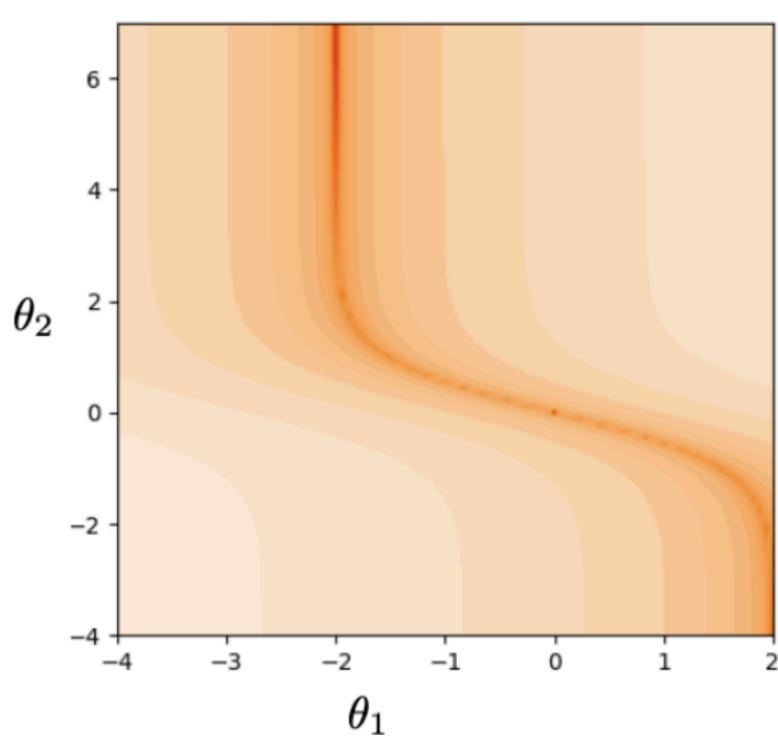
1. conflicting gradients
2. large positive curvature
3. difference in gradient magnitude

Is PCGrad **provably** better under these three conditions?

