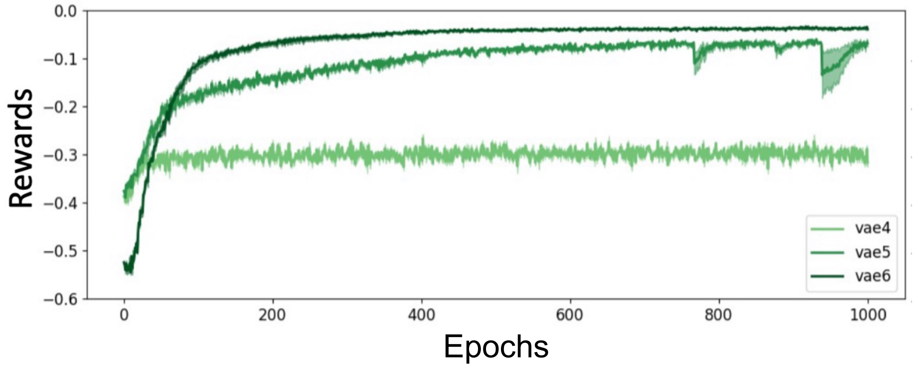
KTH ROYAL INSTITUTE OF TECHNOLOGY

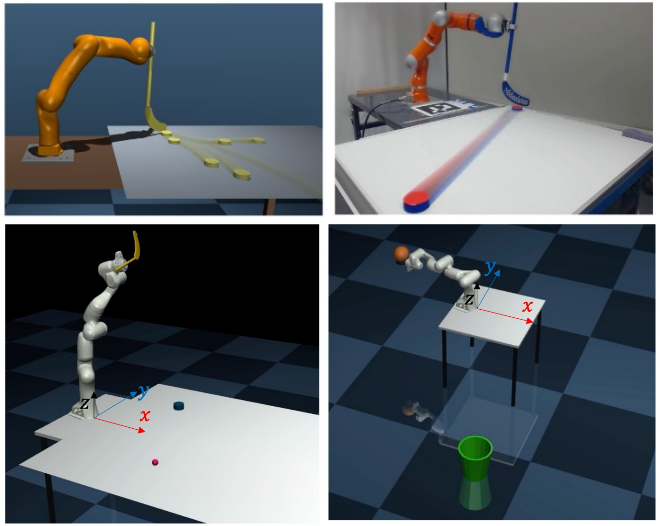
FDD3359 Reinforcement Learning Course
Meta Reinforcement Learning

Ali Ghadirzadeh

Why Meta Reinforcement Learning?

Reinforcement Learning
- trains a policy from scratch
- is usually very data-inefficient
- does not usually transfer well

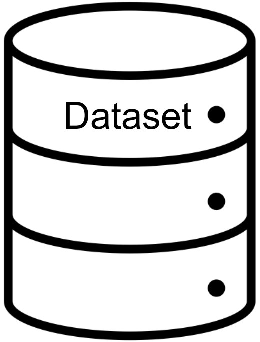




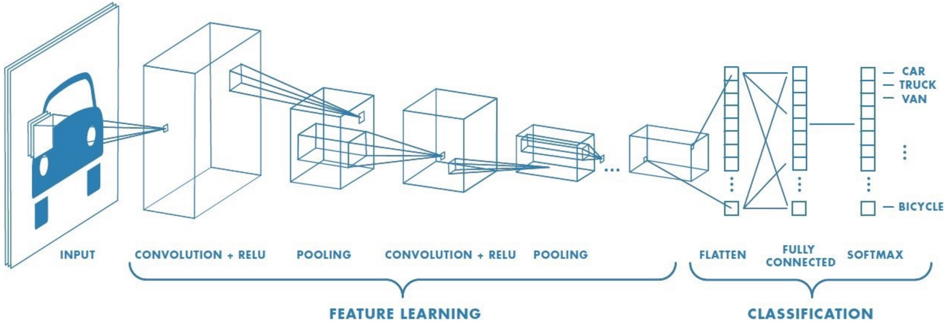
Leveraging prior knowledge to speed up solving new tasks

Meta Learning Background
Few-shot Supervised Learning

Standard Machine Learning



- A dataset
- A big neural network
- A loss function
- Some GPUs
- Stochastic gradient descent



Few-shot Learning
Training set consists of 5 classes

Classify novel test inputs

The Meta-Learning Problem
& Black-Box Meta-Learning
Prof. Chelsea Finn

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Some contents are borrowed from
"The Meta-Learning Problem & Black-Box Meta-Learning"
by Prof. Chelsea Finn

Transfer learning via fine-tuning

Parameters pre-trained on $\mathscr{D}_a$

$$\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\mathrm{tr}})$$

training data
for new task $\mathscr{T}_b$

(typically for many gradient steps)

Meta Learning Background

Some contents are borrowed from
“The Meta-Learning Problem & Black-Box Meta-Learning”
by Prof. Chelsea Finn

Supervised Learning

$$\phi_i = f_\theta(\mathcal{D}_i^{\text{tr}})$$

$$\mathbf{y}^{\text{ts}} = g_{\phi_i}(\mathbf{x}^{\text{ts}})$$

training classes

Given 1 example of 5 classes:

Classify new examples

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

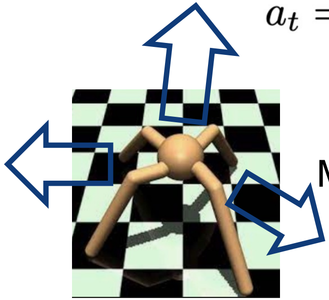
test set $\mathbf{x}_{\text{test}}$

$$\max_\theta \sum_{\mathcal{T}_i} \mathcal{L}(f_\theta(\mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{test}})$$

