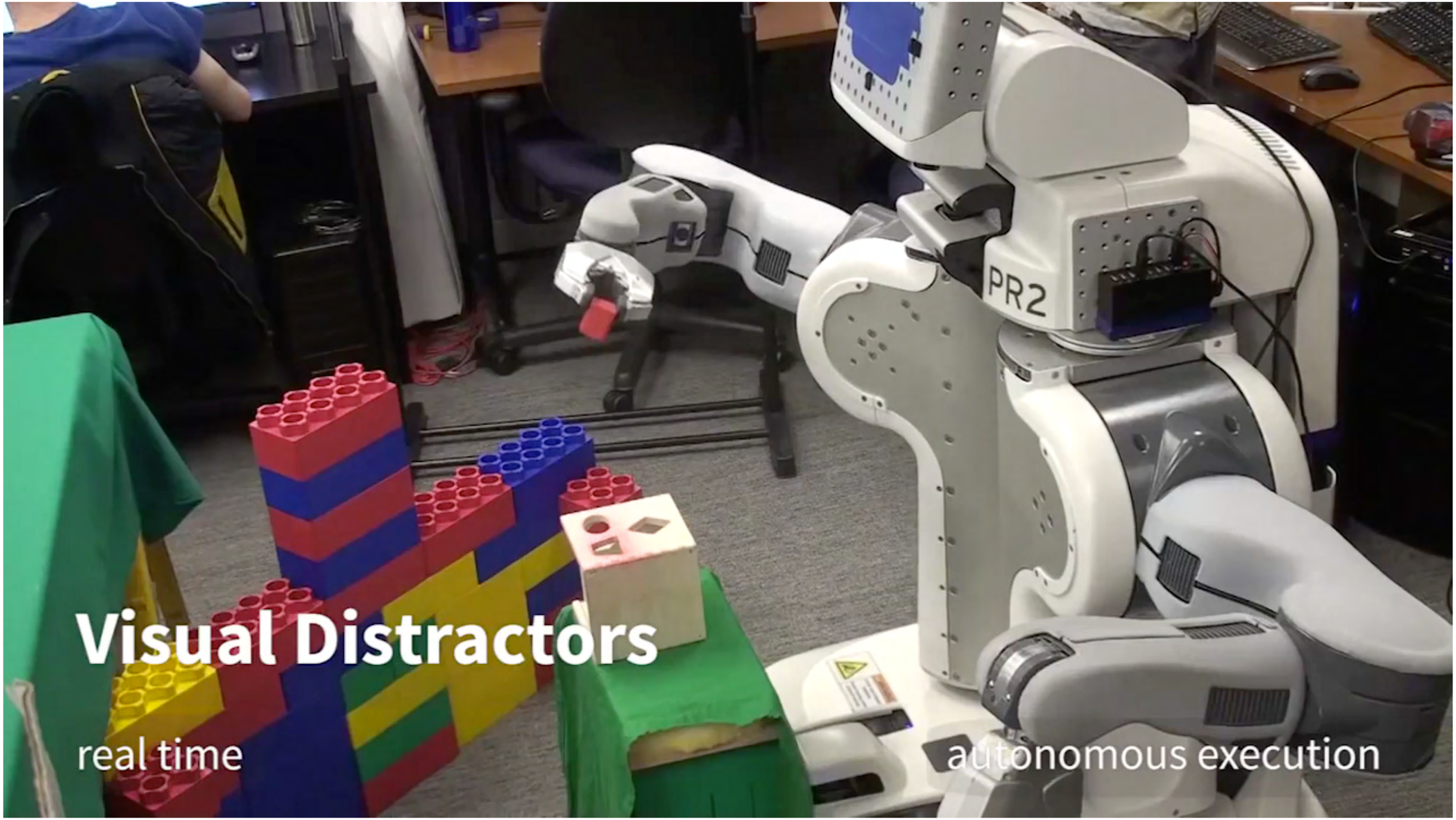
Meta-Learning Frontiers:
Universal, Uncertain, and Unsupervised

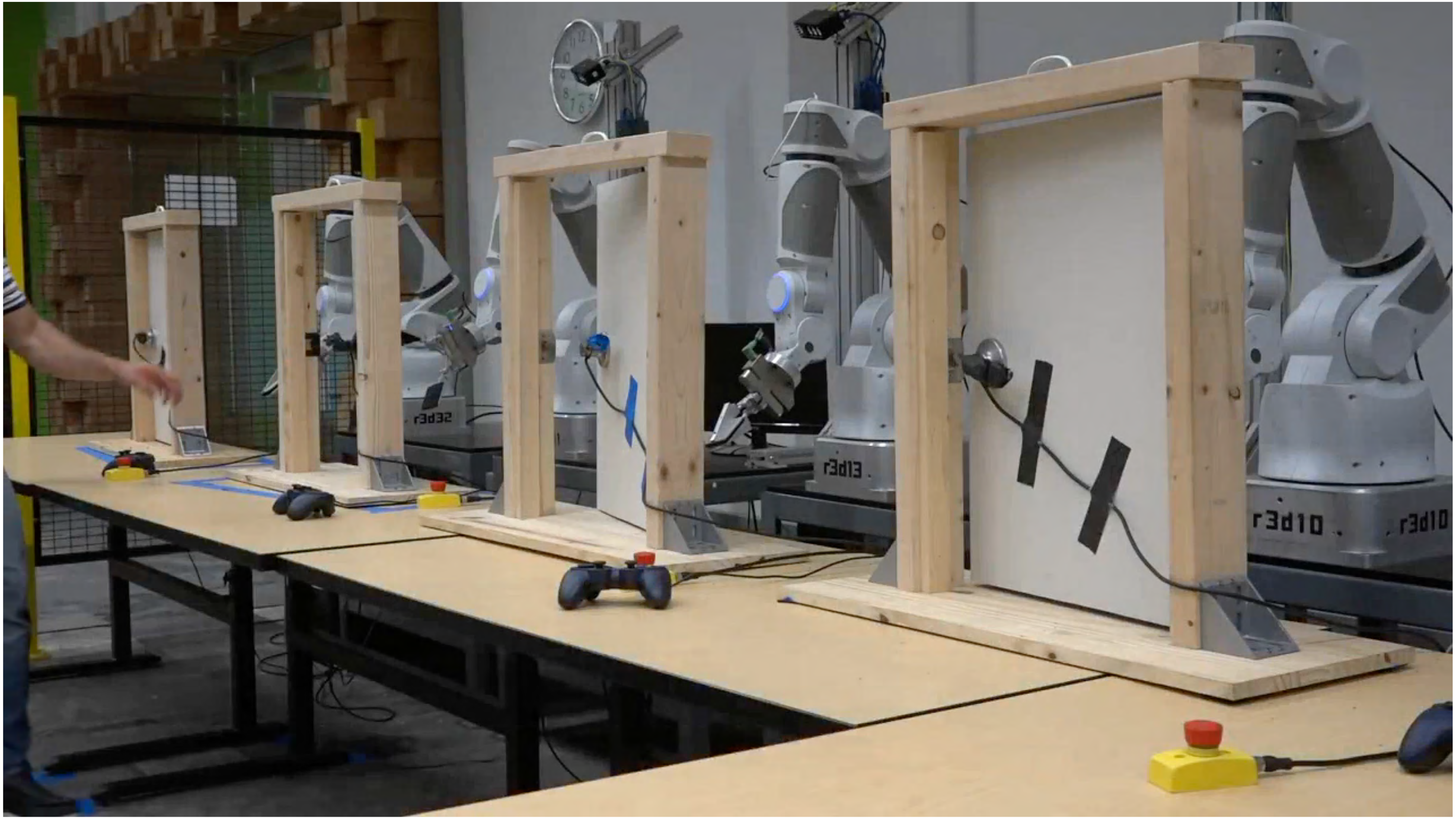
Sergey Levine

Chelsea Finn

UC Berkeley **Google Brain**

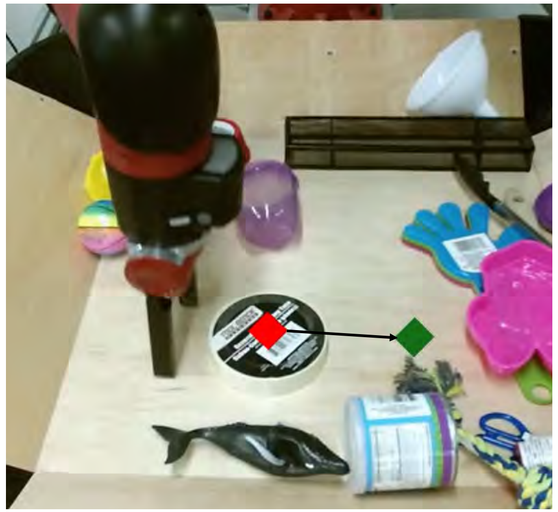


Levine*, Finn*, Darrell, Abbeel, ‘16



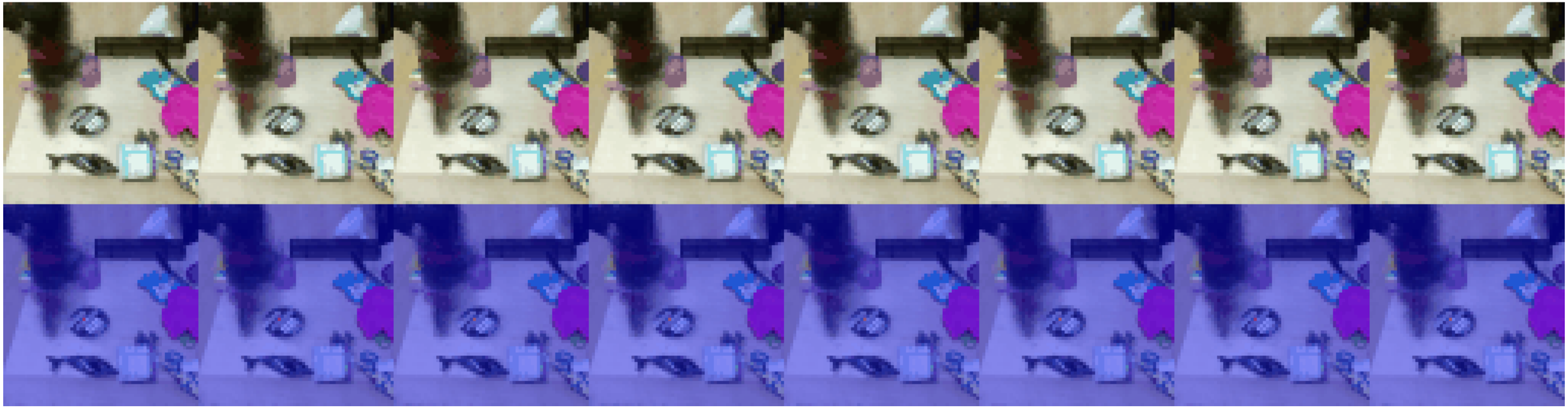
Yahya, Li, Kalakrishnan, Chebotar, Levine, ‘16

Generalizable model-based RL via video prediction

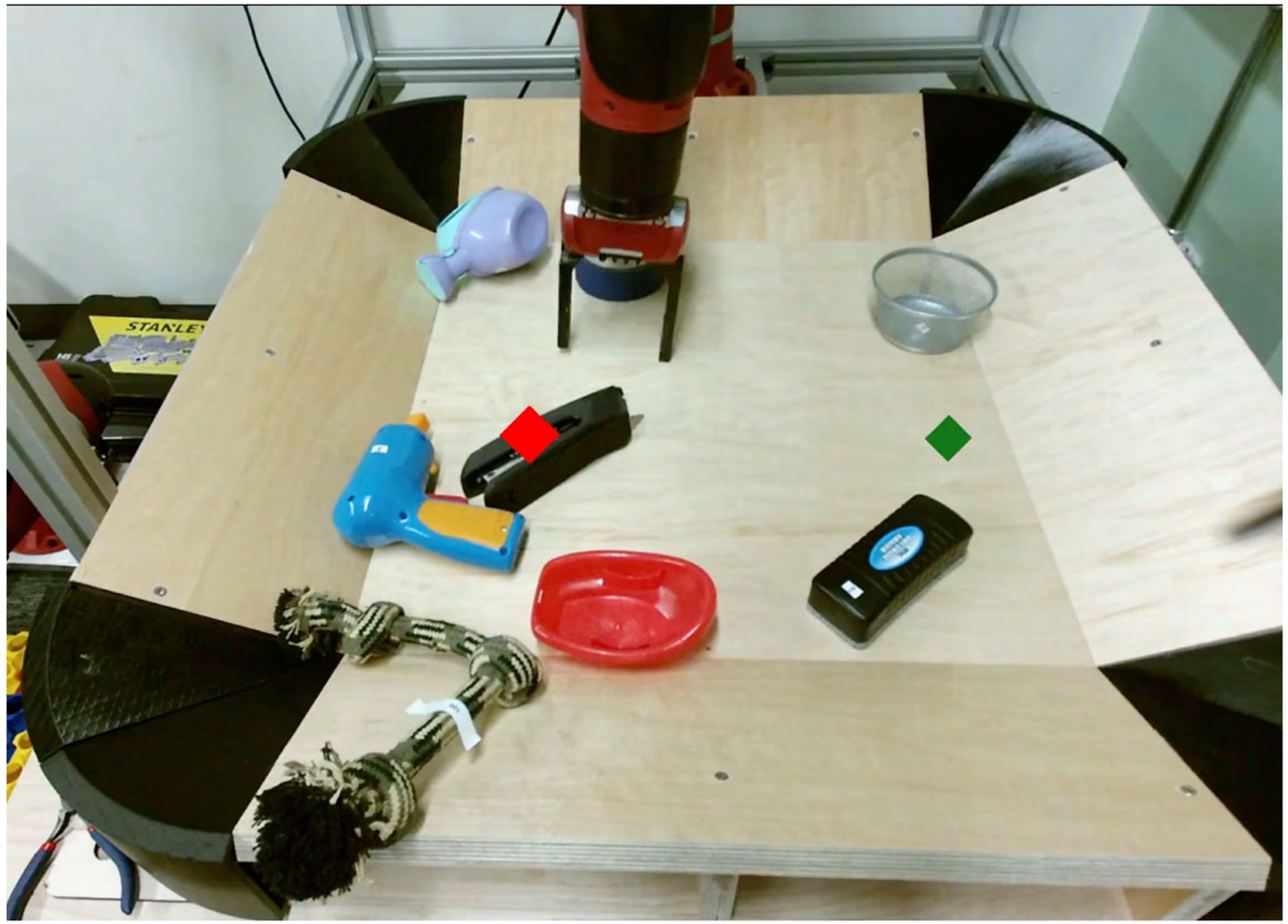


Designated Pixel ◆

Goal Pixel ◆

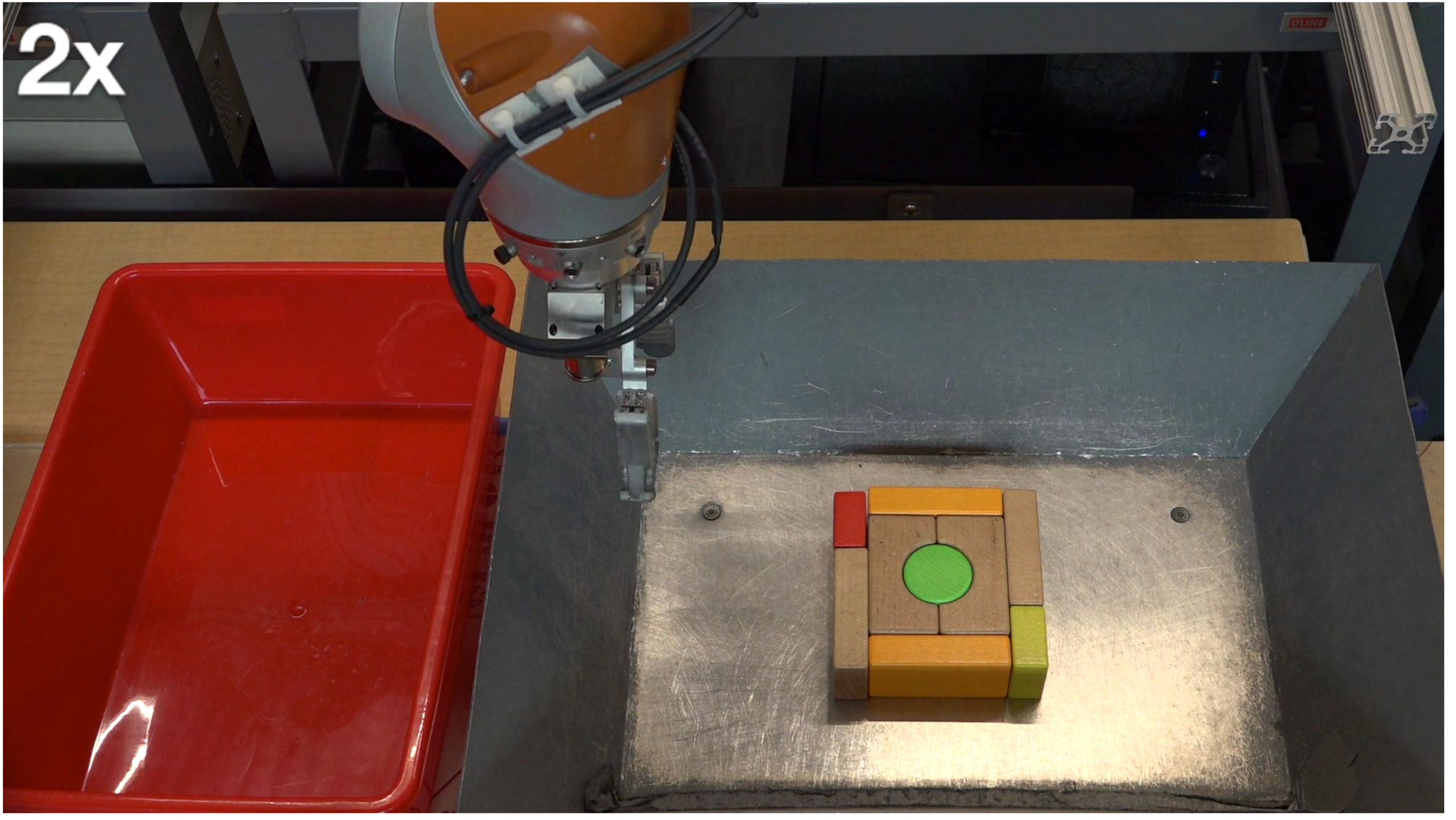


Ebert, Finn, Lee, Levine. Self-Supervised Visual Planning via Temporal Skip Connections. CoRL 2017.