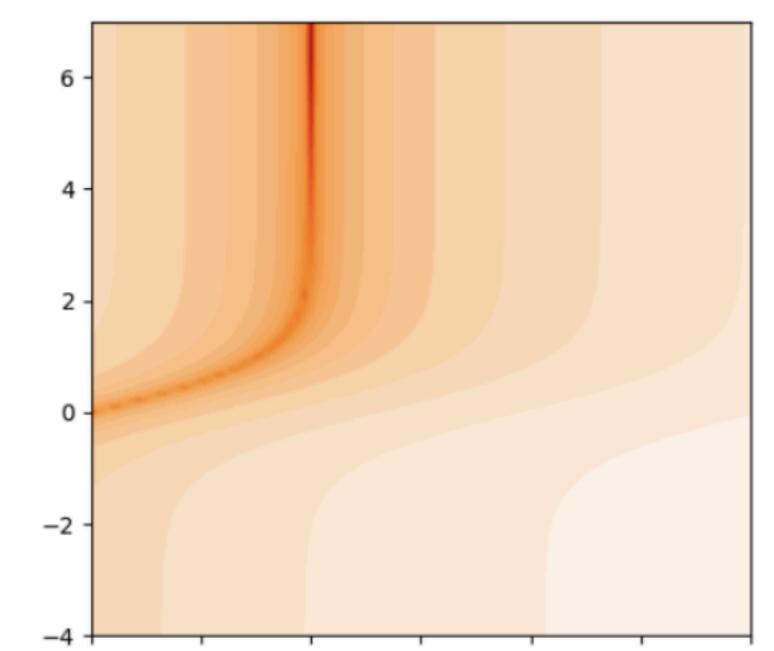
Multi-Task Objective



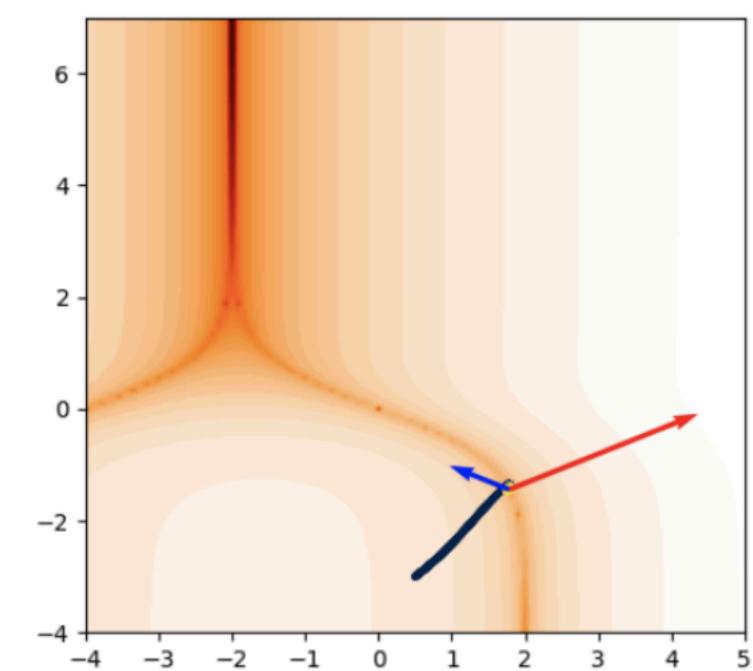
Task 1 Objective



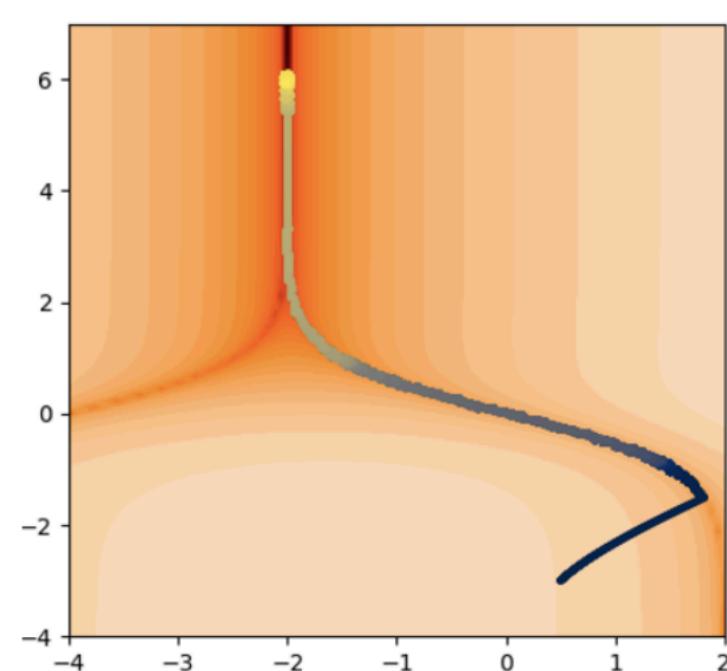
Task 2 Objective



Adam



Adam + PCGrad



Are these three conditions actually **why** we see improvements on large-scale problems?

"tragic triad"

1. conflicting gradients
2. large positive curvature
3. difference in gradient magnitude

Is PCGrad **provably** better under these three conditions?

**short answer:** yes, if large enough conflict, curvature, gradient magnitude difference  
(for two tasks)

**long answer:**

**Theorem 2.** Suppose  $\mathcal{L}$  is differentiable and the gradient of  $\mathcal{L}$  is Lipschitz continuous with constant  $L > 0$ . Let  $\theta^{MT}$  and  $\theta^{PCGrad}$  be the parameters after applying one update to  $\theta$  with  $\mathbf{g}$  and PCGrad-modified gradient  $\mathbf{g}^{PC}$  respectively, with step size  $t > 0$ . Moreover, assume  $\mathbf{H}(\mathcal{L}; \theta, \theta^{MT}) \geq \ell \|\mathbf{g}\|_2^2$  for some constant  $\ell \leq L$ , i.e. the multi-task curvature is lower-bounded. Then  $\mathcal{L}(\theta^{PCGrad}) \leq \mathcal{L}(\theta^{MT})$  if