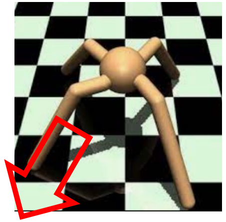
Reinforcement Learning

$$\phi_i = f_\theta(\{s_k, a_k, s_{k+1}, r_k\}_{k=0:T})$$

$$a_t = \pi_{\phi_i}(s_t)$$



Meta-training

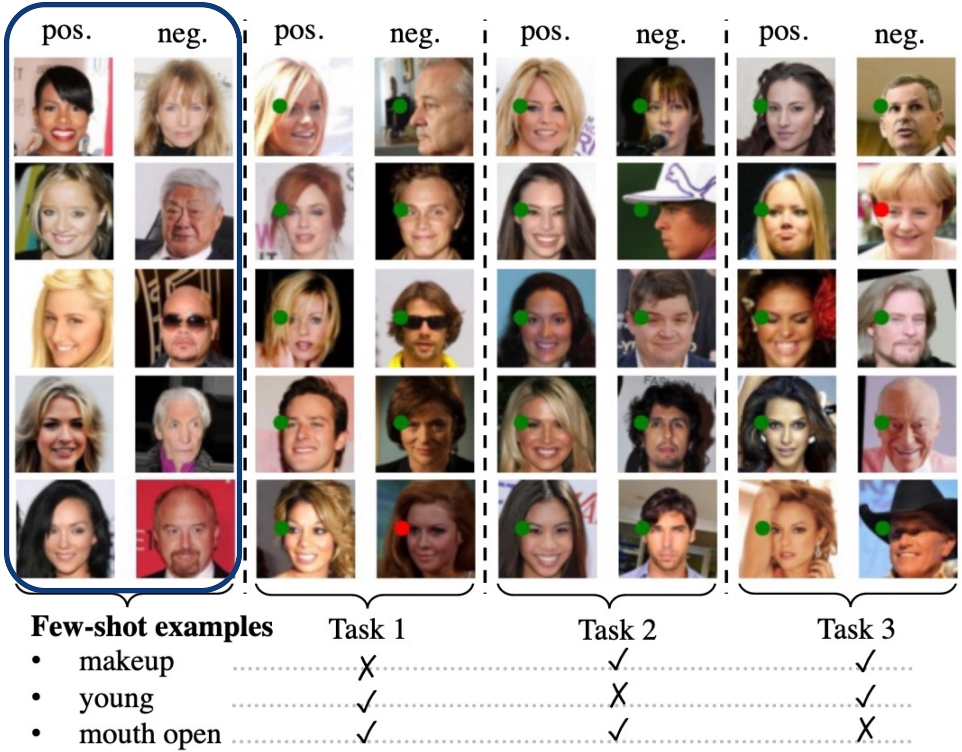


Meta-testing

$$\max_\theta \sum_{\mathcal{M}_i} \mathbb{E}_{\pi_{\phi_i}}[R(\tau)]$$

Meta Learning Background
Few-shot Supervised Learning

Training data



- Probabilistic Formulation
- Active Learning

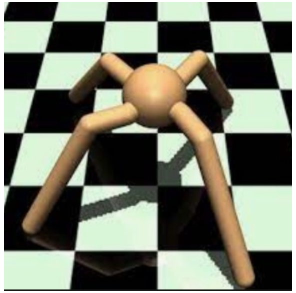
Meta Reinforcement Learning

Meta-learning
(Outer loop optimization)

Exploration Policy

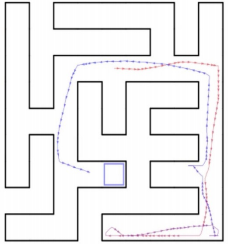
Task Policy

Adaptation
(Inner loop optimization)



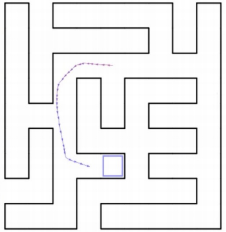
[meta] test time

Given a small amount of experience



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

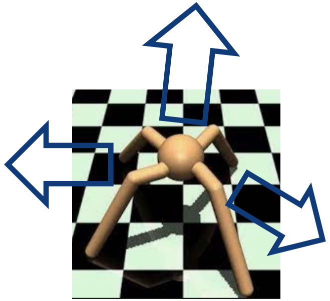
Learn to solve the task

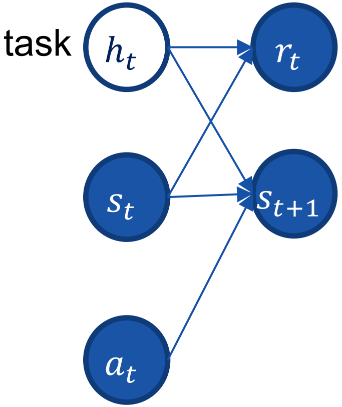


$\mathbf{s}_t \rightarrow \mathbf{a}_t$

Meta Reinforcement Learning

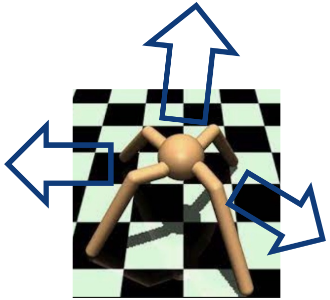
POMDP View

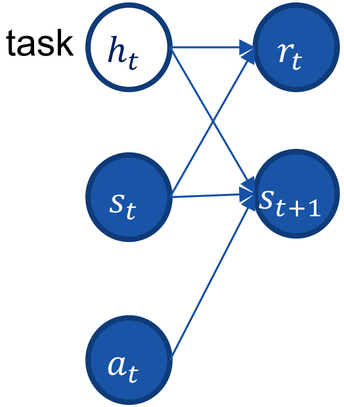




Meta Reinforcement Learning

POMDP View

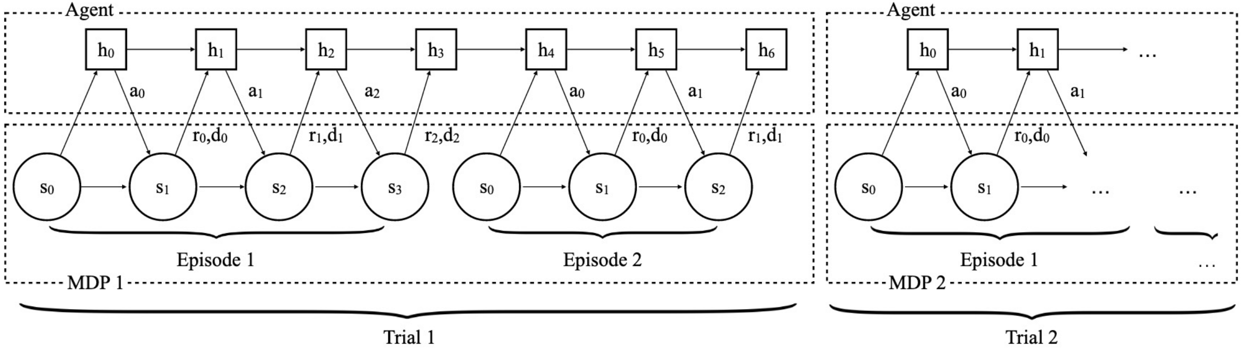




Recurrent Policy Training

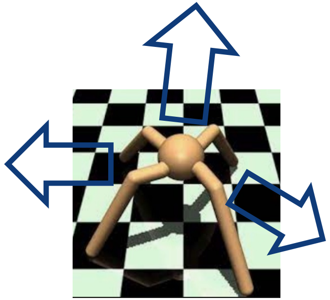
RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning