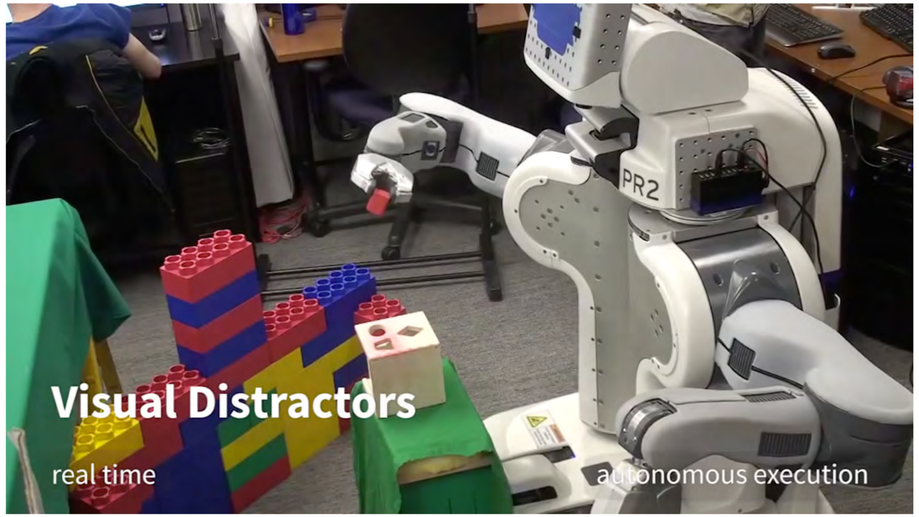


Ebert, Finn, Lee, Levine. Self-Supervised Visual Planning via Temporal Skip Connections. CoRL 2017.

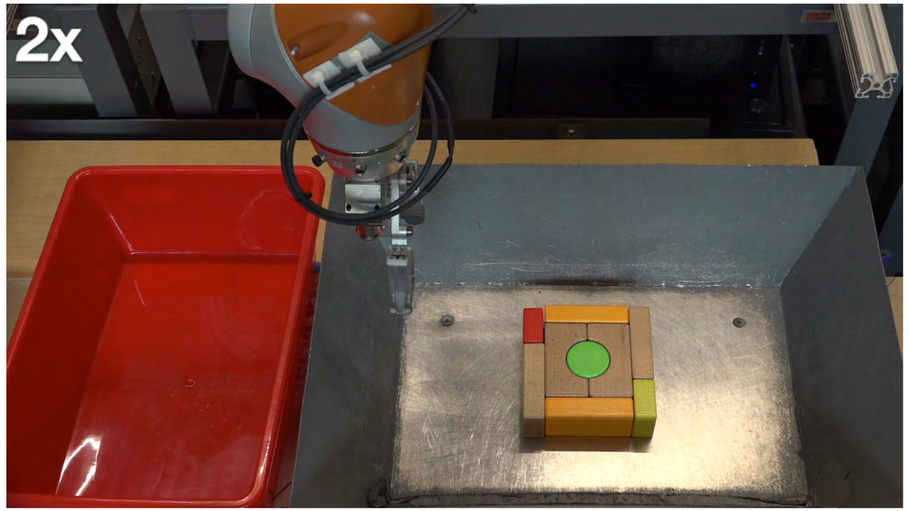
Kalashnikov, Irpan, Pastor, Ibarz, Herzong, Jang, Quillen, Holly, Kalakrishnan, Vanhoucke, Levine. **QT-Opt: Scalable Deep Reinforcement Learning of Vision-Based Robotic Manipulation Skills**



Kalashnikov, Irpan, Pastor, Ibarz, Herzong, Jang, Quillen, Holly, Kalakrishnan, Vanhoucke, Levine. **QT-Opt: Scalable Deep Reinforcement Learning of Vision-Based Robotic Manipulation Skills**



about four hours



about four weeks, nonstop

people can learn new skills **extremely** quickly

how?

we never learn from scratch!

what to transfer?

representations?
models?
can we just optimize for what we **really** want?
can we *learn* to *learn*?

Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference

Outline

- **Meta-Learning Problem Statement**
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference

The Meta-Learning Problem

Supervised Learning:

Inputs: $\mathbf{x}$ Outputs: $\mathbf{y}$ Data: $\{(\mathbf{x}, \mathbf{y})_i\}$

$$\mathbf{y} = f(\mathbf{x}; \theta)$$

Meta-Supervised Learning:

Inputs: $\mathcal{D}_{\text{train}}$ $\mathbf{x}_{\text{test}}$ Outputs: $\mathbf{y}_{\text{test}}$ Data: $\{\mathcal{D}_i\}$

$$\overbrace{\{(\mathbf{x}, \mathbf{y})_{1:K}\}}^{\mathcal{D}_{\text{train}}}$$

$$\mathcal{D}_i : \{(\mathbf{x}, \mathbf{y})_j\}$$

$$\mathbf{y}_{\text{test}} = f(\mathcal{D}_{\text{train}}, \mathbf{x}_{\text{test}}; \theta)$$

Why is this view useful?

Reduces the problem to the design & optimization of *f*.

Finn, Levine. *Meta-learning and Universality: Deep Representation…* ICLR 2018

Example: Few-Shot Classification

Given 1 example of 5 classes:

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

Classify new examples

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

meta-training

$\mathcal{T}_1$

$\mathcal{T}_2$

training classes

diagram adapted from Ravi & Larochelle '17

Outline

- Meta-Learning Problem Statement
- **Model-Agnostic Meta-Learning (MAML)**
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference

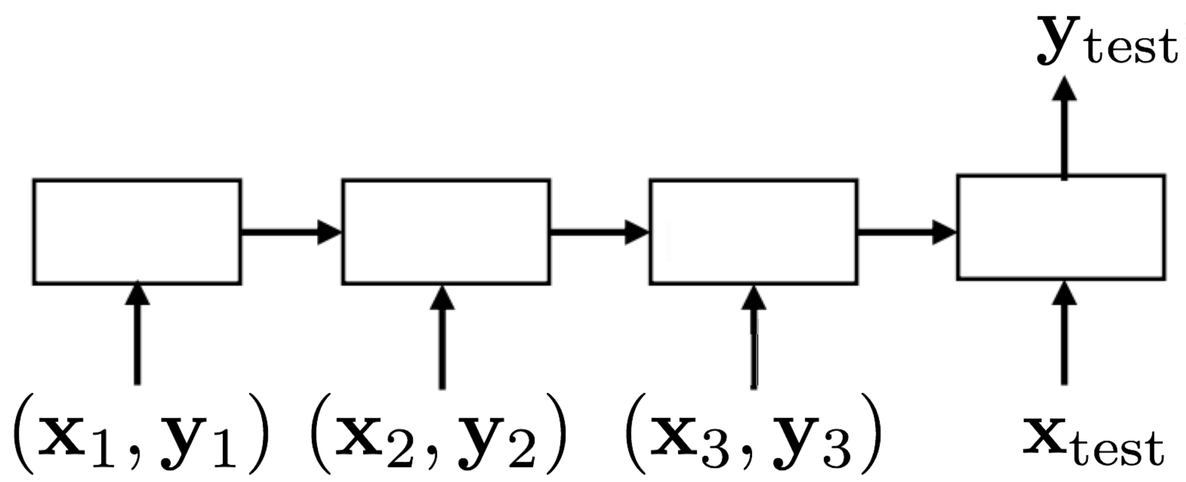
Design of *f* ?

$\mathcal{D}_{\text{train}}\ \mathbf{x}_{\text{test}} \longrightarrow \mathbf{y}_{\text{test}}$

Recurrent network
(LSTM, NTM, Conv)

$$\mathbf{y}_{\text{test}} = f(\mathcal{D}_{\text{train}}, \mathbf{x}_{\text{test}}; \theta)$$

Santoro et al. '16, Duan et al. '17, Wang et al. '17, Munkhdalai & Yu '17, Mishra et al. '17, …



— complex model for complex task of learning

— impractical data requirements

Learning Few-Shot Adaptation

pretrained parameters

Fine-tuning $\phi_j \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}^j_{\text{train}}(\theta)$

[test-time]

training data for new task

Our method $\min_\theta \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta))$

Key idea: Train over many tasks, to learn parameter vector θ that transfers

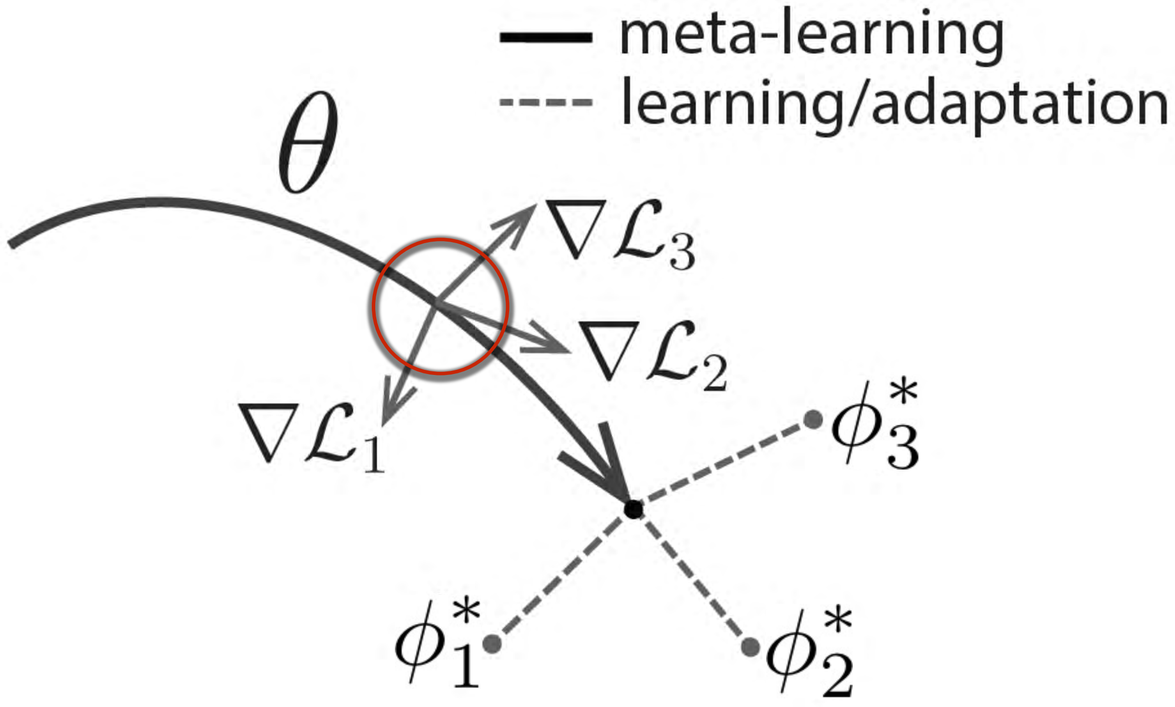
Finn, Abbeel, Levine ICML '17

Learning Few-Shot Adaptation

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta))$$

θ parameter vector being meta-learned

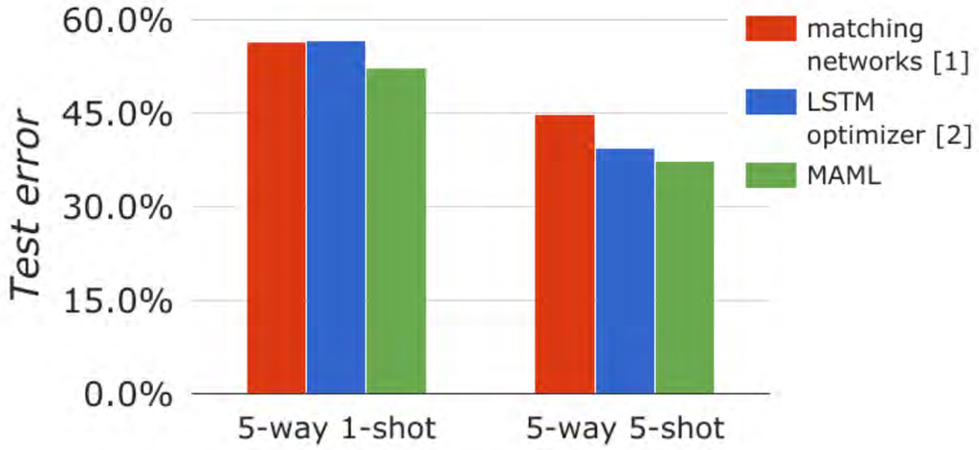
ϕ_i^* optimal parameter vector for task i



Model-**A**gnostic **M**eta-**L**earning

Finn, Abbeel, Levine ICML '17

MiniImagenet Few-shot Classification



…and the results keep getting better

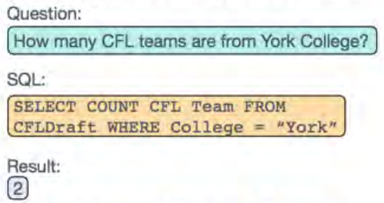
MiniImagenet few-shot benchmark: 5-shot 5-way

Finn et al. ‘17: 63.11%

Li et al. ‘17: 64.03%

Kim et al. ‘18 (AutoMeta): 76.29%

Program Synthesis

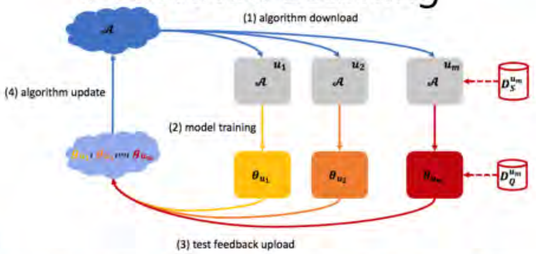


Huang, Wang, Singh, Yih, He NAACL ‘18

Learning to Learn Distributions

Reed, Chen, Paine, van den Oord, Eslami, Rezende, Vinyals, de Freitas ICLR ‘18

Federated Learning

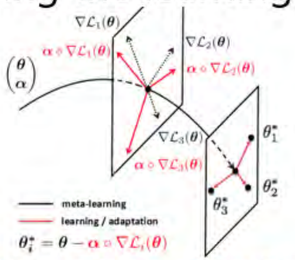


Chen, Dong, Li, He arXiv ‘18

Multi-Agent Competitions

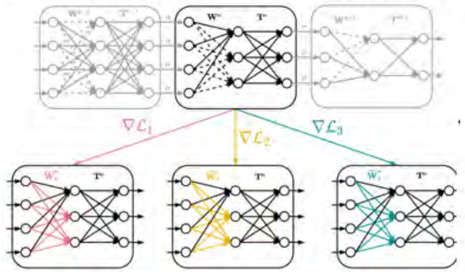
Al-Shedivat, Bansal, Burda, Sutskever Mordatch, Abbeel ICLR ‘18

Learning the learning rate



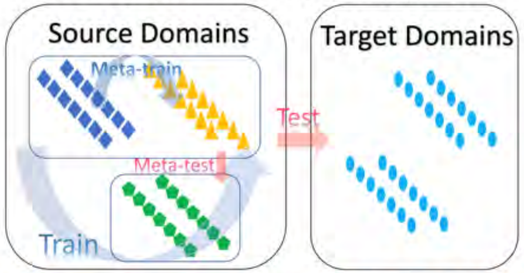
Li, Zhou, Chen, Li arXiv ‘17

Masked Transformations



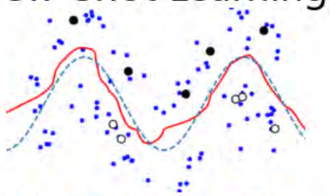
Lee & Choi arXiv ‘18

Domain Generalization



Li, Yang, Song, Hospedales AAAI ‘18

Semi-Supervised Few-Shot Learning



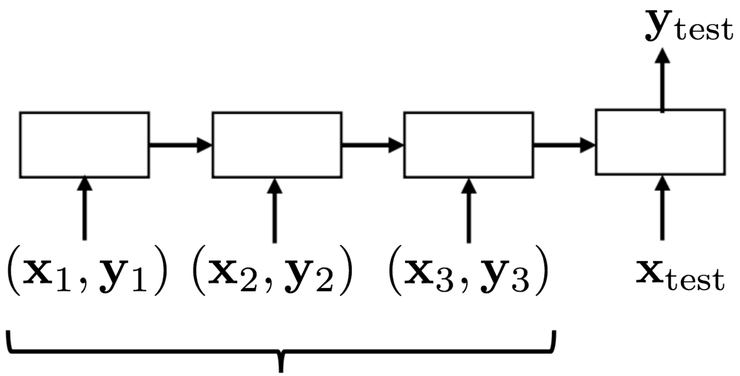
Boney & Ilin ICLR workshop track ‘18

Design of f ?

$$\mathcal{D}_{\text{train}}\ \mathbf{x}_{\text{test}} \longrightarrow \mathbf{y}_{\text{test}}$$

Recurrent network

$$\mathbf{y}_{\text{test}} = f(\mathcal{D}_{\text{train}}, \mathbf{x}_{\text{test}}; \theta)$$



network implements the
"learned learning procedure"

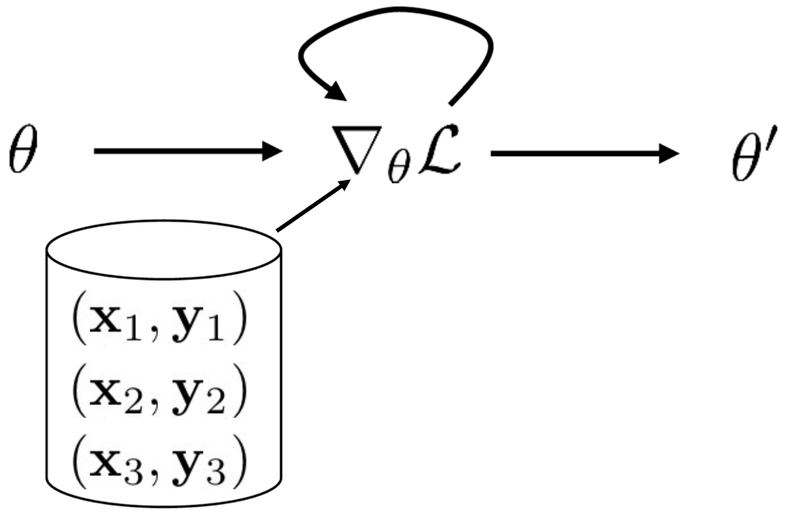
Does it converge?
- Sort of?