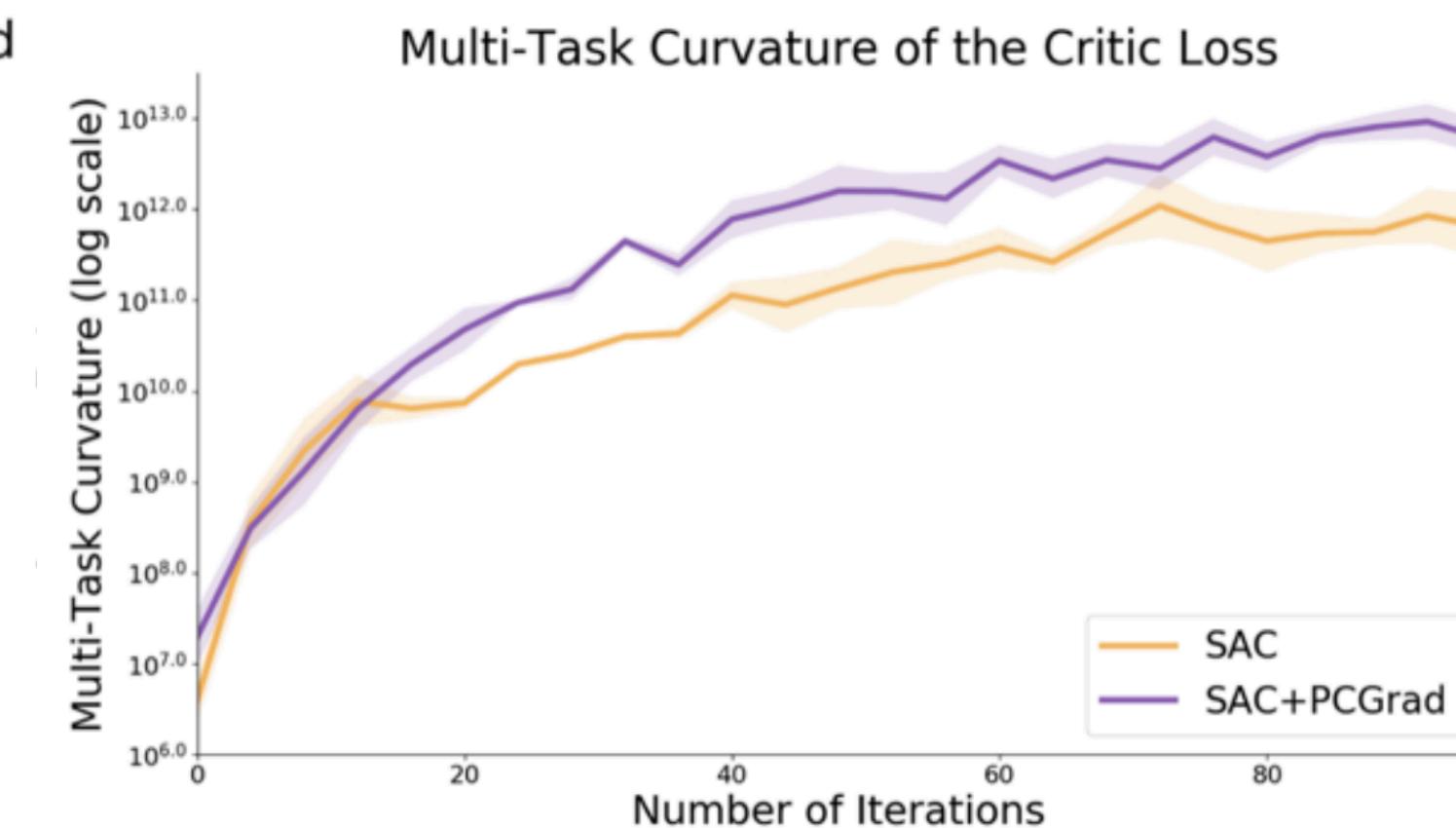
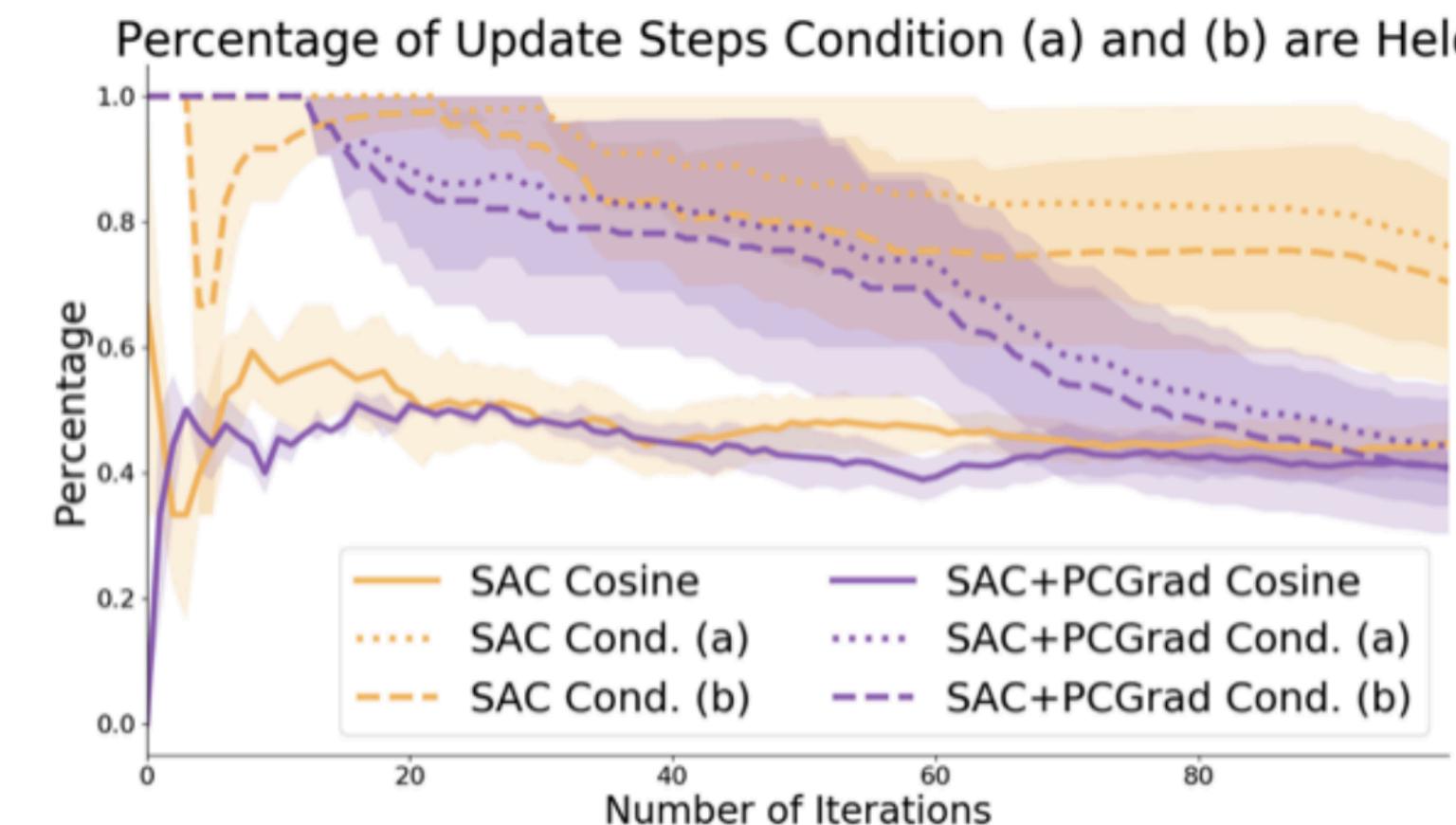
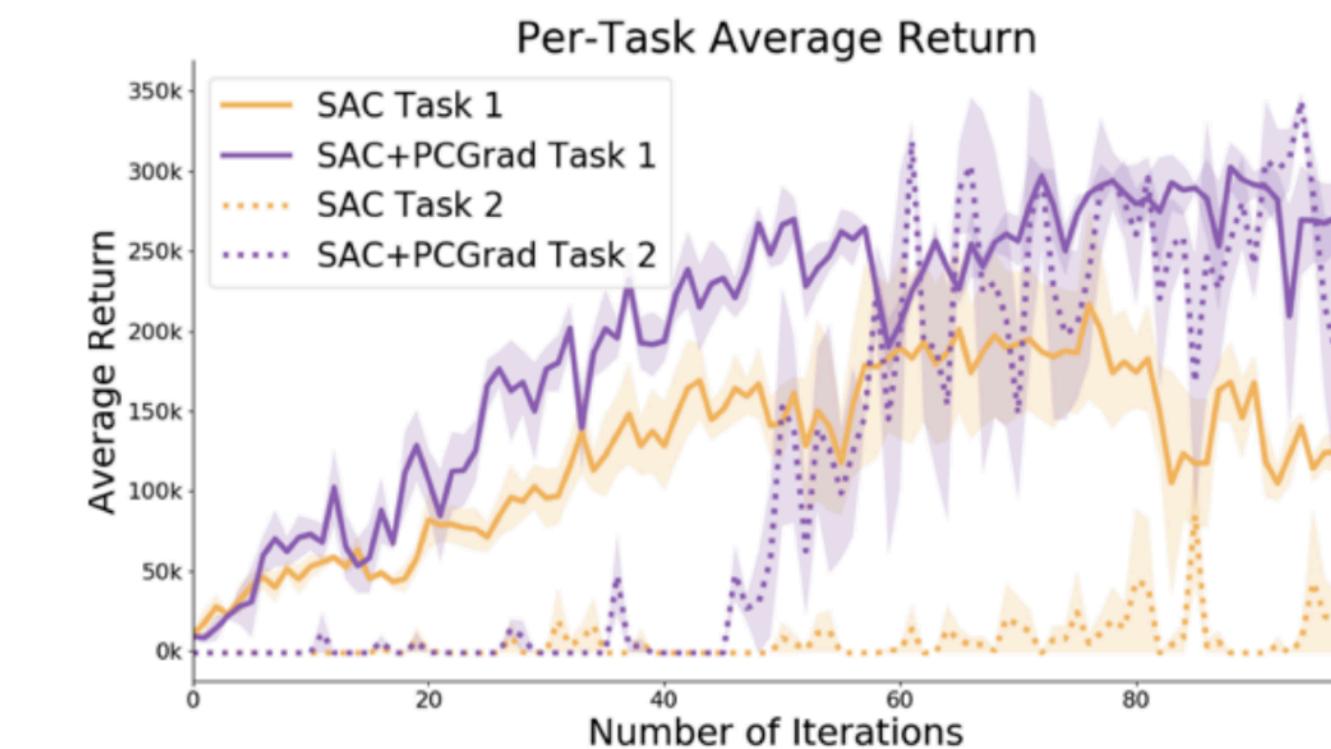
- (a)  $\cos \phi_{12} \leq -\Phi(\mathbf{g}_1, \mathbf{g}_2)$ ,
- (b)  $\ell \geq \xi(\mathbf{g}_1, \mathbf{g}_2)L$ , and
- (c)  $t \geq \frac{2}{\ell - \xi(\mathbf{g}_1, \mathbf{g}_2)L}$ .