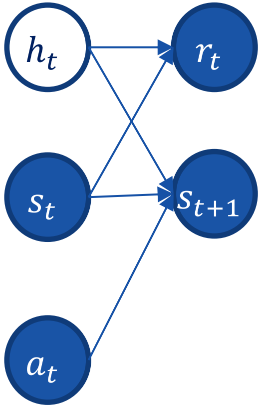
Yan Duan†‡, John Schulman†‡, Xi Chen†‡, Peter L. Bartlett†, Ilya Sutskever‡, Pieter Abbeel†‡



Meta Reinforcement Learning

POMDP View

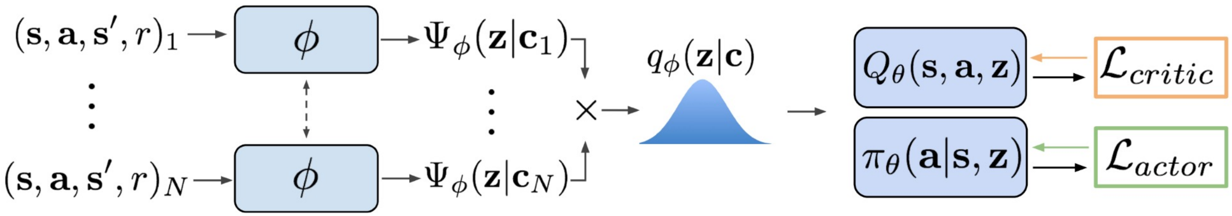




Latent-Variable Policy Training

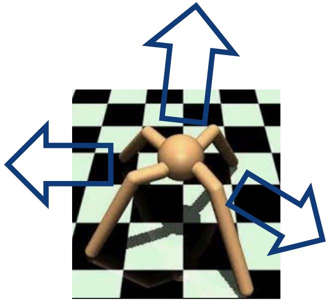
Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables

Kate Rakelly [1 *] Aurick Zhou [1 *] Deirdre Quillen [1] Chelsea Finn [1] Sergey Levine [1]



Some contents are borrowed from
“Optimization-Based Meta-Learning”
by Prof. Chelsea Finn

Optimization-Based Meta-Learning



Fine-tuning

pre-trained parameters

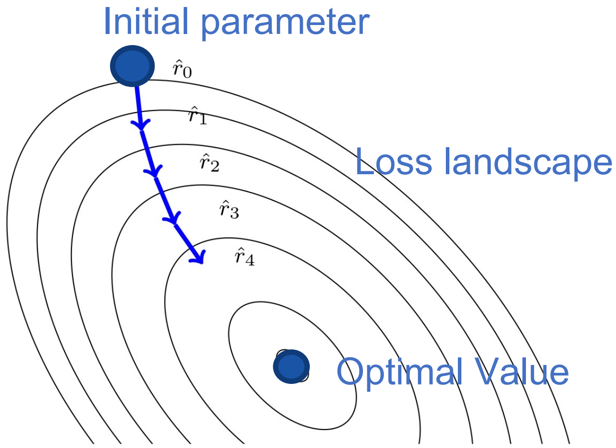
$$\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\mathrm{tr}})$$

training data
for new task

(typically for many gradient steps)

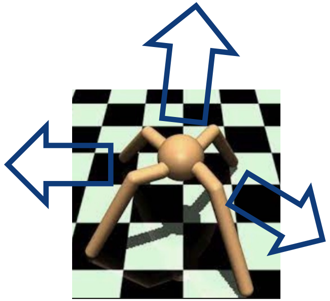
How to improve learning efficiency?

- Parameter Initialization θ
- Learning rate α
- Loss function $\mathcal{L}$
- Training data $\mathcal{D}^{tr}$



Optimization-Based Meta-Learning

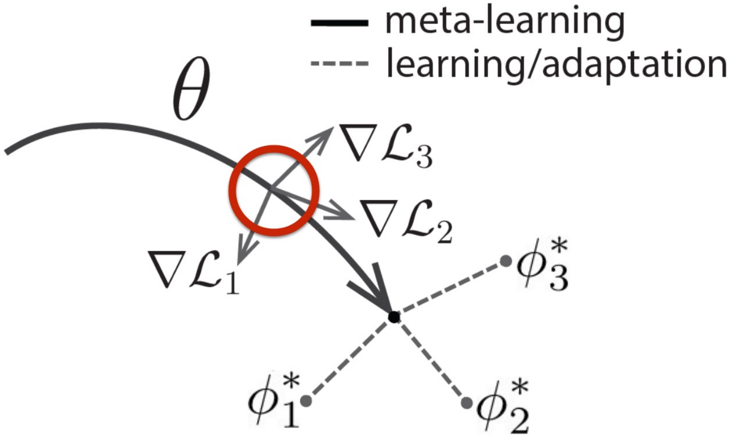
Some contents are borrowed from “Optimization-Based Meta-Learning” by Prof. Chelsea Finn



Meta-learning $\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$

θ parameter vector being meta-learned

ϕ_i^* optimal parameter vector for task i



Model-Agnostic Meta-Learning

Optimization-Based Meta-Learning

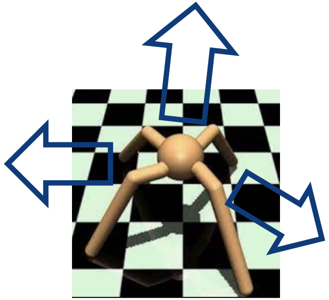
ProMP: Proximal Meta-Policy Search

Jonas Rothfuss* UC Berkeley, KIT jonas.rothfuss@gmail.com

Dennis Lee*, Ignasi Clavera* UC Berkeley {dennisl88,iclavera}@berkeley.edu

Tamim Asfour Karlsruhe Inst. of Technology (KIT) asfour@kit.edu

Pieter Abbeel UC Berkeley, Covariant.ai pabbeel@cs.berkeley.edu

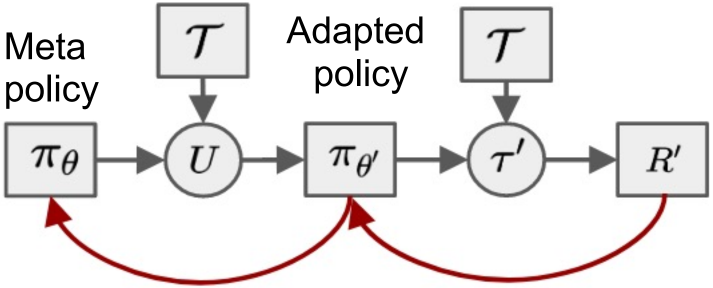


RL Objective

$$J = \mathbb{E}_{\tau \sim \pi_\theta}[\tau]$$

Policy Gradient

$$\nabla_\theta J_i(\theta) = E_{\tau \sim \pi_\theta, \mathcal{T}_i}\left[\left(\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\right)\left(\sum_t r_i(\mathbf{s}_t, \mathbf{a}_t)\right)\right]$$