What does it converge to?
- Who knows…

What to do if not good enough?
- Nothing

MAML

$$\mathbf{y}_{\text{test}} = f(\mathbf{x}_{\text{test}}; \theta - \alpha \nabla_\theta \mathcal{L}(\mathcal{D}_{\text{train}}))$$



Does it converge?
- Yes (it's gradient descent…)

What does it converge to?
- A local optimum (it's gradient descent…)

What to do if not good enough?
- Keep taking gradient steps (it's gradient descent..)

Does this structure come at a cost?

Recurrent network

$$\mathbf{y}_{\text{test}} = f(\mathcal{D}_{\text{train}}, \mathbf{x}_{\text{test}}; \theta)$$

MAML

$$\mathbf{y}_{\text{test}} = f(\mathbf{x}_{\text{test}}; \theta - \alpha \nabla_\theta \mathcal{L}(\mathcal{D}_{\text{train}}))$$

Does this structure come at a cost?

For a sufficiently deep *f*,
MAML function can approximate any function of $\mathcal{D}_{\text{train}}, \mathbf{x}_{\text{test}}$

Finn & Levine, ICLR 2018

Assumptions:

- nonzero α
- loss function gradient does not lose information about the label
- datapoints in $\mathcal{D}_{\text{train}}$ are unique

Why is this interesting?

MAML has benefit of inductive bias without losing expressive power.

Is this structure useful?

Finn, Levine. *Meta-learning and Universality: Deep Representation…* ICLR 2018

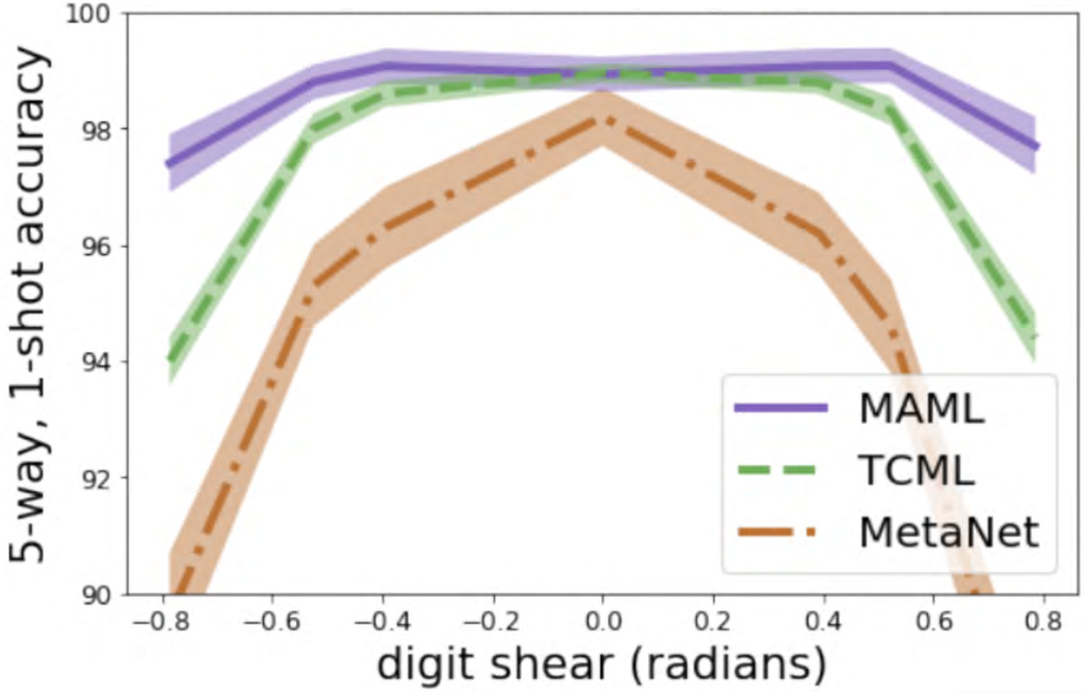
How well can methods generalize to similar, but extrapolated tasks?

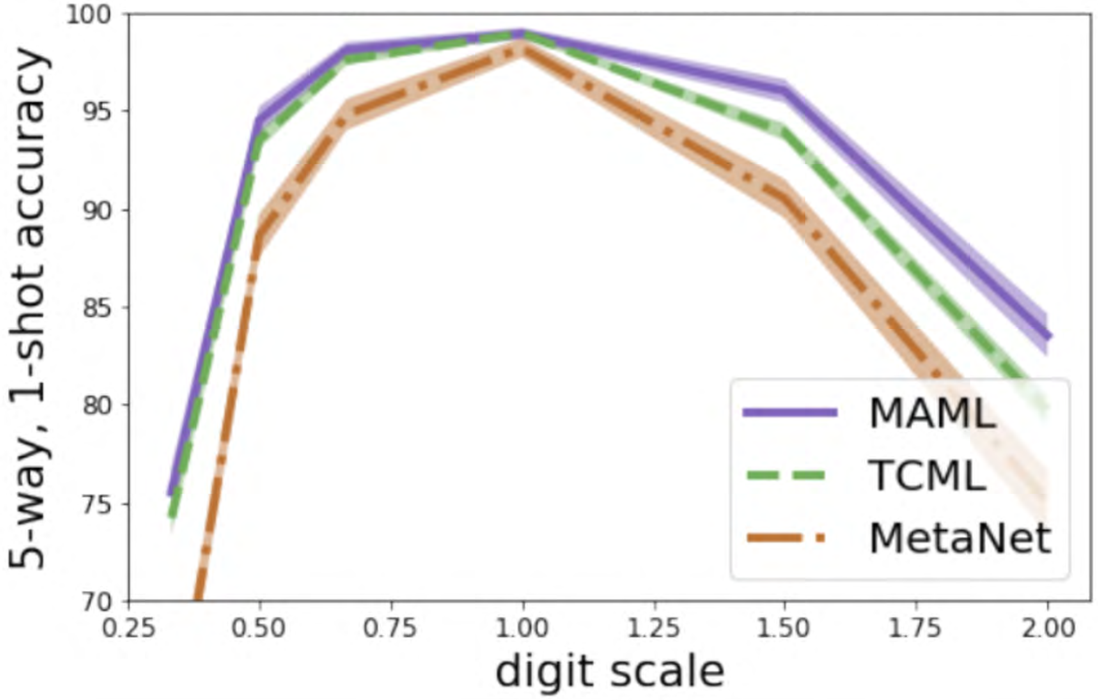
The world is non-stationary.

MAML **TCML**, **MetaNetworks**

Omniglot image classification

performance





task variability

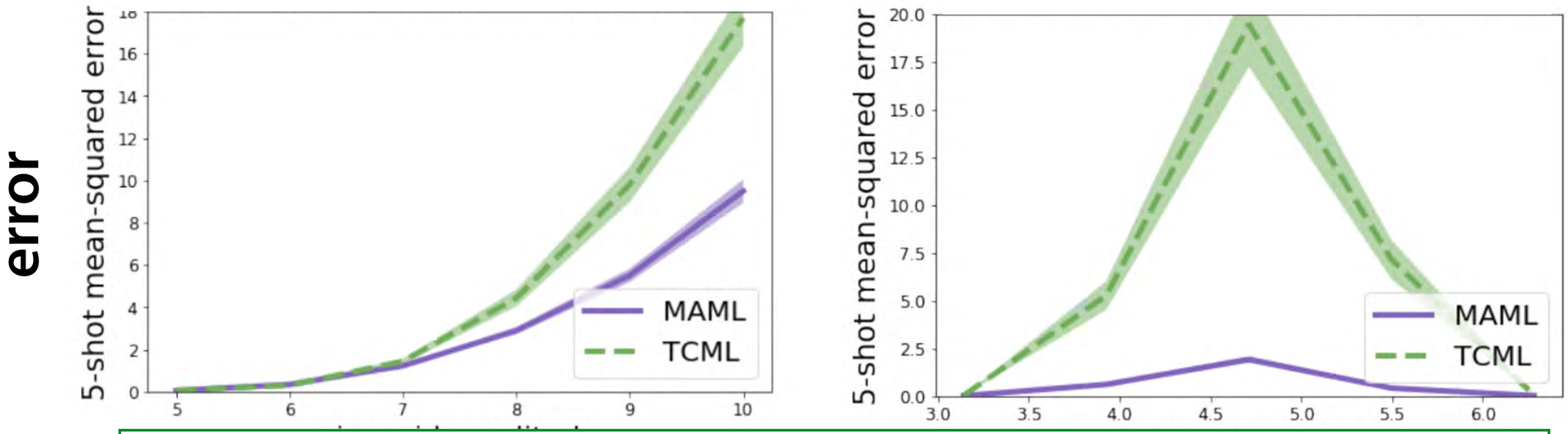
Finn, Levine. *Meta-learning and Universality: Deep Representation…* ICLR 2018

How well can methods generalize to similar, but extrapolated tasks?

The world is non-stationary.

MAML **TCML**

Sinusoid curve regression



Takeaway: Strategies learned with MAML consistently generalize better to out-of-distribution tasks

Finn, Levine. *Meta-learning and Universality: Deep Representation…* ICLR 2018

Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- **Probabilistic Interpretation of MAML**
- Meta-Learning with Automated Task Proposals
- Extensions to Robot Imitation & Intent Inference

A probabilistic interpretation

that's nice but is it… *Bayesian?*

kind of… but it can be more Bayesian

A useful property:

start from $\phi = \theta$ and follow gradient of $\log p(\mathbf{Y}|\mathbf{X}, \phi)$ for K steps

this is equivalent to MAP on $p(\phi|\mathbf{X}, \mathbf{Y})$

for a prior $p(\phi|\theta) = \mathcal{N}(\theta, \Sigma)$

and for a *linear* model $E[\mathbf{Y}] = \mathbf{X}^T\phi$

(Santos, 1996)

Grant, Finn, Levine, Darrell, Griffiths. "Recasting gradient-based meta-learning as hierarchical Bayes."

A probabilistic interpretation

start from $\phi = \theta$ and follow gradient of $\log p(\mathbf{Y}|\mathbf{X}, \phi)$ for K steps

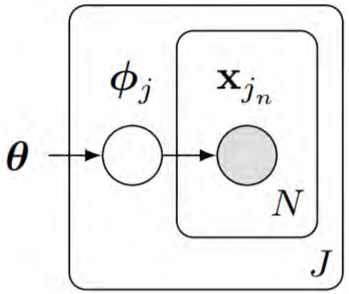
this is equivalent to MAP on $p(\phi|\mathbf{X}, \mathbf{Y})$

for a prior $p(\phi|\theta) = \mathcal{N}(\theta, \Sigma)$

and for a *linear* model $E[\mathbf{Y}] = \mathbf{X}^T\phi$

MAML adaptation:

$$\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^i_{\text{train}})$$



MAP inference in this model

$$p(\phi_i|\theta) = \mathcal{N}(\theta, \Sigma)$$

estimate Hessian for neural nets with KFAC

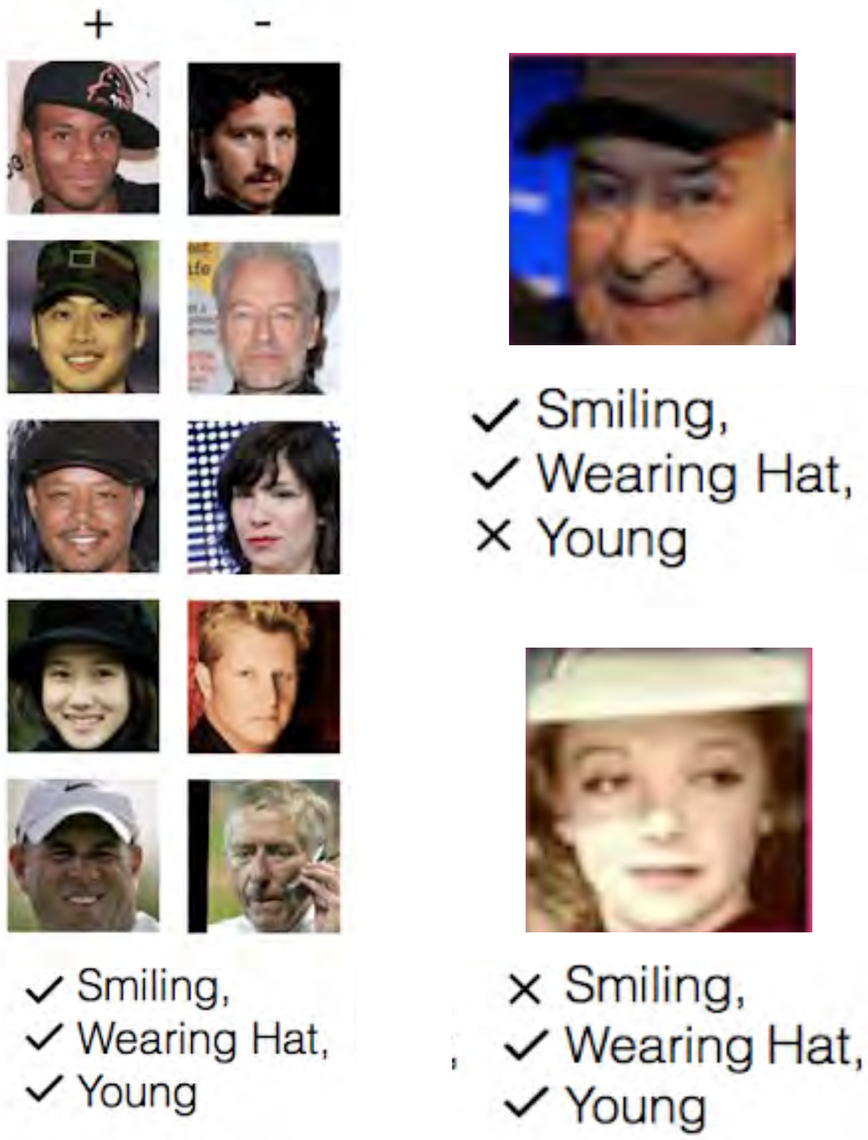
can we do better than MAP?

can use Laplace estimate:

$$-\log p(\mathbf{X} \mid \boldsymbol{\theta}) \approx \sum_j \left[-\log p\left(\mathbf{X}_j \mid \hat{\boldsymbol{\phi}}_j\right) - \log p\left(\hat{\boldsymbol{\phi}}_j \mid \boldsymbol{\theta}\right) + \tfrac{1}{2}\log\det(\mathbf{H}_j) \right]$$

Grant, Finn, Levine, Darrell, Griffiths. “Recasting gradient-based meta-learning as hierarchical Bayes.”

Modeling ambiguity



?

?

Can we learn to generate hypotheses about the underlying function?