*Proof.* See Appendix B.

# Why does it work?

(Part 2)

Are these three conditions actually *why* we see improvements on large-scale problems?



# 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?**

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:

- Train on **broad, dense** task distributions
- Avoid **conflicting gradients**

# Today: What doesn't work very well?

(and how might we fix it)

Meta-learning for addressing distribution shift

Capturing equivariances with meta-learning

Adapting to distribution shift

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

what set of distinct tasks do we train on?

what challenges arise?

Open Challenges

# Open Challenges in Multi-Task and Meta Learning

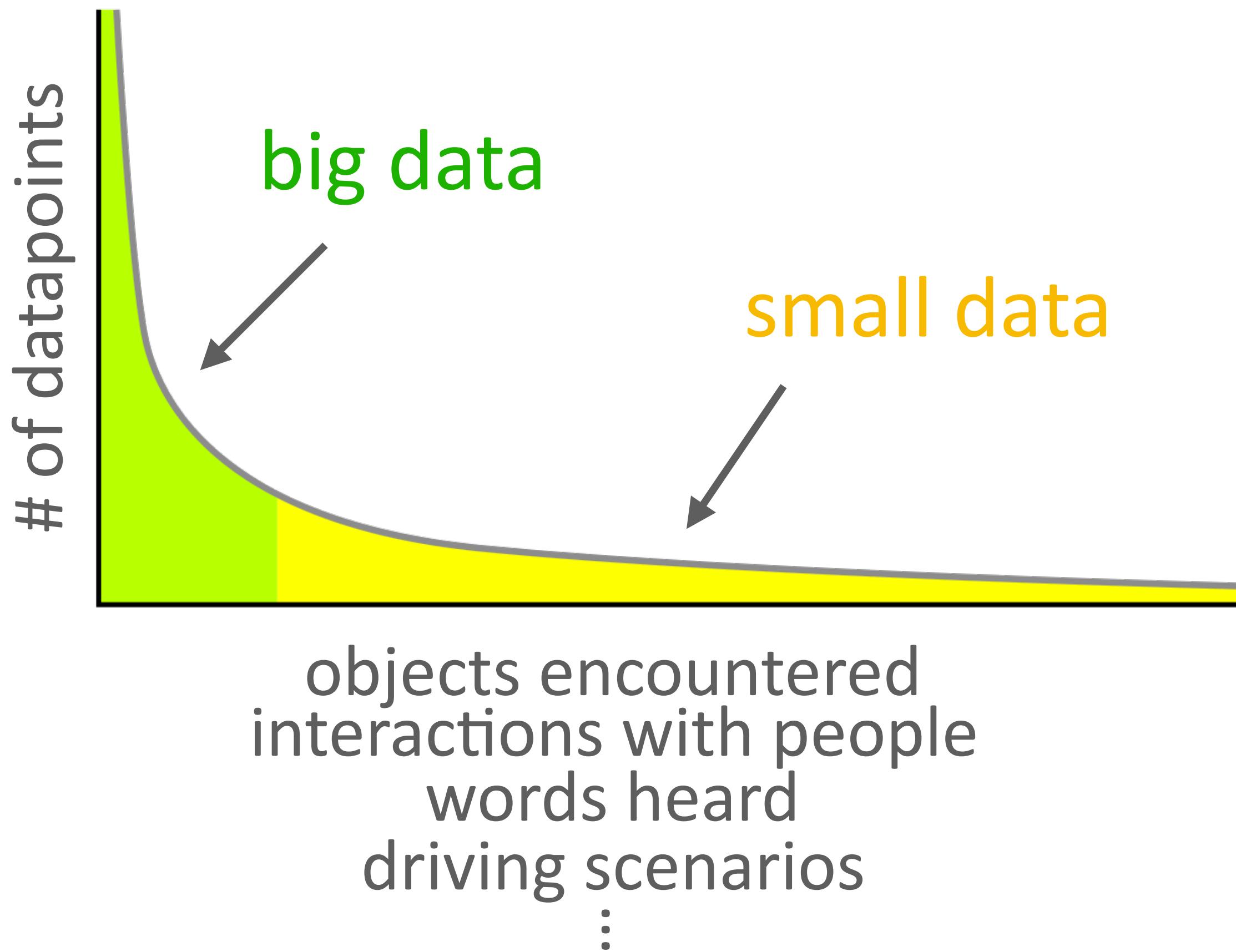
(that we haven't previously covered)

# 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've seen some generalization to the tail:

- prototypical clustering networks for dermatological diseases
- adaptive risk minimization

Further hints might come from domain adaptation, robustness literature.

We learned how to do few-shot learning

...but these few-shot tasks may be from  
a different distribution

# 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.

One initial paper: Liang, Wu, Ziyin, Morency, Salakhutdinov. Learning in Low-resource Modalities via Cross-modal Generalization. 2020.