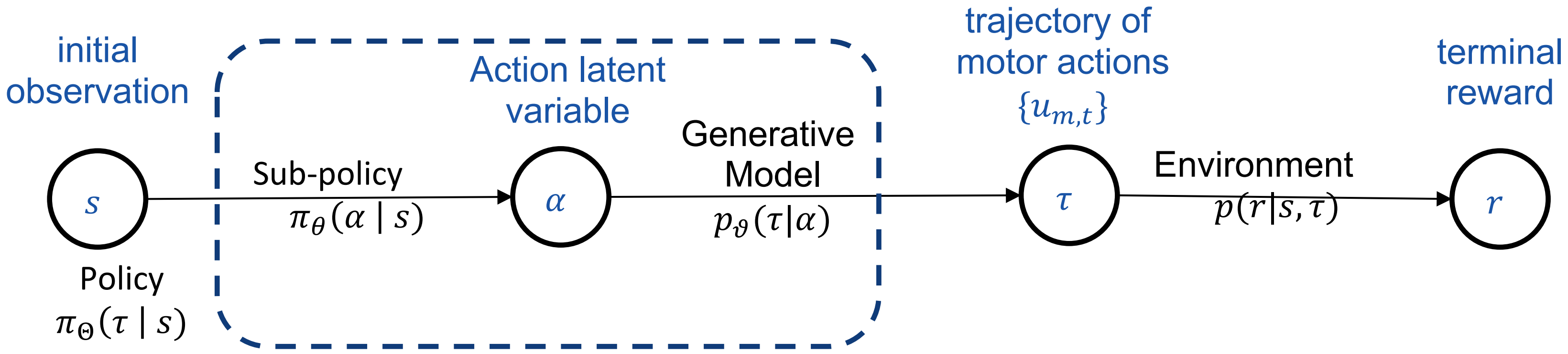
Credit assignment

$$\nabla_\theta J(\theta) = \mathbb{E}_{\mathcal{T} \sim \rho(\mathcal{T})}\left[\mathbb{E}_{\substack{\boldsymbol{\tau} \sim P_\mathcal{T}(\boldsymbol{\tau}|\theta) \\ \boldsymbol{\tau}' \sim P_\mathcal{T}(\boldsymbol{\tau}'|\theta')}}\left[\nabla_\theta J_{\text{post}}(\boldsymbol{\tau}, \boldsymbol{\tau}') + \nabla_\theta J_{\text{pre}}(\boldsymbol{\tau}, \boldsymbol{\tau}')\right]\right]$$

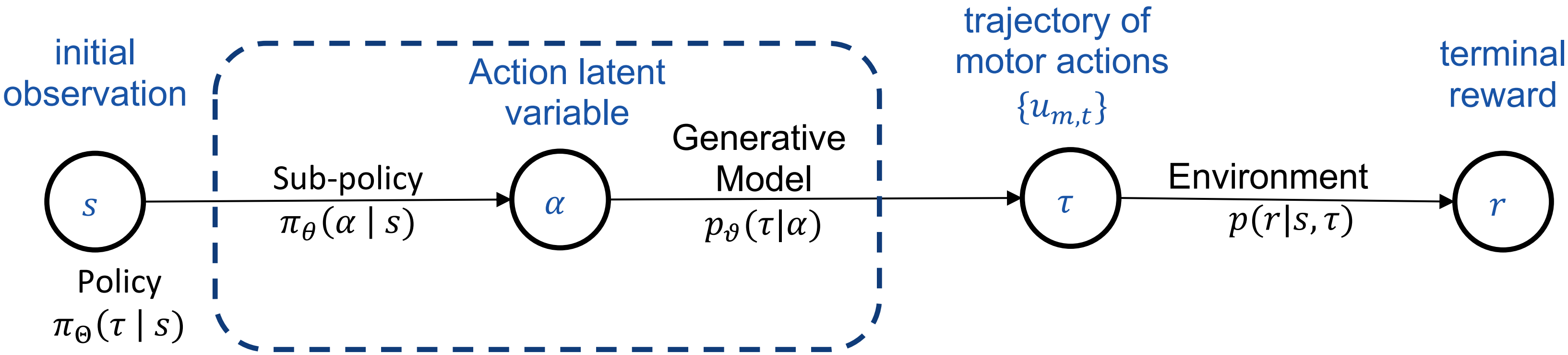
$$\nabla_\theta J_{\text{post}}(\boldsymbol{\tau}, \boldsymbol{\tau}') = \underbrace{\nabla_{\theta'} \log \pi_{\theta'}(\boldsymbol{\tau}') R(\boldsymbol{\tau}')}_{\nabla_{\theta'} J^{\text{outer}}} \underbrace{\left(I + \alpha R(\boldsymbol{\tau}) \nabla_\theta^2 \log \pi_\theta(\boldsymbol{\tau})\right)}_{\text{transformation from } \theta' \text{ to } \theta}$$

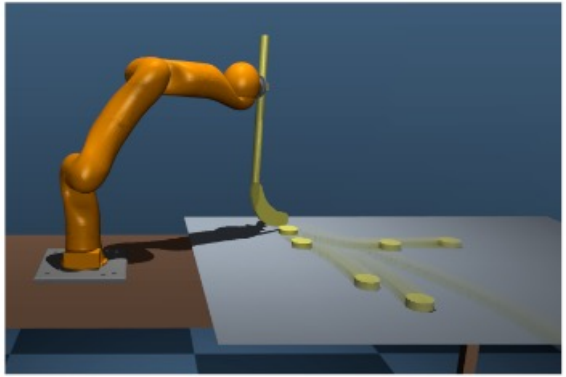
$$\nabla_\theta J^I_{\text{pre}}(\boldsymbol{\tau}, \boldsymbol{\tau}') = \alpha \nabla_\theta \log \pi_\theta(\boldsymbol{\tau}) \left(\underbrace{(\nabla_\theta \log \pi_\theta(\boldsymbol{\tau}) R(\boldsymbol{\tau}))^\top}_{\nabla_\theta J^{\text{inner}}} \underbrace{(\nabla_{\theta'} \log \pi_{\theta'}(\boldsymbol{\tau}') R(\boldsymbol{\tau}'))}_{\nabla_{\theta'} J^{\text{outer}}} \right)$$