Important for:

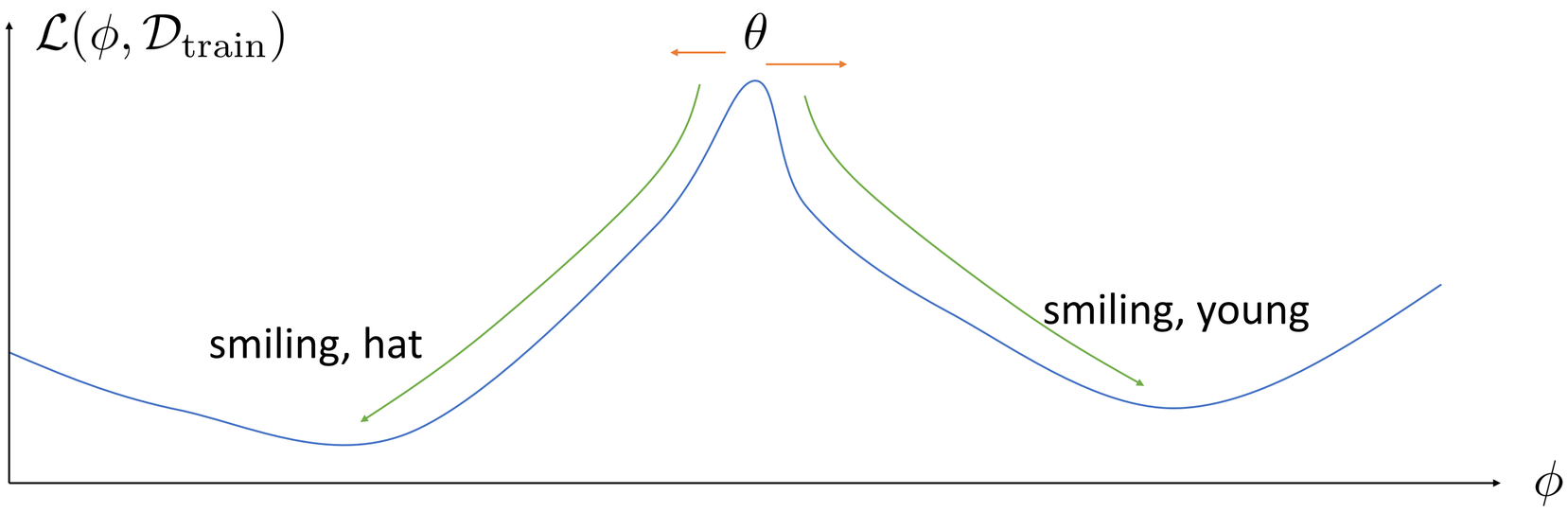
- learning to **actively learn**
- **safety-critical few-shot learning** (e.g. medical imaging)
- learning to **explore** in meta-RL

Modeling ambiguity

✓ Smiling,
✓ Wearing Hat,
✓ Young

Can we *sample* classifiers?

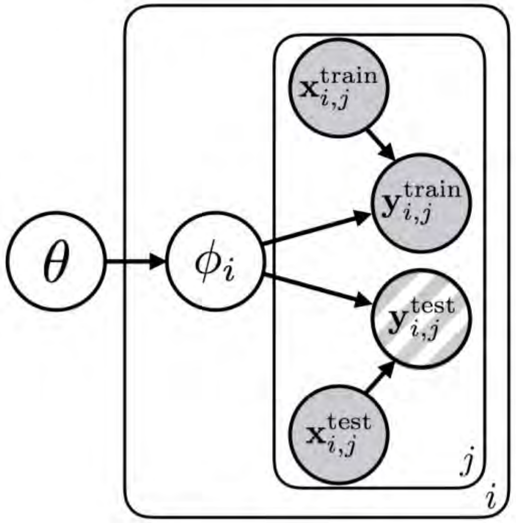
Intuition: we want to learn a prior where a random *kick* can put us in different modes



$$\phi \leftarrow \theta + \epsilon$$

$$\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}(\phi, \mathcal{D}_{\text{train}})$$

Meta-learning with ambiguity



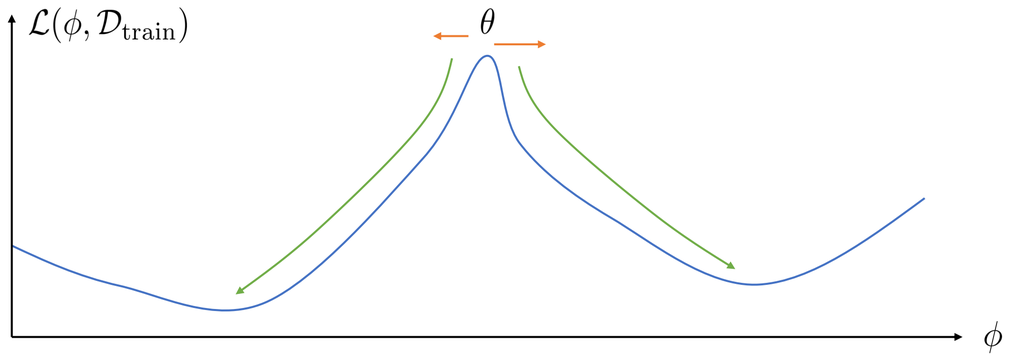
$$\theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$$

$$\phi_i \sim p(\phi_i|\theta)$$

$$\log p(y_i^{\text{train}}|x_i^{\text{train}}, \phi_i)$$

$$\log p(y_i^{\text{test}}|x_i^{\text{test}}, \phi_i)$$

Goal: sample $\phi_i \sim p(\phi_i|x_i^{\text{train}}, y_i^{\text{train}}, \cancel{x_i^{\text{test}}})$



Finn*, Xu*, Levine. Probabilistic Model-Agnostic Meta-Learning

Sampling parameter vectors

$\theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$ $\qquad$ $\log p(y_i^{\text{train}} | x_i^{\text{train}}, \phi_i)$

$\phi_i \sim p(\phi_i | \theta)$ $\qquad$ $\log p(y_i^{\text{test}} | x_i^{\text{test}}, \phi_i)$

Goal: sample $\phi_i \sim p(\phi_i | x_i^{\text{train}}, y_i^{\text{train}})$

$$p(\phi_i | x_i^{\text{train}}, y_i^{\text{train}}) \propto \int p(\theta) p(\phi_i | \theta) p(y_i^{\text{train}} | x_i^{\text{train}}, \phi_i) d\theta$$

$\Rightarrow$ this is completely intractable!

what if we knew $p(\phi_i | \theta, x_i^{\text{train}}, y_i^{\text{train}})$?

$\Rightarrow$ now sampling is easy! just use ancestral sampling!

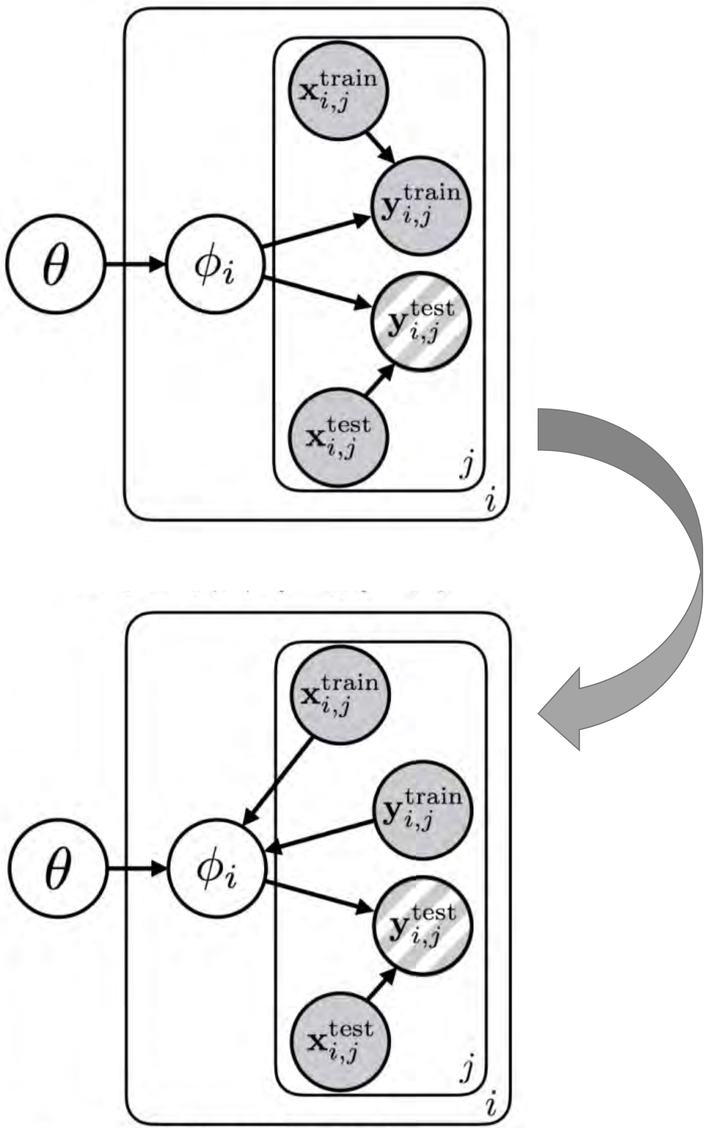
key idea: $p(\phi_i | \theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \delta(\hat{\phi}_i)$

approximate with MAP

this is **extremely** crude

but **extremely** convenient!

$$\hat{\phi}_i \approx \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}} | x_i^{\text{train}}, \theta)$$

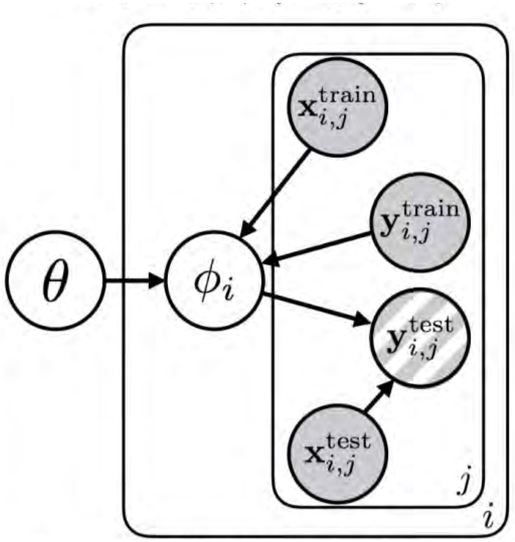


Finn*, Xu*, Levine. Probabilistic Model-Agnostic Meta-Learning

Sampling parameter vectors

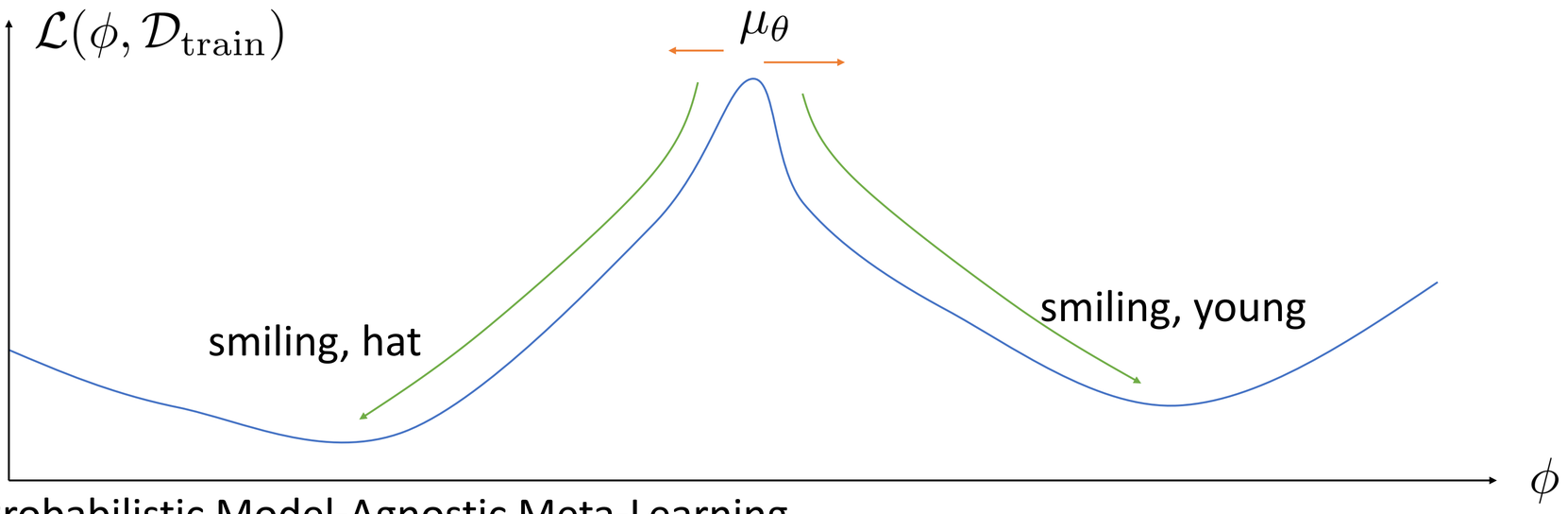
$$\theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$$

key idea: $p(\phi_i | \theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \delta(\hat{\phi}_i) \qquad \hat{\phi}_i \approx \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}} | x_i^{\text{train}}, \theta)$



What does ancestral sampling look like?

1. $\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$
2. $\phi_i \sim p(\phi_i | \theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}} | x_i^{\text{train}}, \theta)$

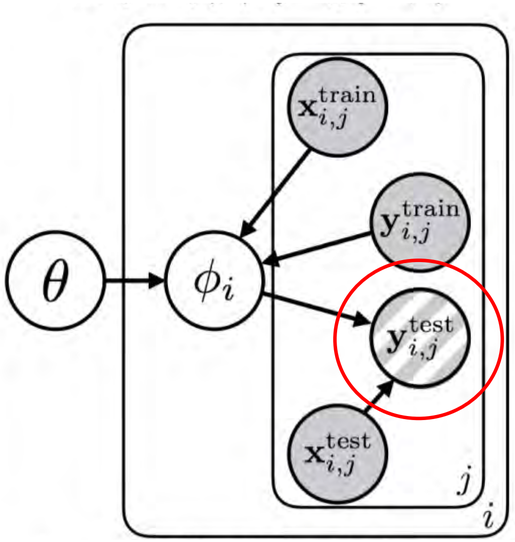


Finn*, Xu*, Levine. Probabilistic Model-Agnostic Meta-Learning

Training the model – not so simple!

$$\theta \sim p(\theta) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$$

key idea: $p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \delta(\hat{\phi}_i) \qquad \hat{\phi}_i \approx \theta + \alpha \nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$



Can't do ancestral sampling anymore!

need to figure out $p(\theta|x_i^{\text{test}}, y_i^{\text{test}})$ (intractable inference problem)

approximate using $q(\theta|x_i^{\text{test}}, y_i^{\text{test}})$, train via variational bound

$$\log p(x_i^{\text{train}}, y_i^{\text{train}}, x_i^{\text{test}}, y_i^{\text{test}}|\mu_\theta, \Sigma_\theta)$$

$$\leq E_{\theta\sim q}\left[\log \int p(\theta)p(\phi_i|\theta)p(y_i^{\text{train}}|x_i^{\text{train}}, \phi_i)p(y_i^{\text{test}}|x_i^{\text{test}}, \phi_i)d\phi_i - \log q(\theta|x_i^{\text{test}}, y_i^{\text{test}})\right]$$

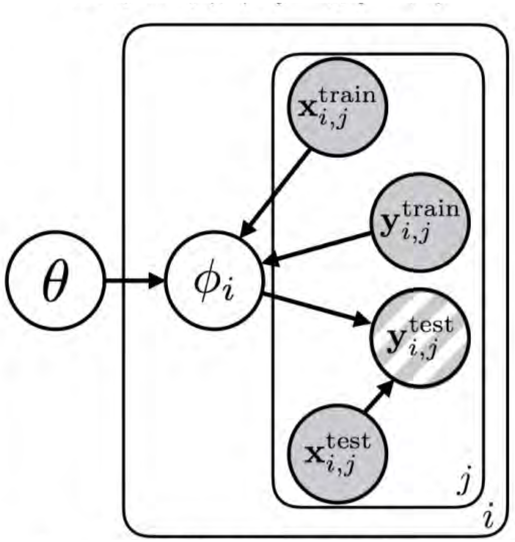
$$\approx E_{\theta\sim q}\left[\log p(\theta) + \log p(y_i^{\text{test}}|x_i^{\text{test}}, \hat{\phi}_i(\theta)) - \log q(\theta|x_i^{\text{test}}, y_i^{\text{test}})\right]$$

$$= E_{\theta\sim q}\left[\log p(y_i^{\text{test}}|x_i^{\text{test}}, \hat{\phi}_i(\theta))\right] - D_{\text{KL}}(q(\theta|x_i^{\text{test}}, y_i^{\text{test}})\|p(\theta))$$

Training the model with amortized inference

need to figure out $p(\theta|x_i^{\text{test}}, y_i^{\text{test}})$ (intractable inference problem)

approximate using $q(\theta|x_i^{\text{test}}, y_i^{\text{test}})$, train via variational bound



$$E_{\theta\sim q}\left[\log p(y_i^{\text{test}}|x_i^{\text{test}}, \hat{\phi}_i(\theta))\right] - D_{\text{KL}}(q(\theta|x_i^{\text{test}}, y_i^{\text{test}})\|p(\theta))$$

how do we represent q?

big neural network that outputs parameters? too intractable...

idea: $q(\theta|x_i^{\text{test}}, y_i^{\text{test}}) = \mathcal{N}(\mu_\theta + \alpha\nabla_{\mu_\theta}\log p(y_i^{\text{test}}|x_i^{\text{test}}, \mu_\theta), \Sigma_q)$

intuitive interpretation: *cheat* during meta-training by peeking at test set

...but minimize divergence against the prior

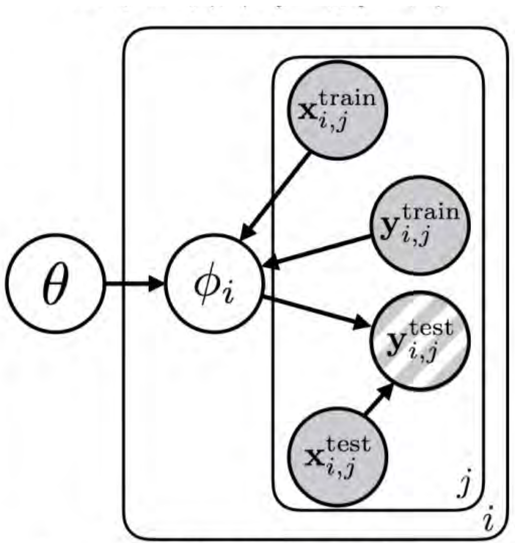
1. $\theta \leftarrow \mu_\theta + \alpha\nabla_{\mu_\theta}\log p(y_i^{\text{test}}|x_i^{\text{test}}, \mu_\theta) + \epsilon$
2. $\phi_i \sim p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha\nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$