# 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

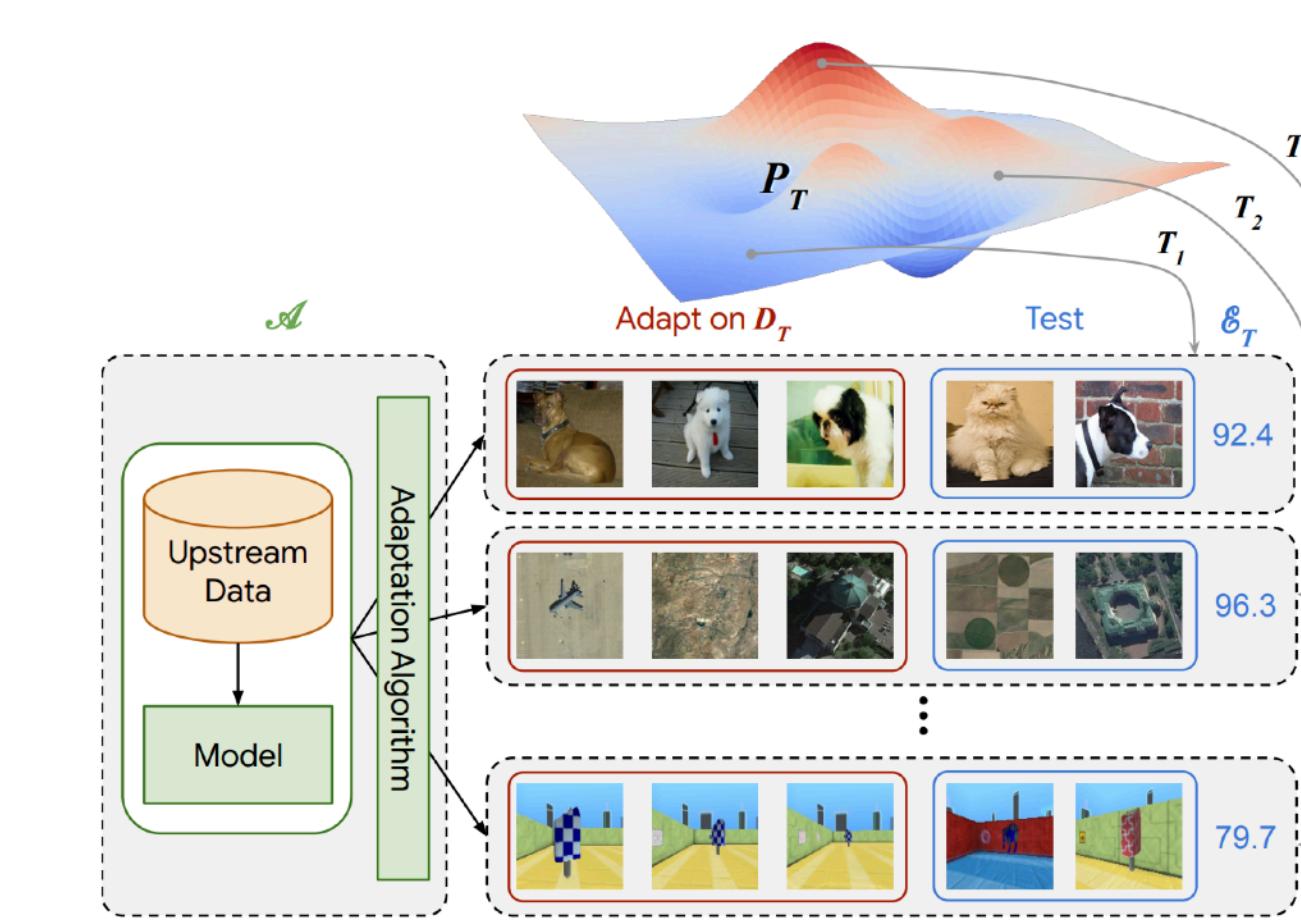
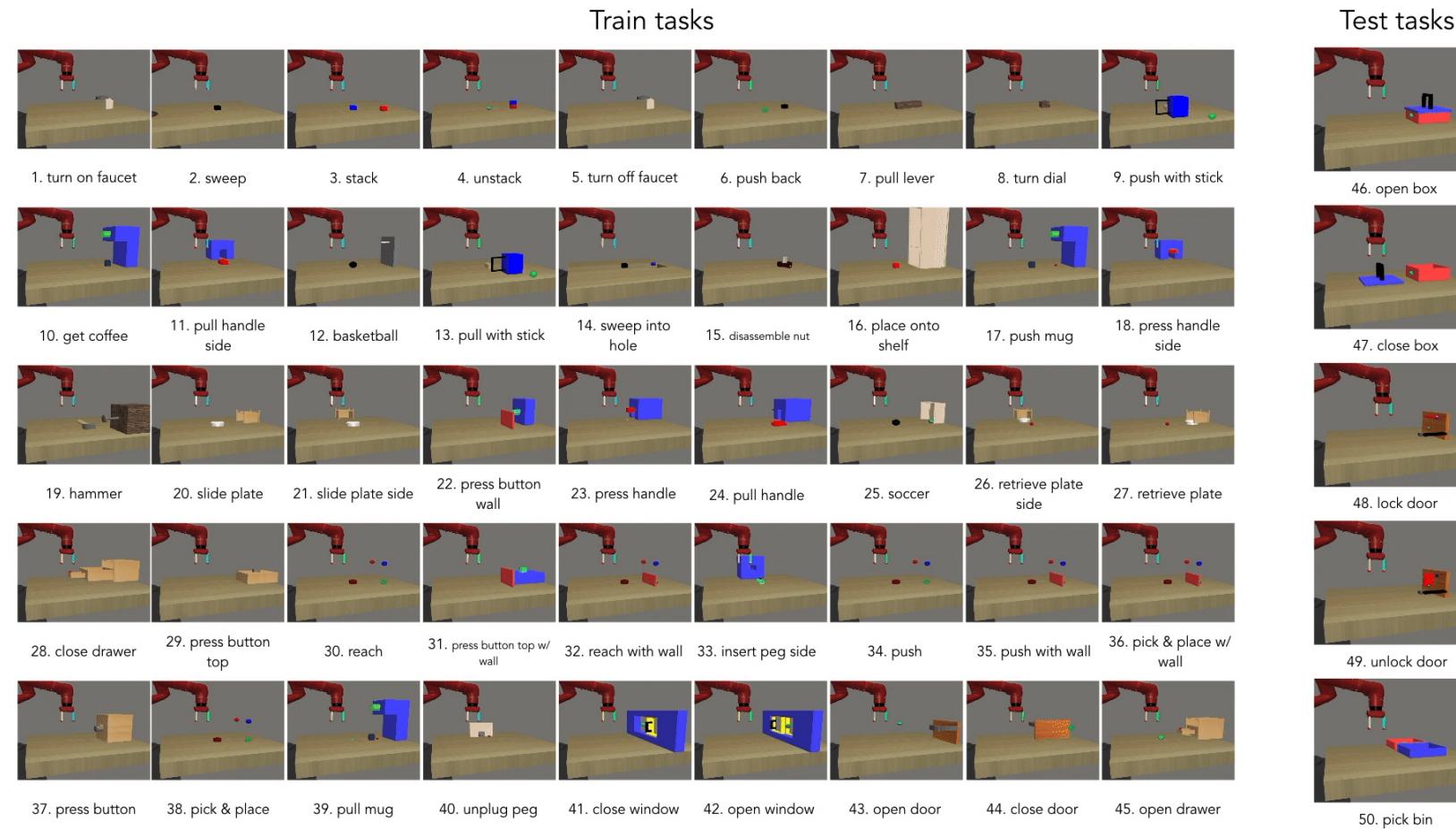
|               |
|---------------|
| ILSVRC        |
| Omniglot      |
| Aircraft      |
| Birds         |
| Textures      |
| Quick Draw    |
| Fungi         |
| VGG Flower    |
| Traffic Signs |
| MSCOCO        |