RL with Generative Models



Sim-to-real Transfer Learning





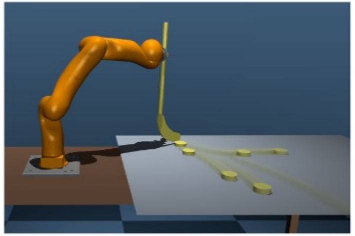
Options:

- Learn a general model
- Learn a learning agent

Arndt et al., ICRA20
Meta Reinforcement Learning for Sim-to-real Domain Adaptation.

Learning a dynamic task efficiently?

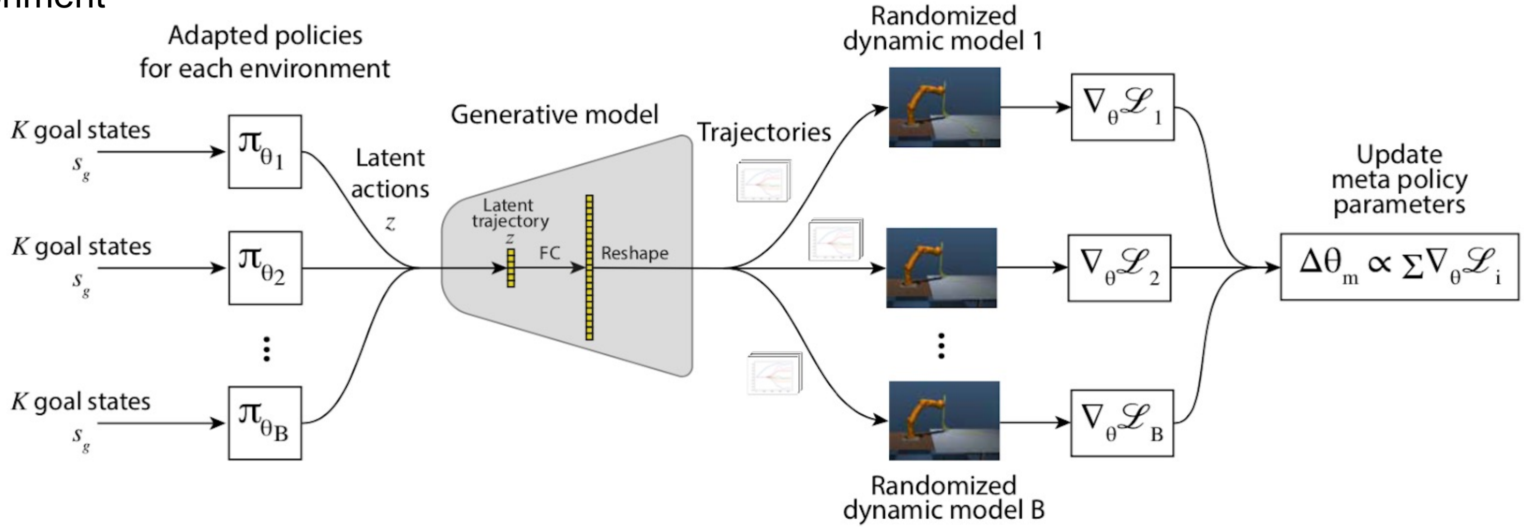
Sim-to-real Transfer Learning



Sim environment

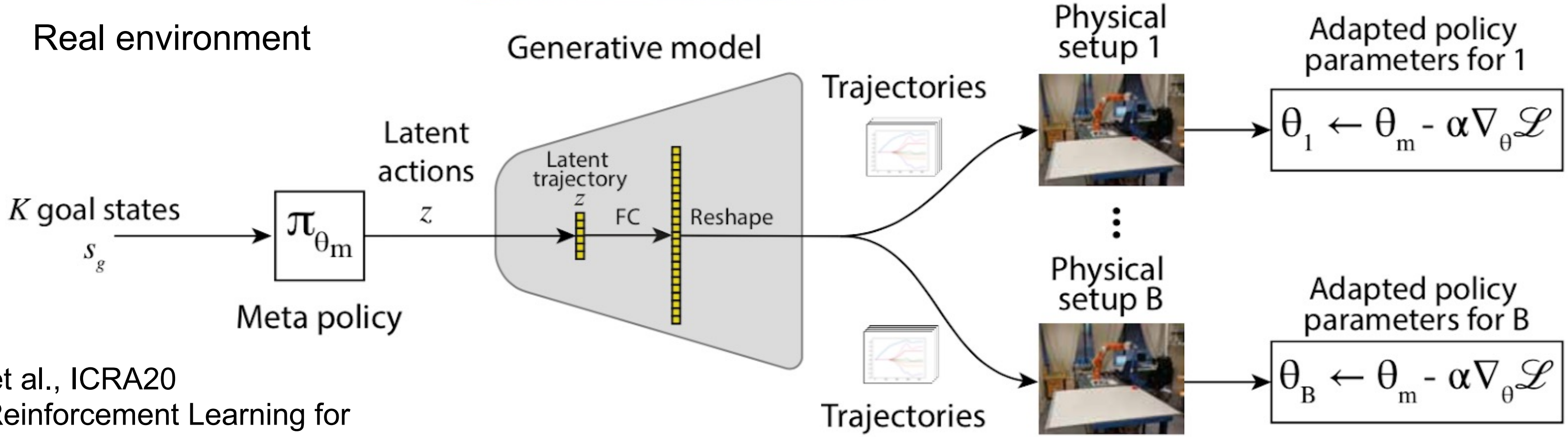
TABLE I: Randomized parameters

Parameter	Minimum	Maximum
x linear friction (μ_x)	0.15	0.95
y linear friction (μ_y)	$0.7\mu_x$	$1.3\mu_x$
Torsional friction (μ_τ)	0.001	0.05
Rotational friction x (μ_{rx})	0.01	0.3
Rotational friction y (μ_{ry})	0.01	0.3
Puck mass	50g	500g
Initial puck position	$\epsilon \sim \mathcal{N}(0, 0.02)$	