Summary

need to figure out $p(\theta|x_i^{\text{test}}, y_i^{\text{test}})$ (intractable inference problem)

approximate using $q(\theta|x_i^{\text{test}}, y_i^{\text{test}})$, train via variational bound

$$E_{\theta\sim q}\left[\log p(y_i^{\text{test}}|x_i^{\text{test}}, \hat{\phi}_i(\theta))\right] - D_{\text{KL}}(q(\theta|x_i^{\text{test}}, y_i^{\text{test}})\|p(\theta))$$



During meta-training:

1. $\theta \leftarrow \mu_\theta + \alpha\nabla_{\mu_\theta} \log p(y_i^{\text{test}}|x_i^{\text{test}}, \mu_\theta) + \epsilon$
2. $\phi_i \sim p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha\nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$

take gradient step on variational bound w.r.t. θ, Σ, and Σ_q

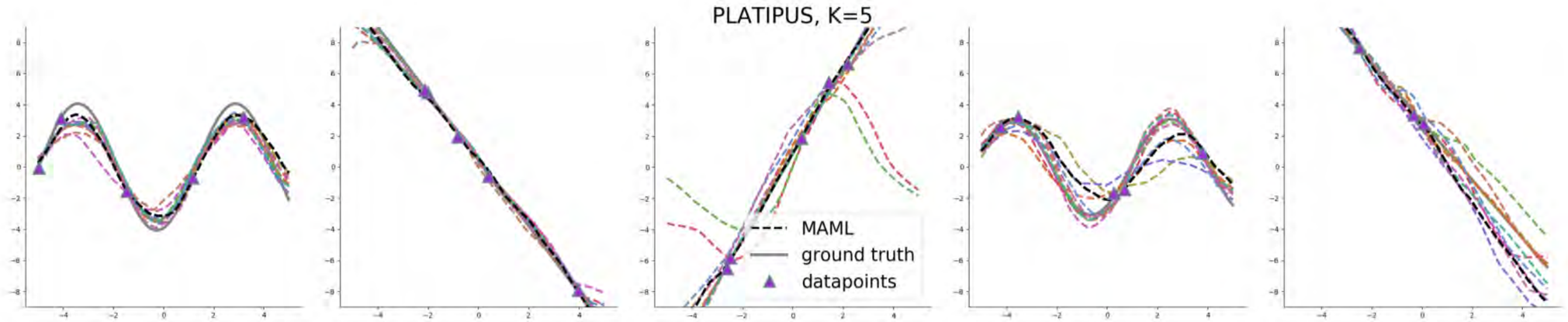
During meta-testing:

1. $\theta \sim \mathcal{N}(\mu_\theta, \Sigma_\theta)$
2. $\phi_i \sim p(\phi_i|\theta, x_i^{\text{train}}, y_i^{\text{train}}) \approx \hat{\phi}_i = \theta + \alpha\nabla_\theta \log p(y_i^{\text{train}}|x_i^{\text{train}}, \theta)$

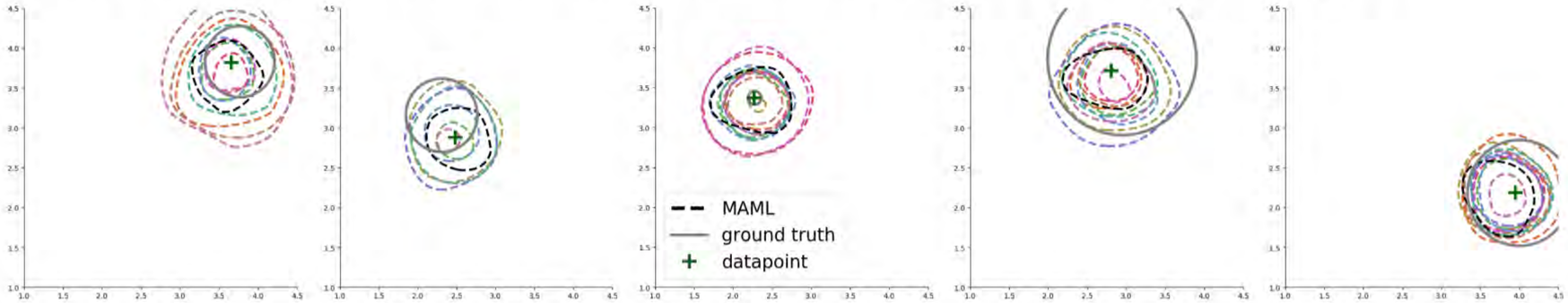
PLATIPUS

Probabilistic LATent model for Incorporating Priors and Uncertainty in few-Shot learning

Ambiguous regression:



Ambiguous classification:



Finn*, Xu*, Levine. Probabilistic Model-Agnostic Meta-Learning

PLATIPUS

Probabilistic LATent model for Incorporating Priors and Uncertainty in few-Shot learning

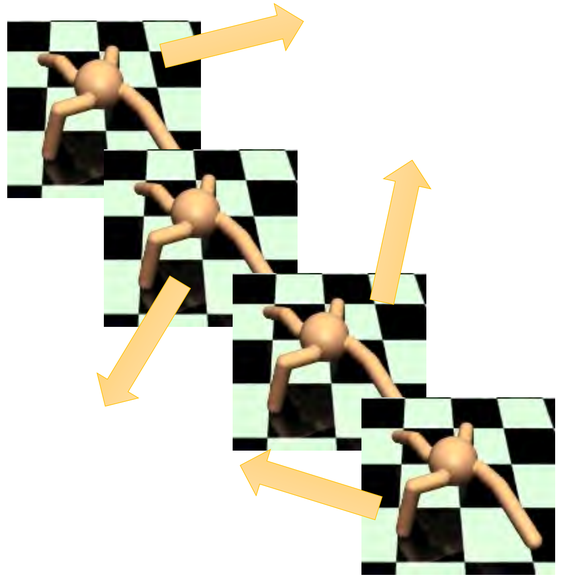
Ambiguous celebA (5-shot)		
	Accuracy	Coverage (max=3)
MAML	$69.26 \pm 2.18\%$	1.00 ± 0.0
MAML + noise	$54.73 \pm 0.8\ \%$	2.60 ± 0.12
PLATIPUS (ours)	$\mathbf{69.97 \pm 1.32\ \%}$	$\mathbf{2.62 \pm 0.11}$

Finn*, Xu*, Levine. Probabilistic Model-Agnostic Meta-Learning

Outline

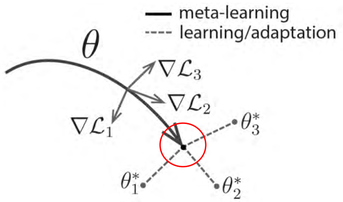
- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- **Meta-Learning with Automated Task Proposals**
- Extensions to Robot Imitation & Intent Inference

Let’s Talk about Meta-Overfitting

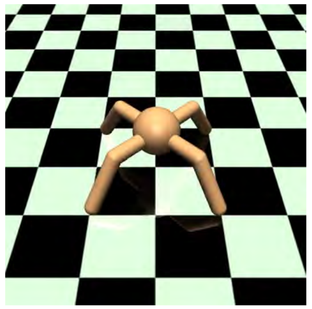


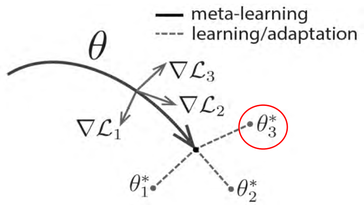
- Meta learning requires task distributions
- When there are too few meta-training tasks, we can *meta-overfit*
- Specifying task distributions is hard, especially for meta-RL!
- Can we propose tasks *automatically*?

after MAML training

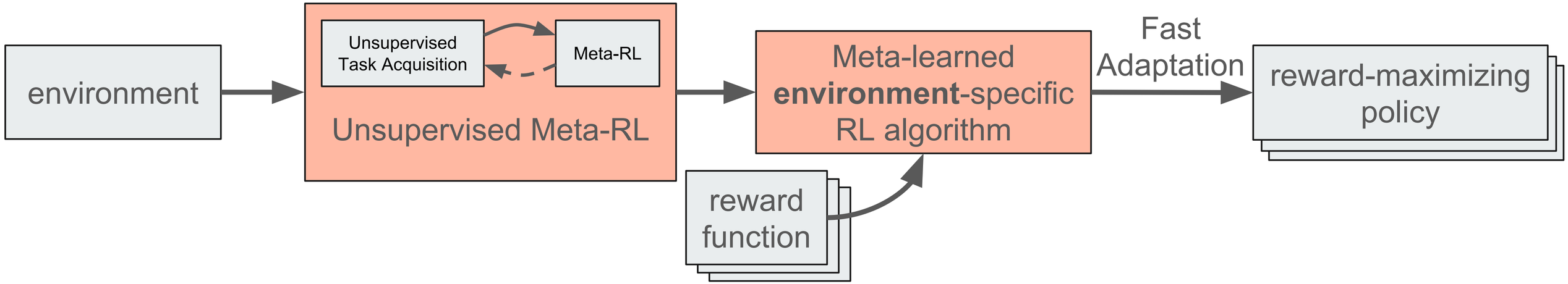


after 1 gradient step





A General Recipe for Unsupervised RL



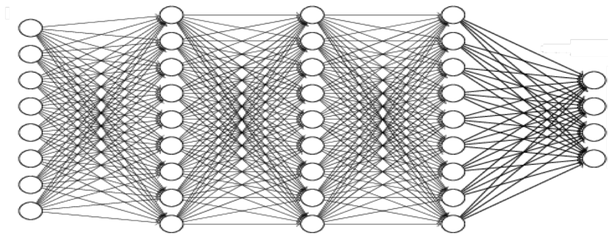
Gupta, Eysenbach, Finn, Levine. Unsupervised Meta-Learning for Reinforcement Learning.

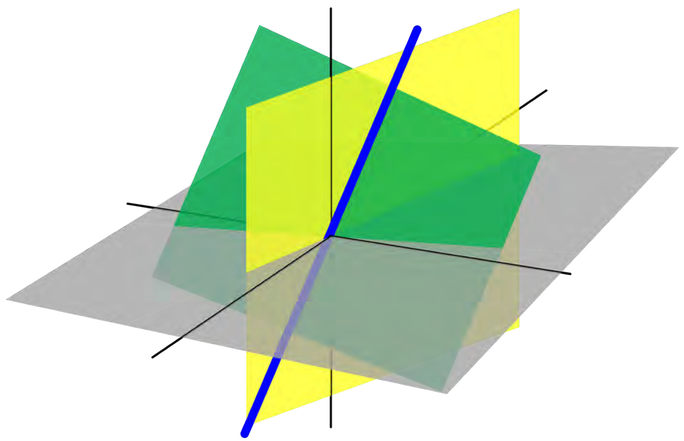
Random Task Proposals

- Use randomly initialize discriminators for reward functions

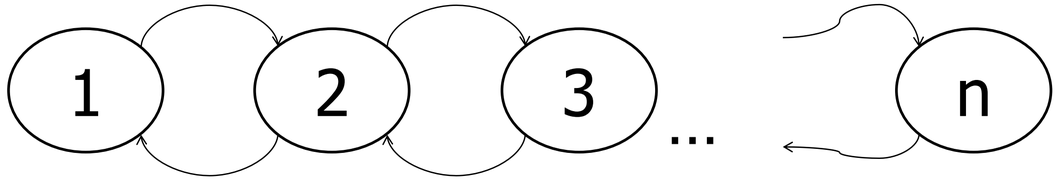
$$R(s, z) = \log p_D(z|s)$$

D → randomly initialized network



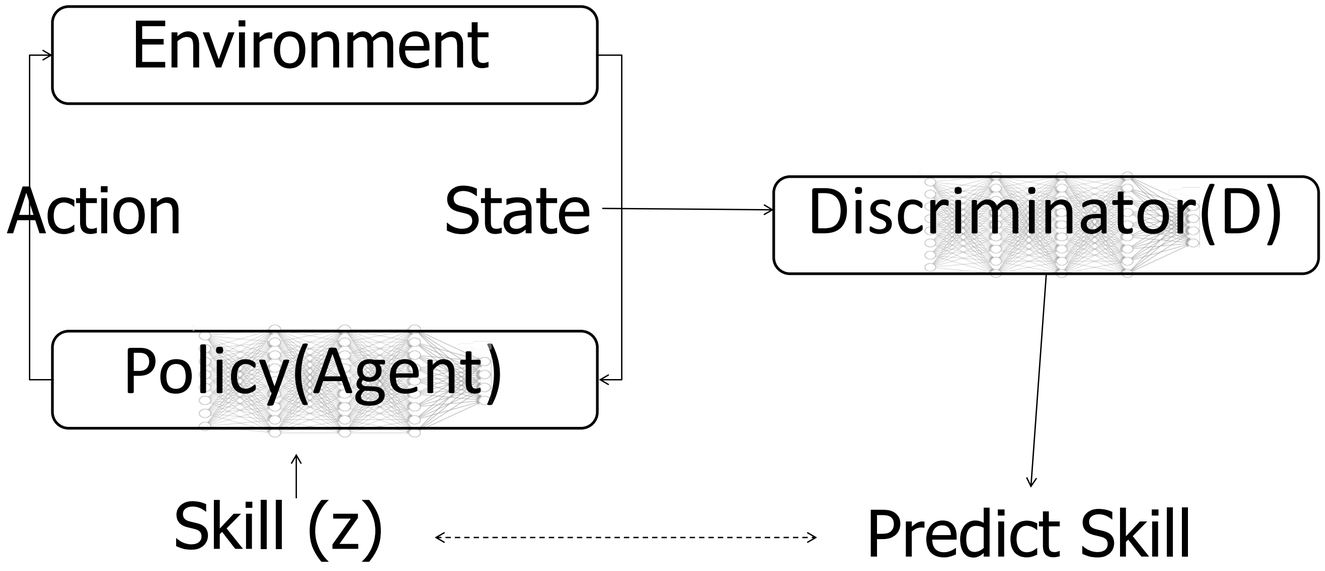


- Important: Random functions over state space, **not** random policies



Random policy – exponential
Random reward – polynomial

Diversity-Driven Proposals

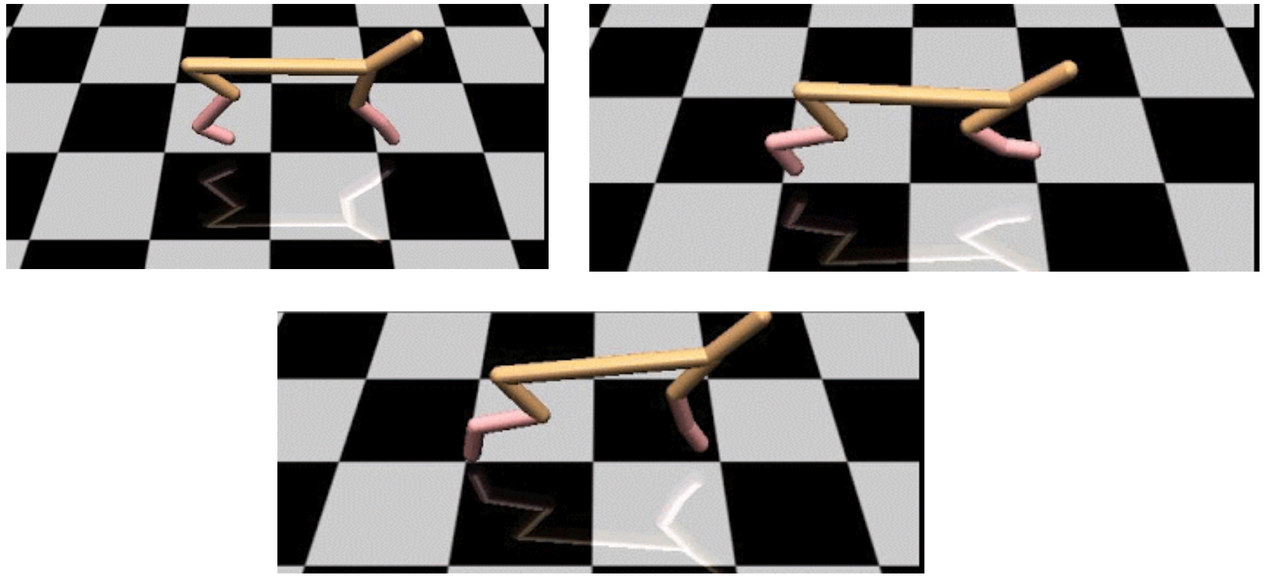


- Policy → visit states which are discriminable
- Discriminator → predict skill from state