Meta-Dataset

Triantafillou et al.'19

Meta-World Benchmark

Yu et al.'19

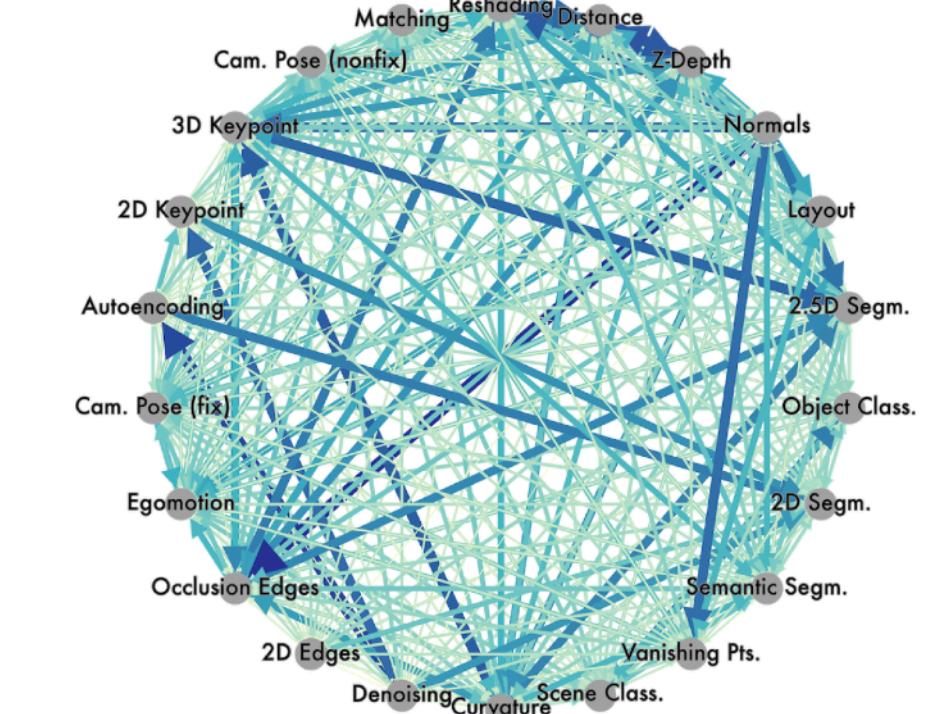


Visual Task Adaptation Benchmark

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

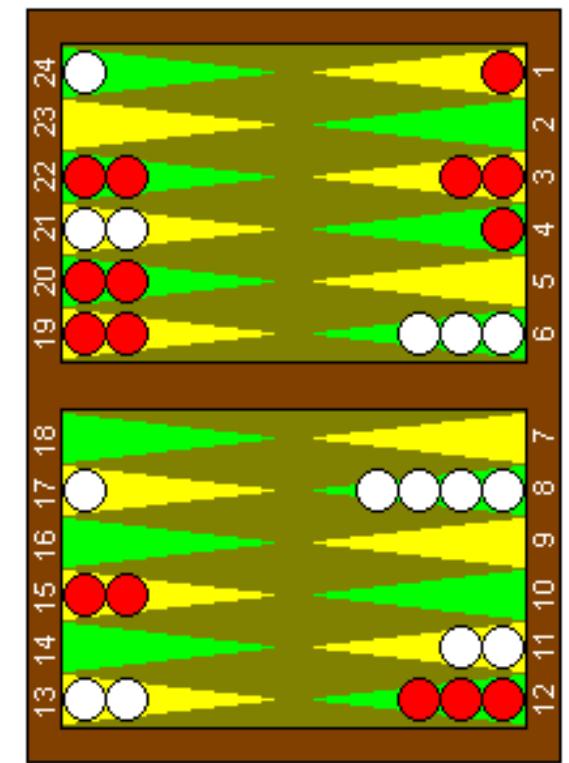
- Breadth: That challenge current algorithms to find common structure
- Realistic: That reflect real-world problems