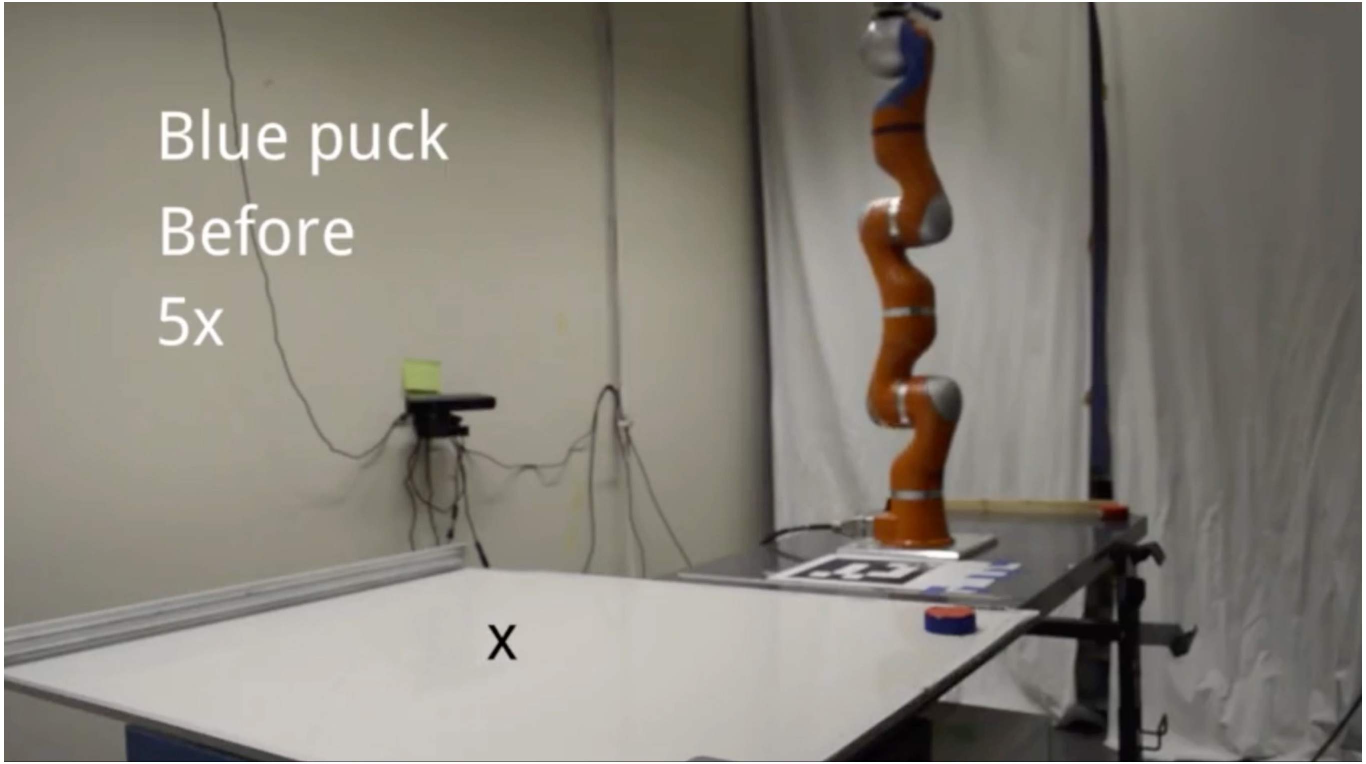
Sim-to-real Transfer Learning

Real environment

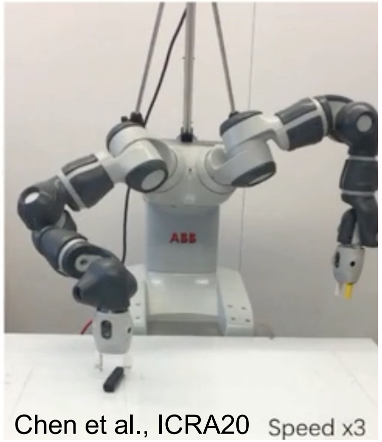


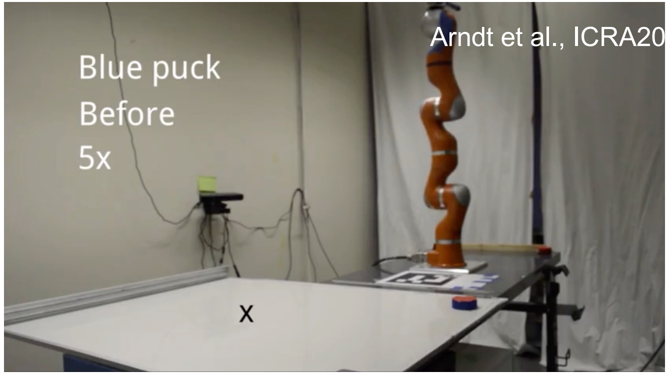
Arndt et al., ICRA20
Meta Reinforcement Learning for
Sim-to-real Domain Adaptation.