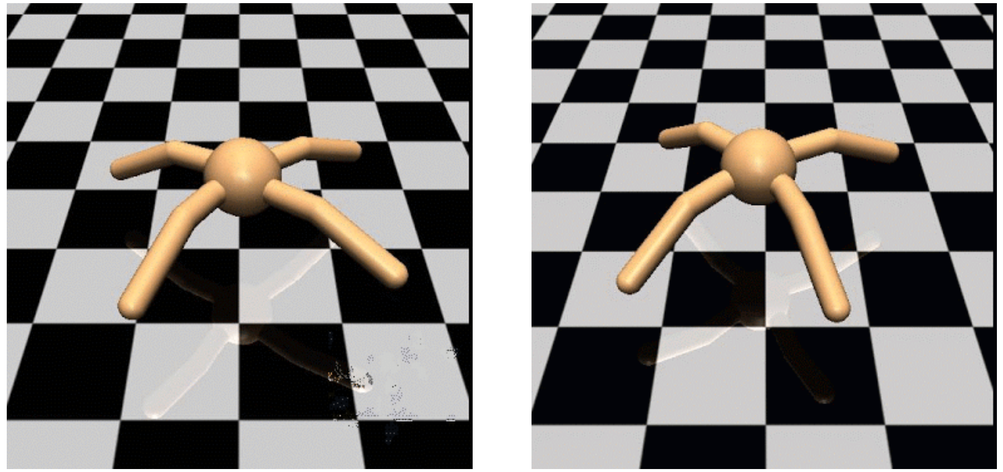
Task Reward for UML: $$R(s, z) = \log p_D(z|s)$$

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

Examples of Acquired Tasks



Cheetah

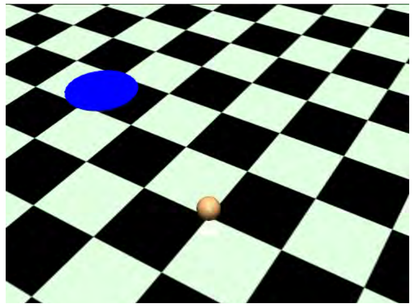


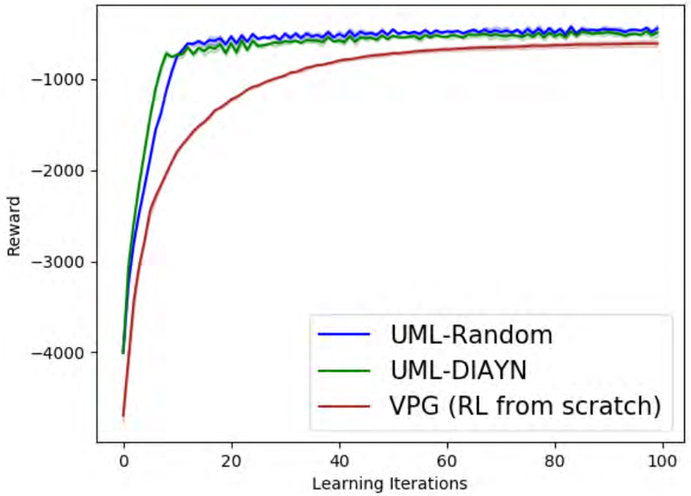
Ant

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

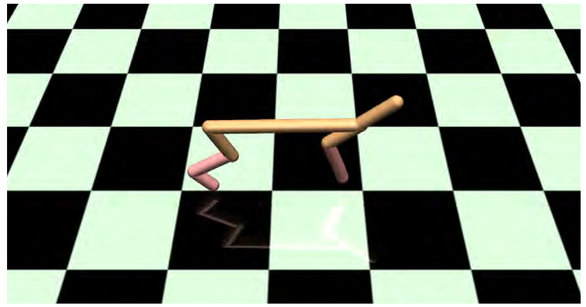
Does it work?

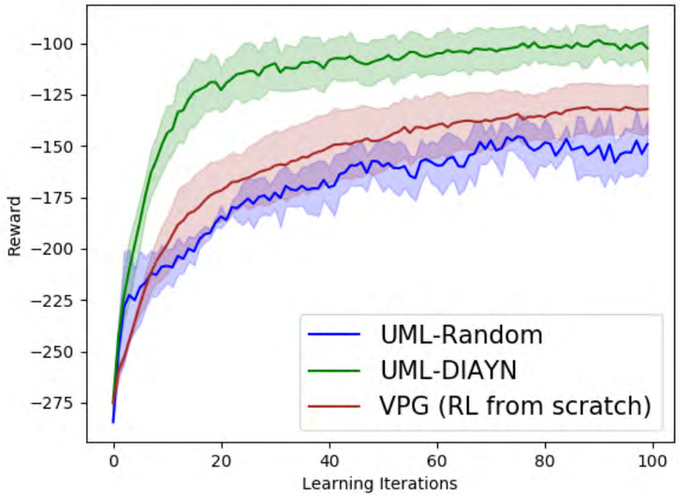
2D Navigation



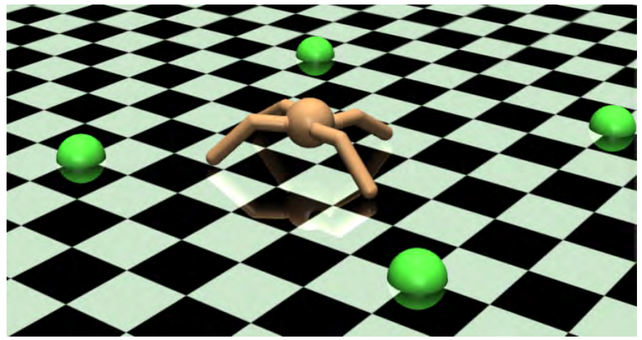


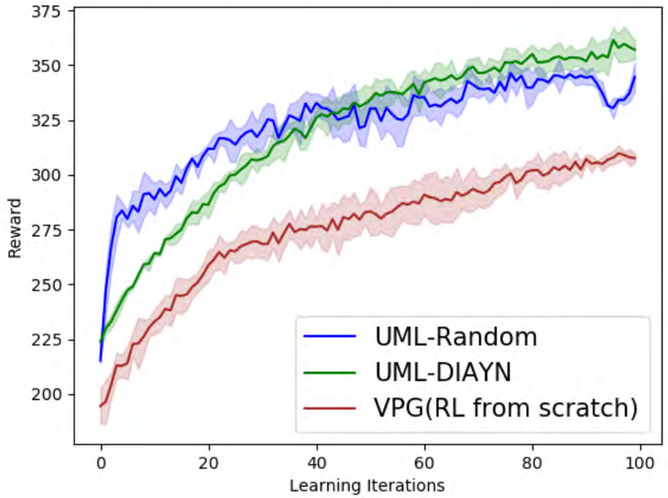
Cheetah





Ant



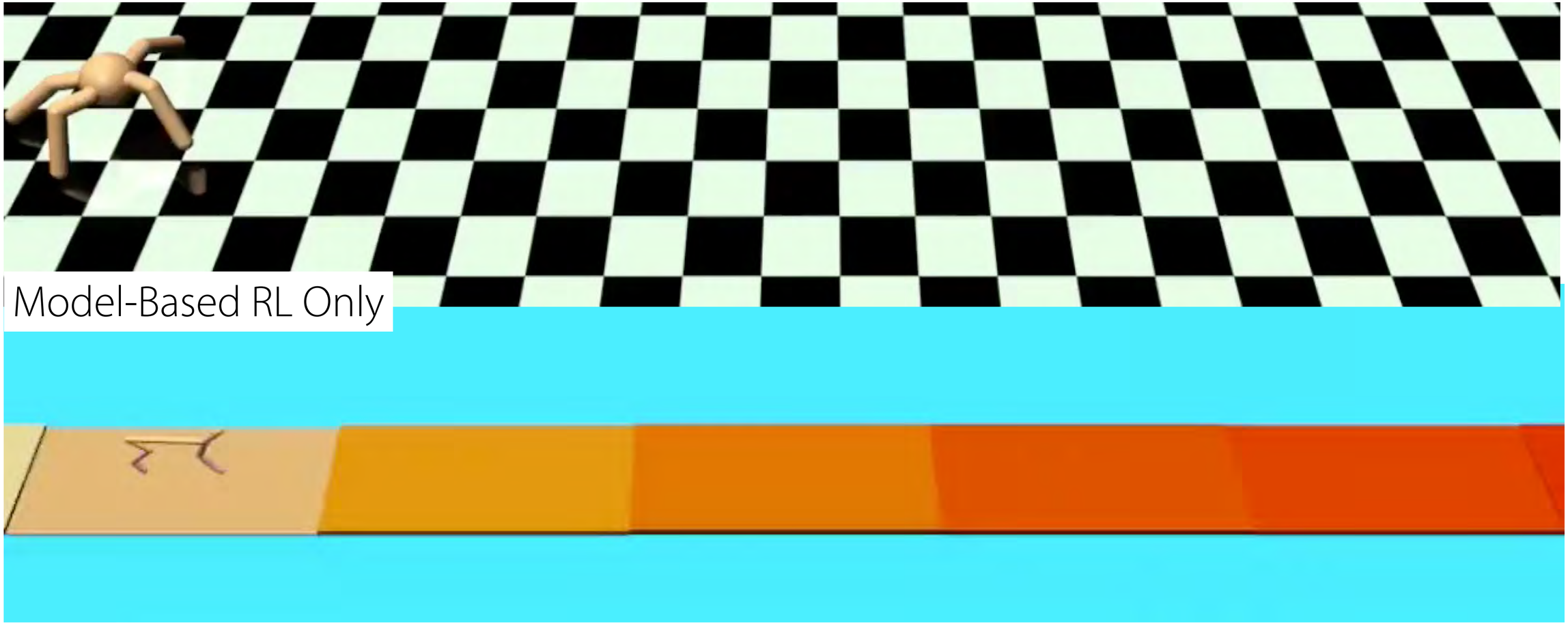


Meta-test performance with rewards

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

What about varying transition models?

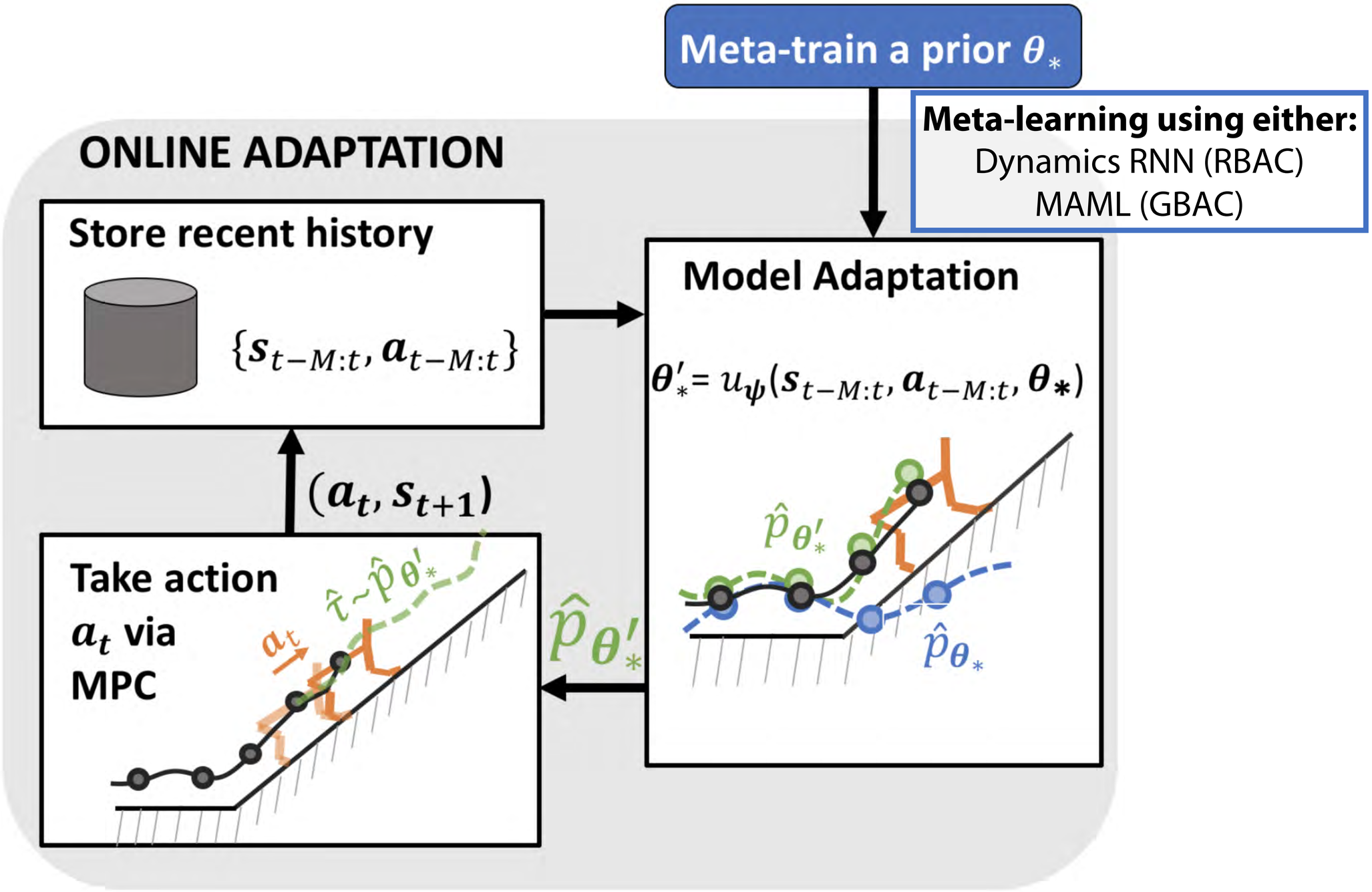
Task distribution is more-or-less free!



online adaptation = **few-shot learning**

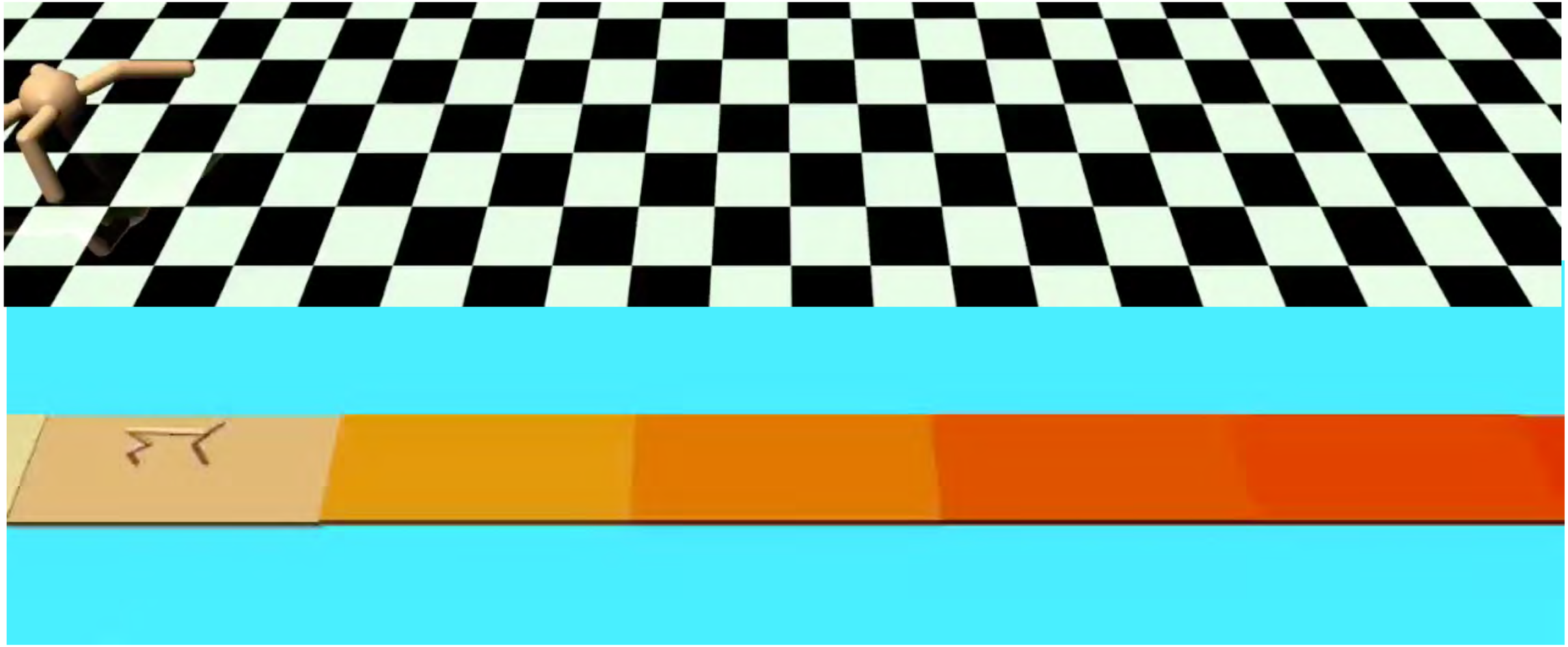
tasks are **temporal**

Clavera*, Nagabandi*, Abbeel, Levine, Finn, arxiv 2018


Meta-train a prior θ_*
Meta-learning using either:
Dynamics RNN (RBAC)
MAML (GBAC)
ONLINE ADAPTATION
Store recent history
$\{s_{t-M:t}, a_{t-M:t}\}$
Model Adaptation
$\theta'_* = u_\psi(s_{t-M:t}, a_{t-M:t}, \theta_*)$
$\hat{p}_{\theta'_*}$
$\hat{p}_{\theta_*}$
(a_t, s_{t+1})
Take action
a_t via
MPC
a_t
$\hat{\tau} \sim \hat{p}_{\theta'_*}$
$\hat{p}_{\theta'_*}$

Online Adaptation via Meta-Learning

with MAML



Clavera*, Nagabandi*, Abbeel, Levine, Finn, arxiv 2018

Outline

- Meta-Learning Problem Statement
- Model-Agnostic Meta-Learning (MAML)
- Probabilistic Interpretation of MAML
- Meta-Learning with Automated Task Proposals
- **Extensions to Robot Imitation & Intent Inference**

One-Shot Visual Imitation Learning

Goal: Given one visual demonstration of a new task, learn a policy

Visual imitation is expensive.