## 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



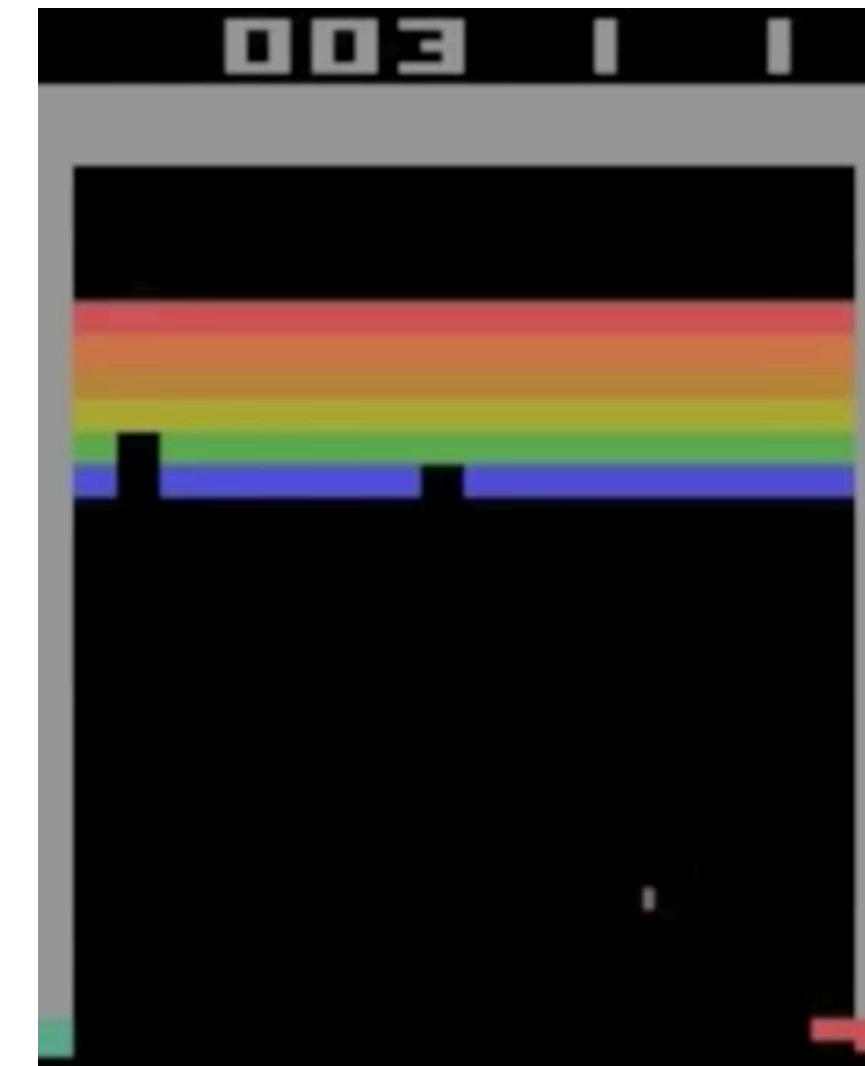
Watson



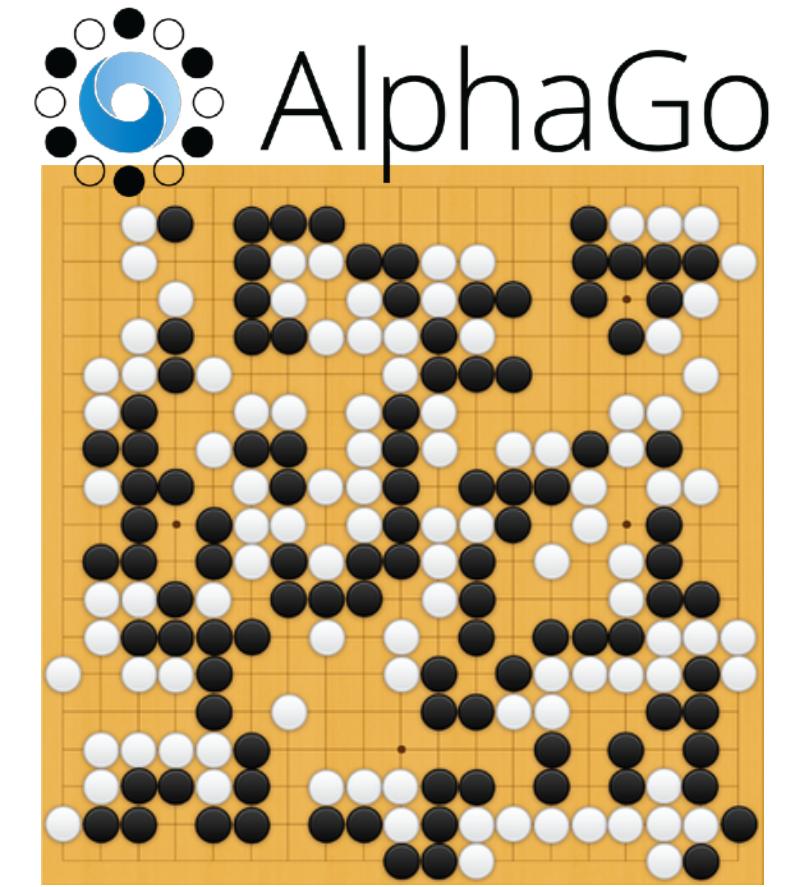
helicopter acrobatics



machine translation



DQN



AlphaGo

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, skill discovery, unsupervised meta-learning)
- perform tasks in sequence (hierarchical RL)
- learn continuously (lifelong learning)

**What's missing?**

# Reminders

Final project presentations next week  
Schedule on Piazza.

Final project report  
Due next Friday midnight.

This is the last lecture!  
We'll leave time for course evaluations at the end.