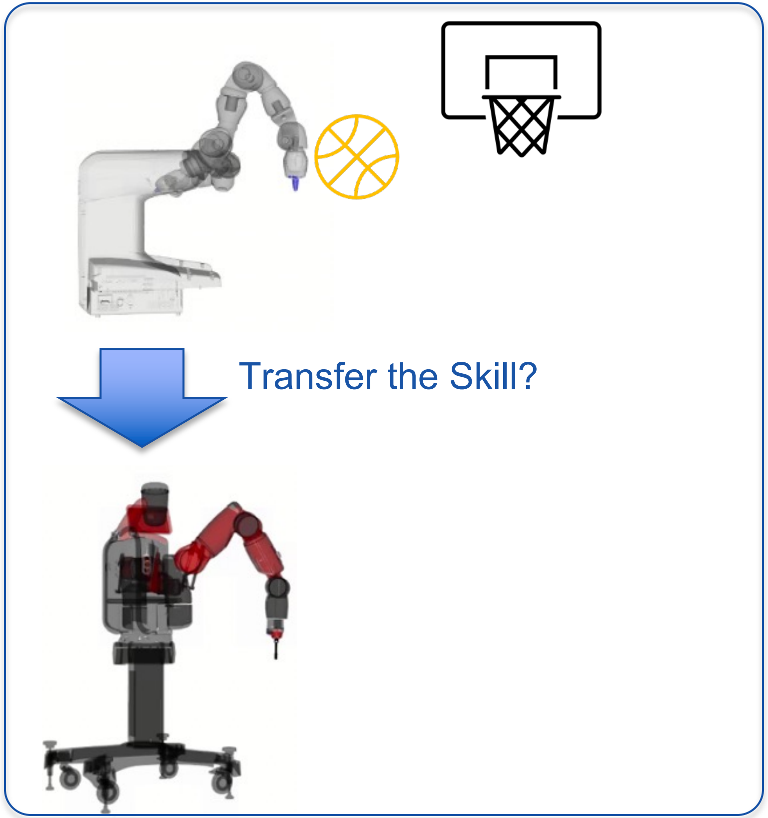
Sim-to-real Transfer Learning



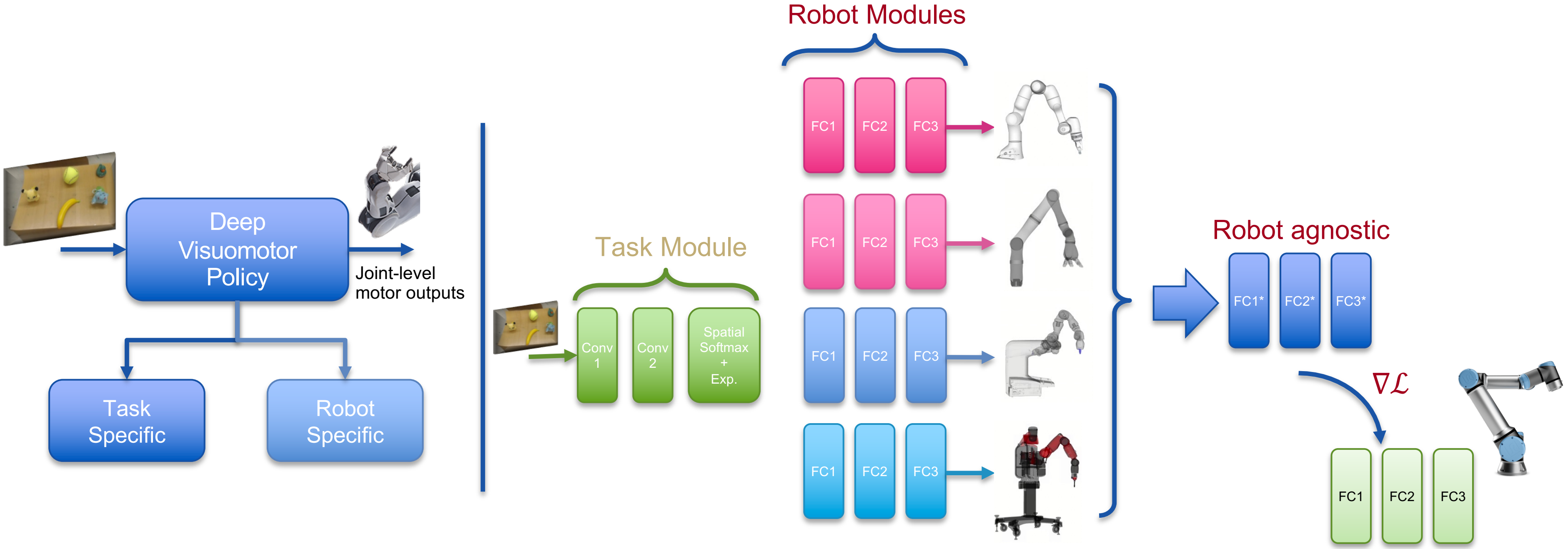
Multi-robot RL



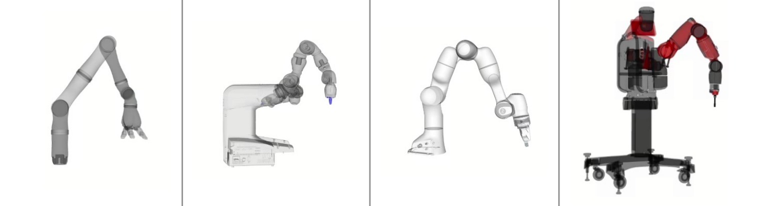




Few-shot adaptation using meta-learning



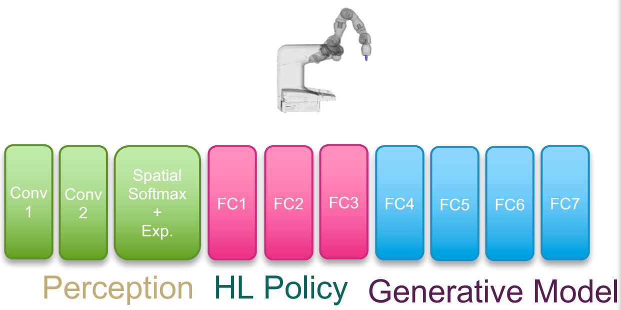
Multi-robot learning



100 rand. robots based on Kinova	100 rand. robots based on YuMi	100 rand. robots based on Franka	100 rand. robots based on Baxter

In totoal 400 different robots in simulation

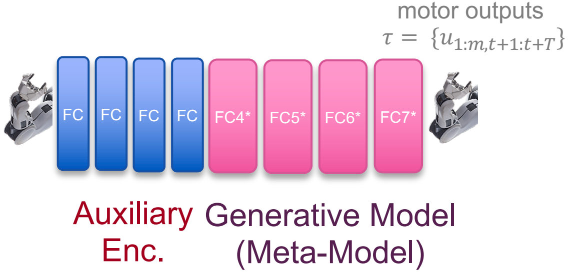
Initial Training on One Robot



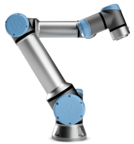
- Adversarial feature training for generalizable robotic visuomotor control, Chen et al., ICRA20
- Data-efficient visuomotor policy training using reinforcement learning and generative models, Ghadirzadeh et al., arXiv20

Meta-learning on all 400 robots

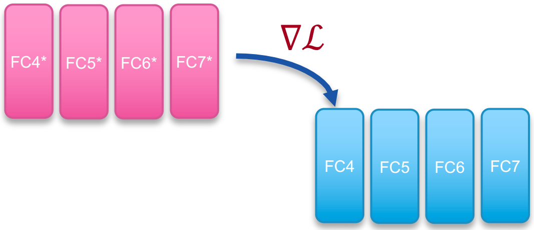
- Collect sequence of motor outputs for each robot using, e.g., RRT* planner
- Obtain initialization over generative model parameters by meta-learning