Rahmanizadeh et al. '17 Zhang et al. '17

learns from raw pixels,

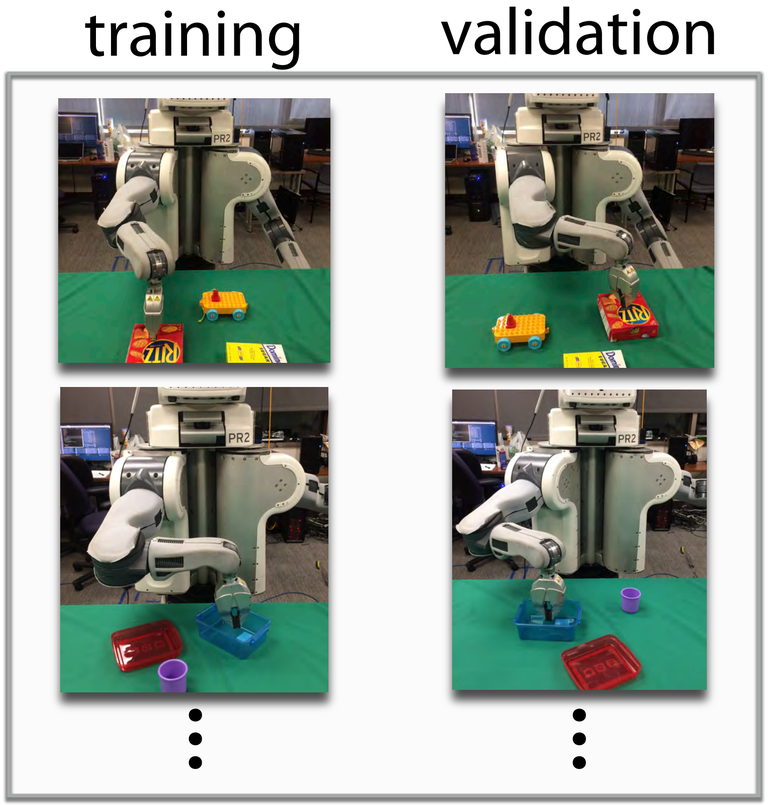
but requires many demonstrations

Through meta-learning: reuse data from other tasks/objects/environments

Finn*, Yu*, Zhang, Abbeel, Levine CoRL '17

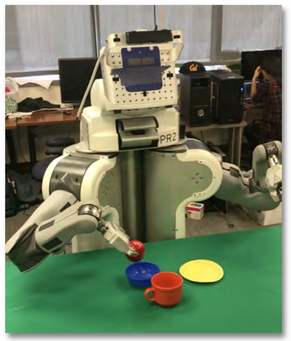
One-Shot Visual Imitation Learning

How? Learn to learn the policy from one demonstration



meta-training tasks

Meta-test time:



Learn policy from one demo.

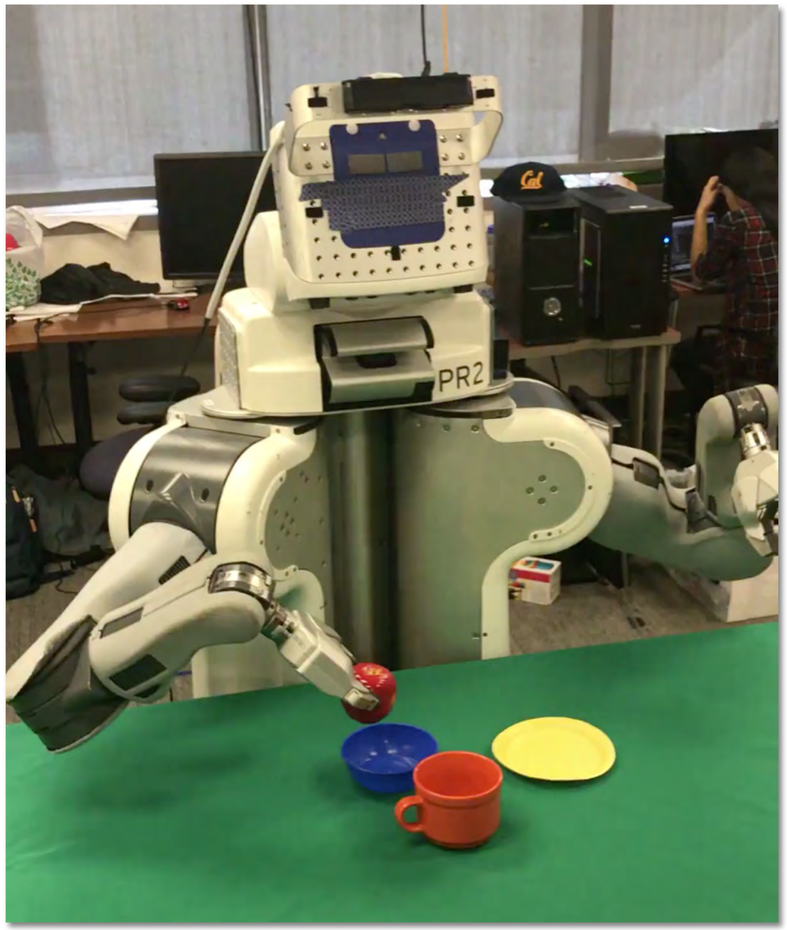
Finn*, Yu*, Zhang, Abbeel, Levine CoRL '17

real-world placing *from pixels*

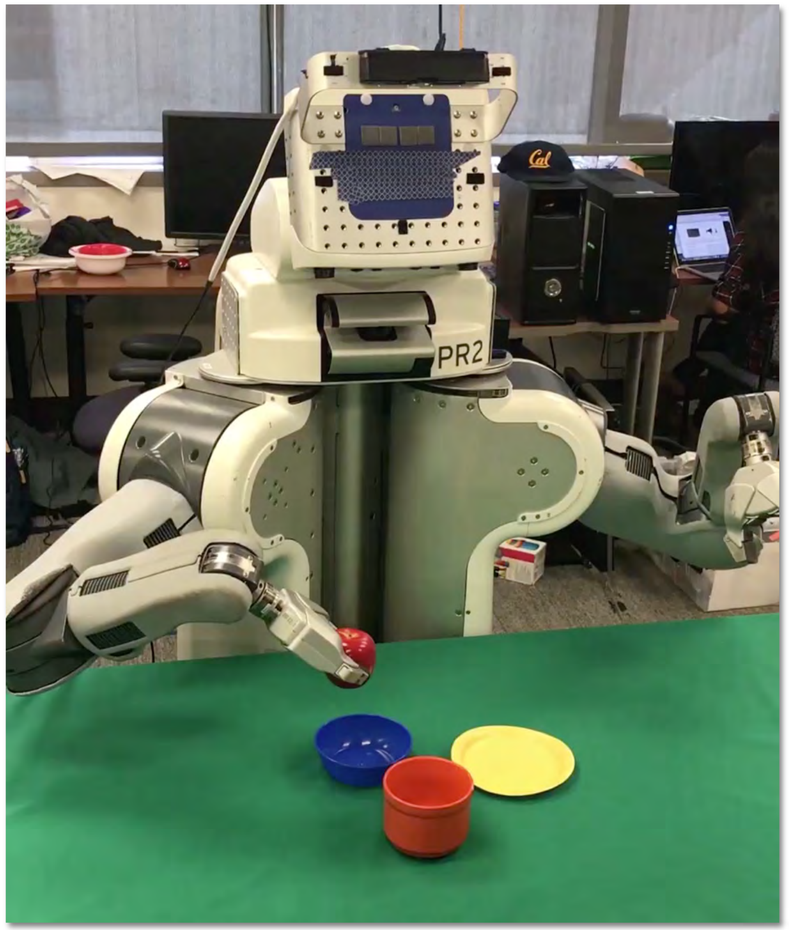
subset of
training objects

held-out test objects

input demo



resulting policy

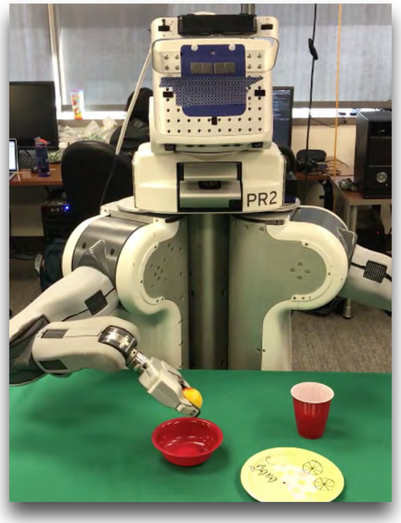


Finn*, Yu*, Zhang, Abbeel, Levine CoRL '17

Few-Shot Learning *from Weak Supervision*

~~Given one teleoperated demonstration:~~
Given a video of a human:

Learn a policy.



Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine RSS '18

~~Given 1 example of 5 classes:~~
Given 1 positive example:

Classify new examples

Grant, Finn, Peterson, Abbott, Levine, Darrell, Griffiths NIPS CIAI Workshop '17

Learning to Learn from Weak Supervision

meta-training

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta))$$

fully supervised **weakly supervised**

meta-test

$$\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$$

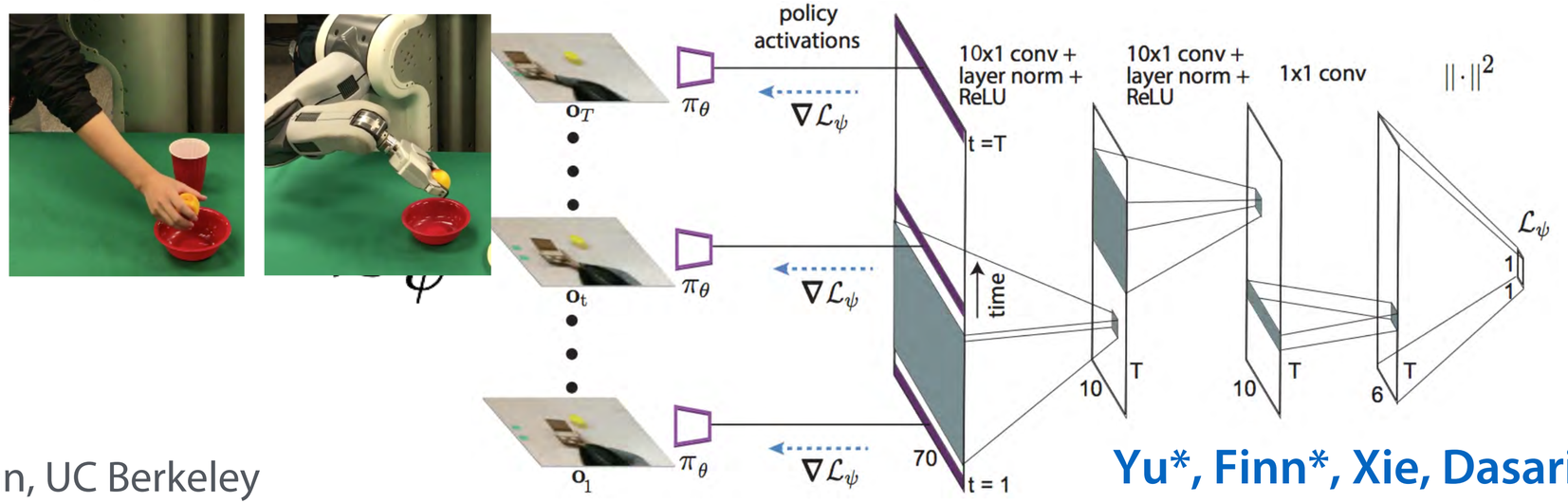
weakly supervised

What if the weakly supervised loss is unavailable?

imitation loss

$$\mathcal{L} = \sum_t \|\pi_\theta(o_t) - a^*_t\|^2$$

$$\min_{\theta,\psi} \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_\psi(\theta))$$



Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine RSS '18

placing *from pixels*

input demo

resulting policy

Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine RSS '18

pick-and-place *from pixels*

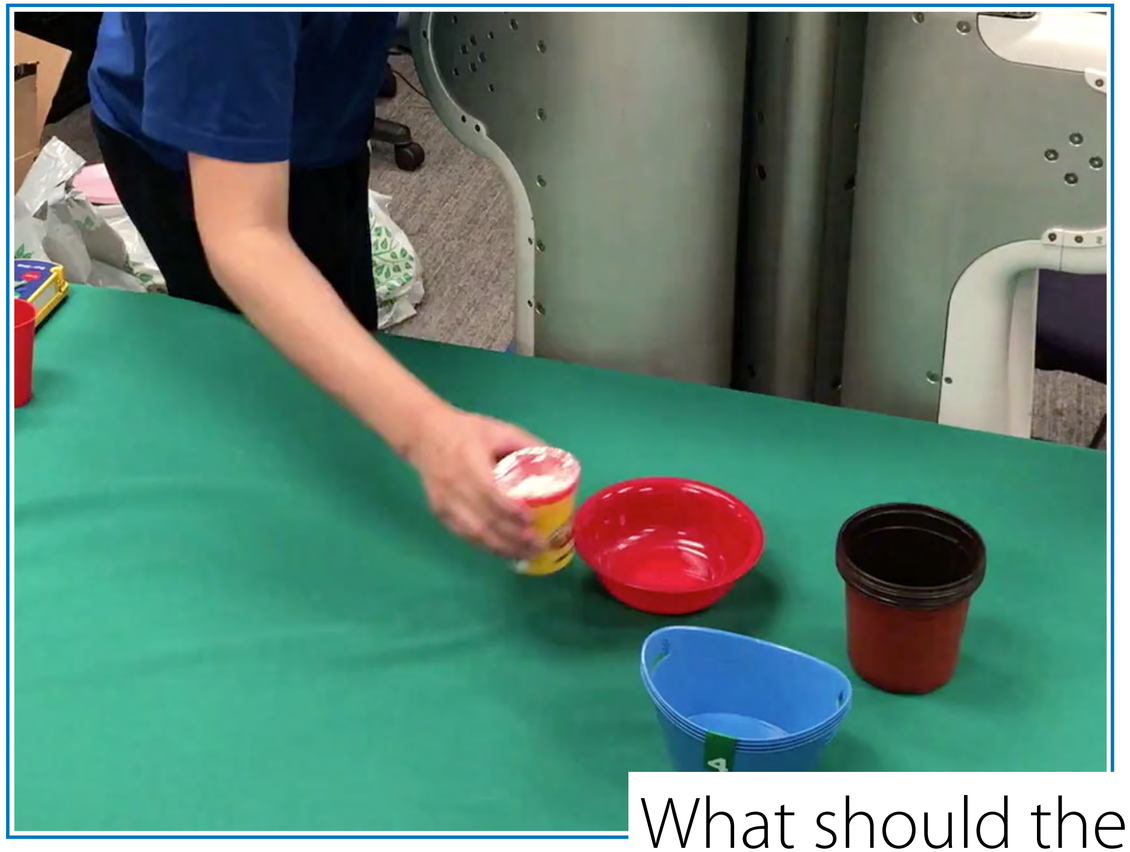
input demo

resulting policy

Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine RSS '18

pick-and-place *from pixels*

input demo



resulting policy

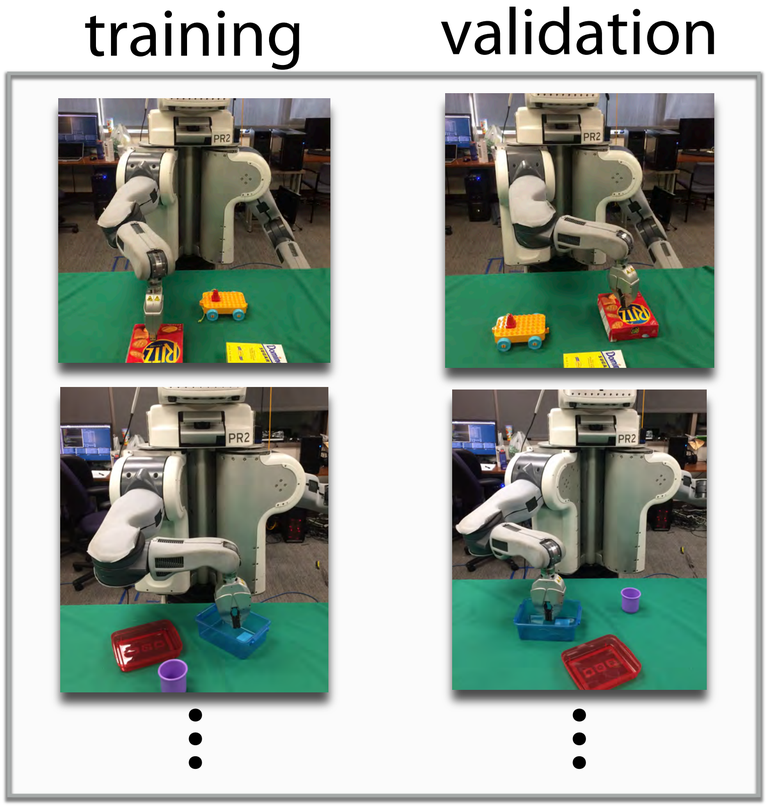
What should the robot do to improve?

Error in **inferring the goal** or **executing on it?**

Can we decouple **goal inference** & **policy learning**?

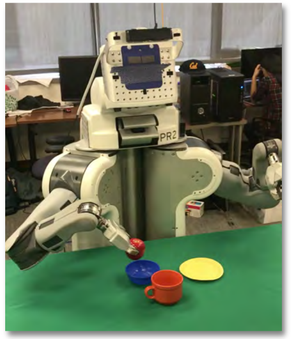
Meta-Learning Approach

How? Learn to learn the policy from one demonstration



meta-training tasks

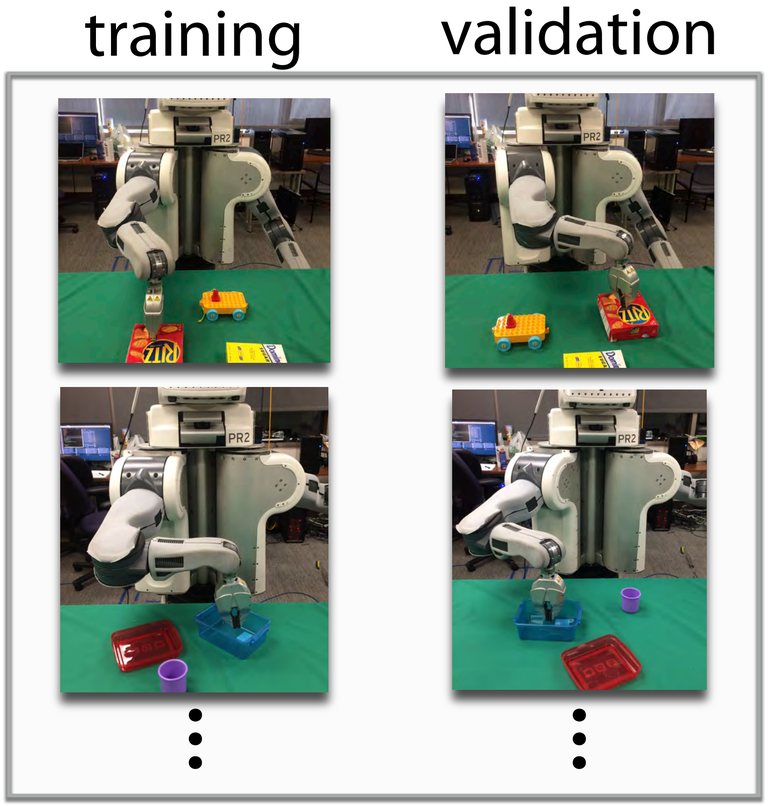
Meta-test time:



Learn policy from one demo.

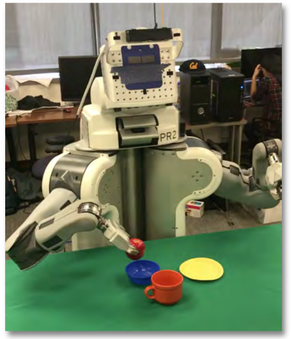
Meta-Learning Approach

How? Learn to **infer the objective** from one demonstration



meta-training tasks

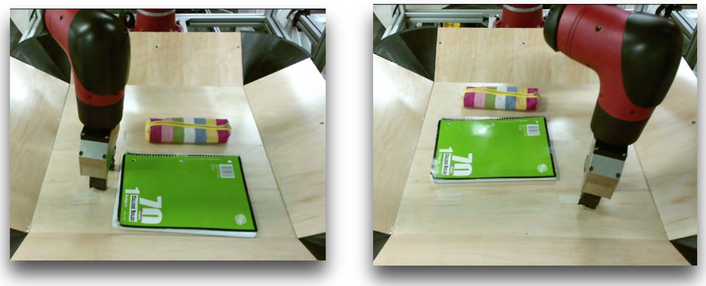
Meta-test time:



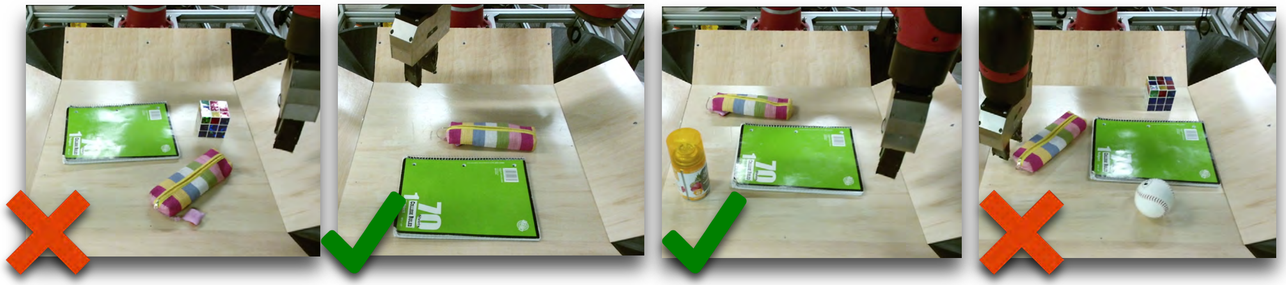
Learn **objective** from one demo.

Learn policy by optimizing reward.

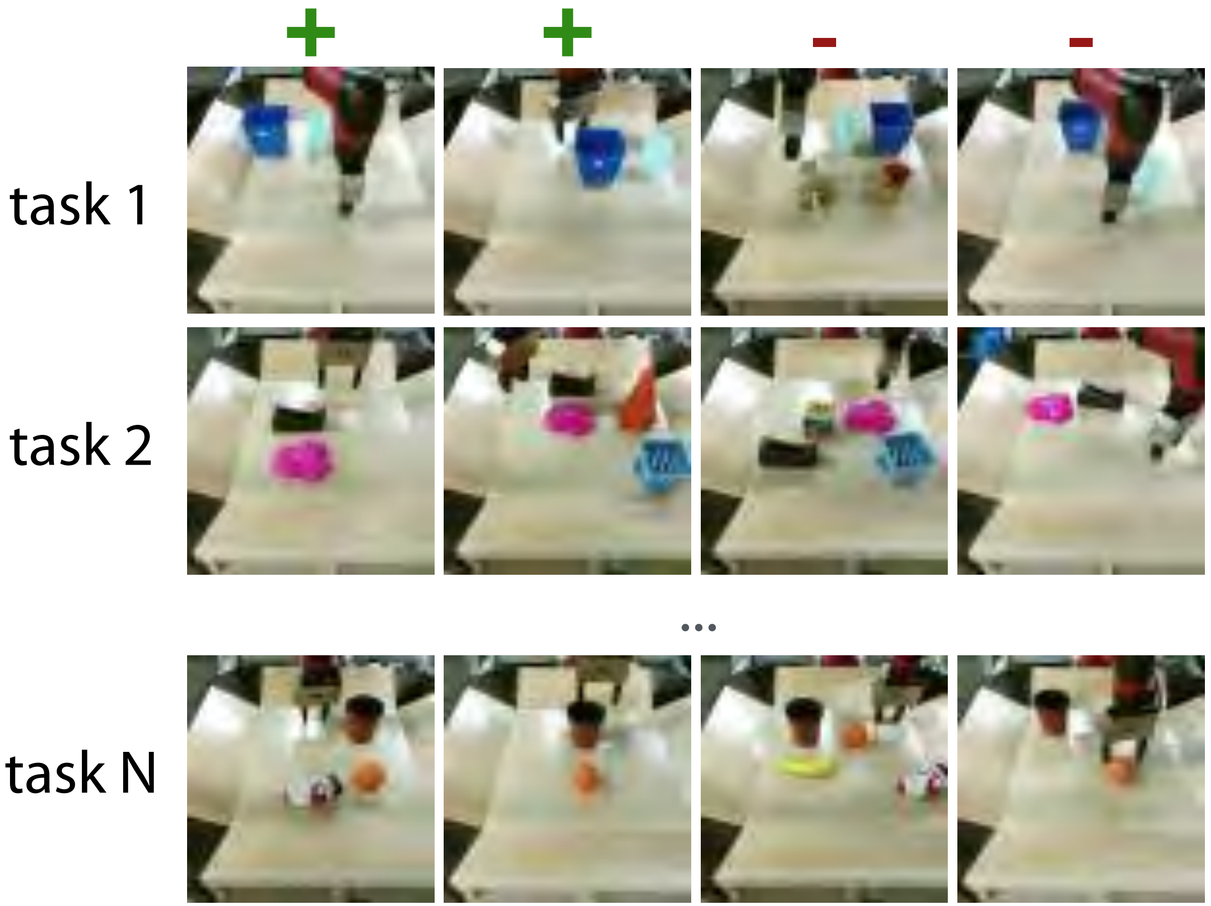
Given a few observations of reaching the goal:



Infer an objective:



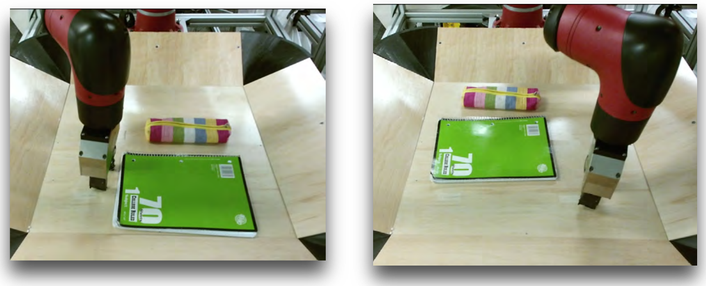
(then run RL or planning)



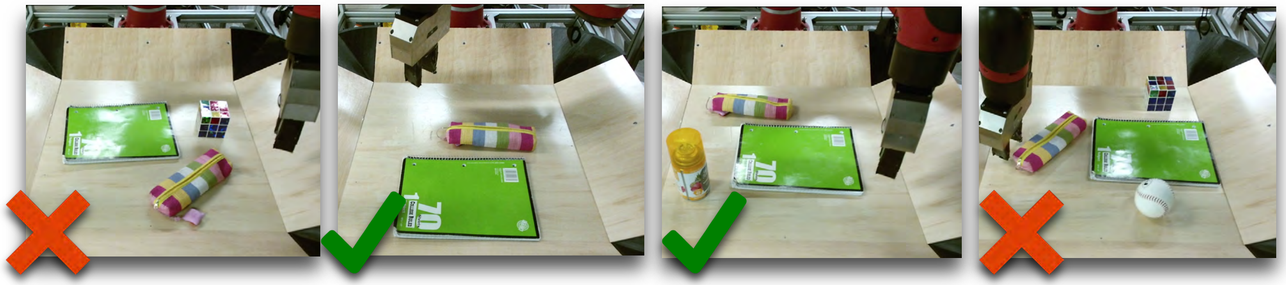
meta-training tasks

Xie, Singh, Levine, Finn '18 (under review)

Given a few observations of reaching the goal:



Infer an objective:



(then run RL or planning)

MAML with supervised loss:

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta))$$

both positive & negatives

only positive examples

another form of weak supervision!

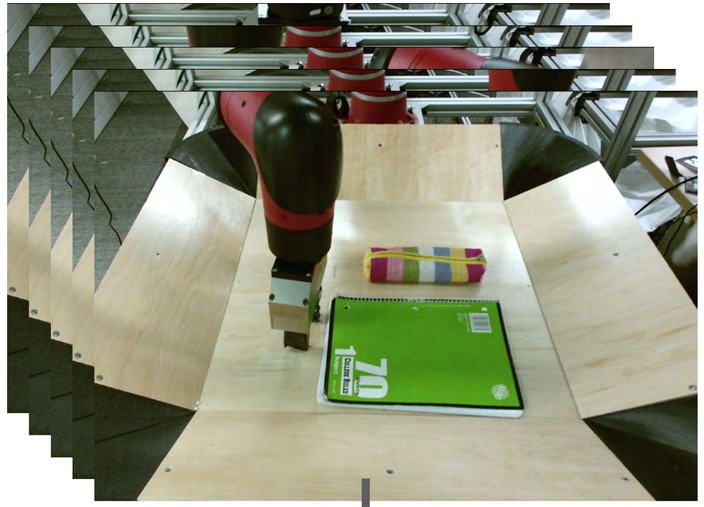
Test time: only need positive examples

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{train}}(\theta)$$

Xie, Singh, Levine, Finn '18 (under review)

Novel Object Positioning via Visual Planning

Given 5 examples of success



infer goal classifier

visual MPC w.r.t. goal classifier

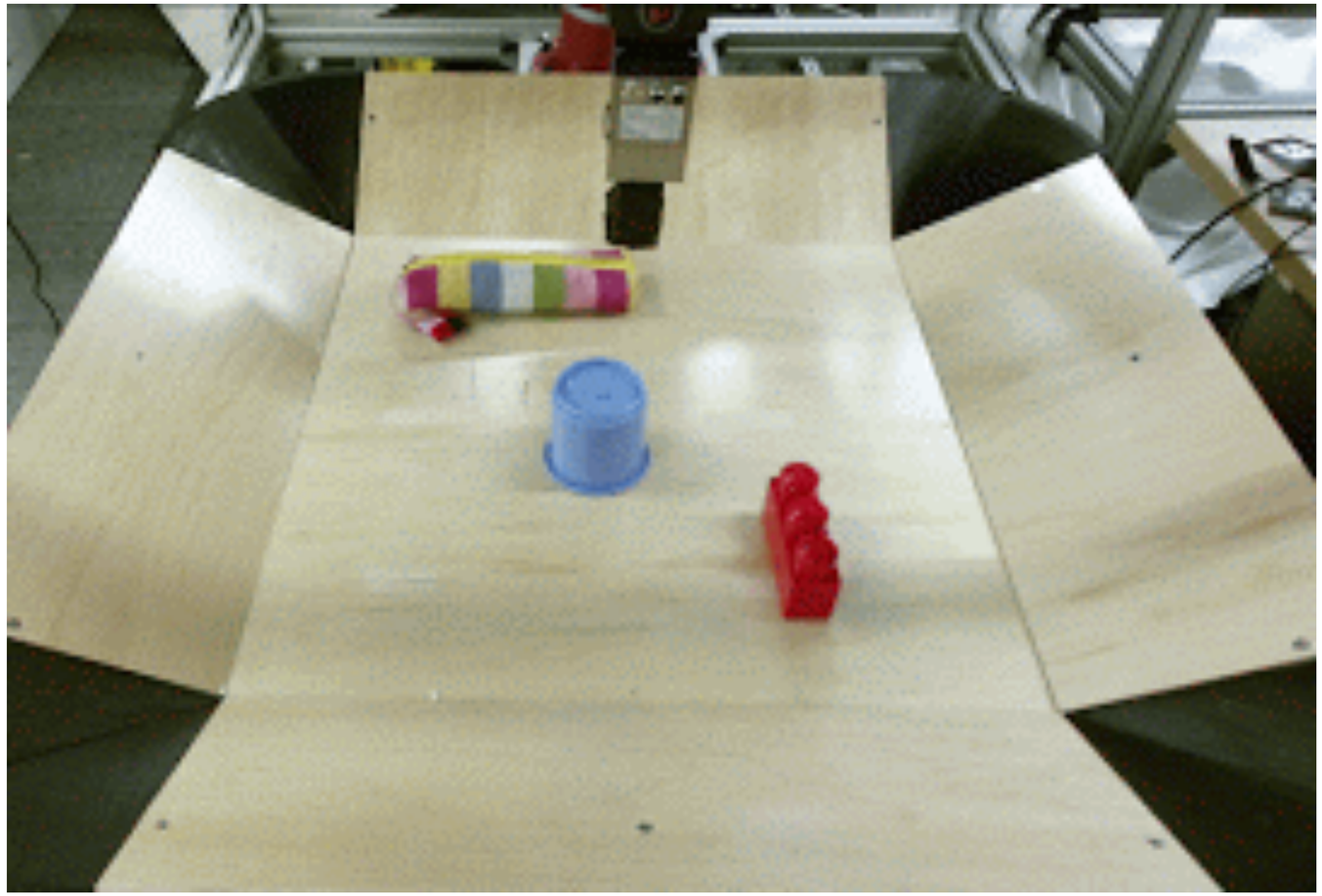
Visual MPC with learned objective

Xie, Singh, Levine, Finn ’18 (under review)

Novel Object Positioning via Visual Planning

Given 5 examples of success

Visual MPC with learned objective



Xie, Singh, Levine, Finn ’18 (under review)

Rope Manipulation via Reinforcement Learning

Given 5 examples of success

infer goal classifier

visual RL w.r.t. goal classifier

RL policy with learned objective

Xie, Singh, Levine, Finn '18 (under review)

Rope Manipulation via Reinforcement Learning

RL policy with learned objective

Given 5 examples of success

Xie, Singh, Levine, Finn ’18 (under review)

Questions?

Meta-Learning Foundations

Finn, Abbeel, Levine, *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*. ICML '17
Finn & Levine, *Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm*. ICLR '18
Grant, Finn, Levine, Darrell, Griffiths, *Recasting Gradient-Based Meta-Learning as Hierarchical Bayes*. ICLR '18
Finn*, Xu*, Levine, *Probabilistic Model-Agnostic Meta-Learning*. under review '18

Meta-Learning for Control

Finn*, Yu*, Zhang, Abbeel, Levine, *One-Shot Visual Imitation Learning via Meta-Learning*. CoRL '17
Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine, *One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning*. RSS '18
Clavera*, Nagabandi*, Fearing, Abbeel, Levine, Finn, *Learning to Adapt: Meta-Learning for Model-Based Control*. under review '18
Xu*, Ratner, Dragan, Levine, Finn, *Learning a Prior over Intent via Meta-Inverse Reinforcement Learning*. under review '18
Gupta, Eysenbach, Finn, Levine, *Unsupervised Meta-Learning for Reinforcement Learning*. under review '18
Xie, Singh, Levine, Finn, *Few-Shot Goal Inference for Visuomotor Learning and Planning*. under review '18