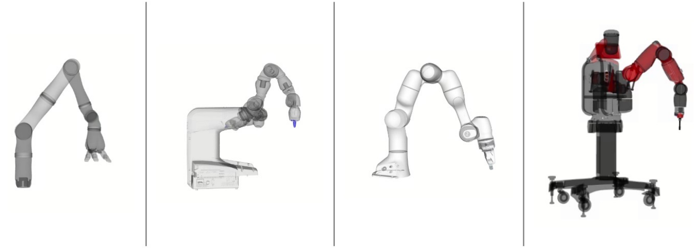
Meta-testing (adapt to novel robots)



- Provide few demonstration for the novel robot
- Adapt the generative model using few gradient descents



Model-Agnostic Meta-Learning (MAML)



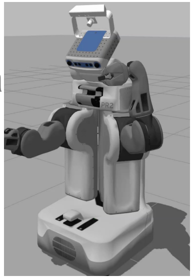
100 rand. robots based on Kinova

100 rand. robots based on YuMi

100 rand. robots based on Franka

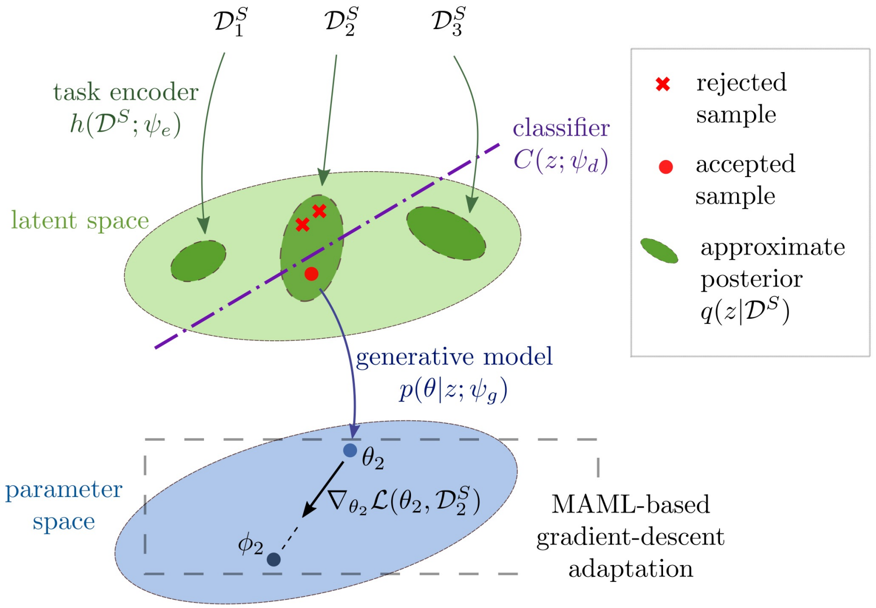
100 rand. robots based on Baxter

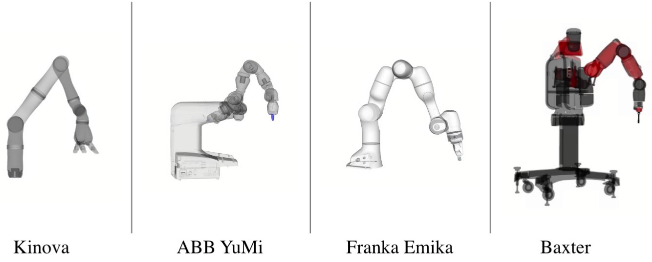
For every **robot** i in dataset
- Collect sequences of motor data
- Split the data and add to
  - Support dataset $\mathcal{D}_{\mathcal{T}_i}^{tr}$
  - Query dataset $\mathcal{D}_{\mathcal{T}_i}^{ts}$

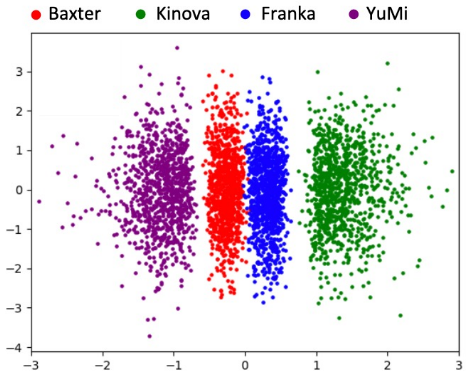


$$\min_{\phi} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}(\phi - \lambda \nabla_{\phi} \mathcal{L}(\phi, \mathcal{D}_{\mathcal{T}_i}^{\mathrm{tr}}), \mathcal{D}_{\mathcal{T}_i}^{\mathrm{ts}})$$

Conditional and Probabilistic MAML

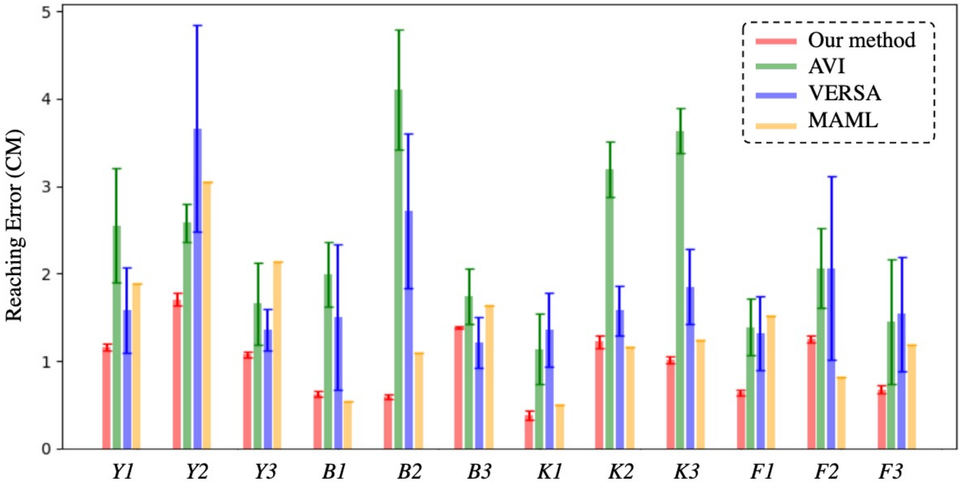






Ghadirzadeh, Poklukar, Chen, Yao, Azizpour, Björkman, Finn & Kragic
Few-shot learning with weak supervision,
ICLR workshop on Learning to Learn, 2021

Experiments 1:
meta-training and -testing on one platform (100 robots)

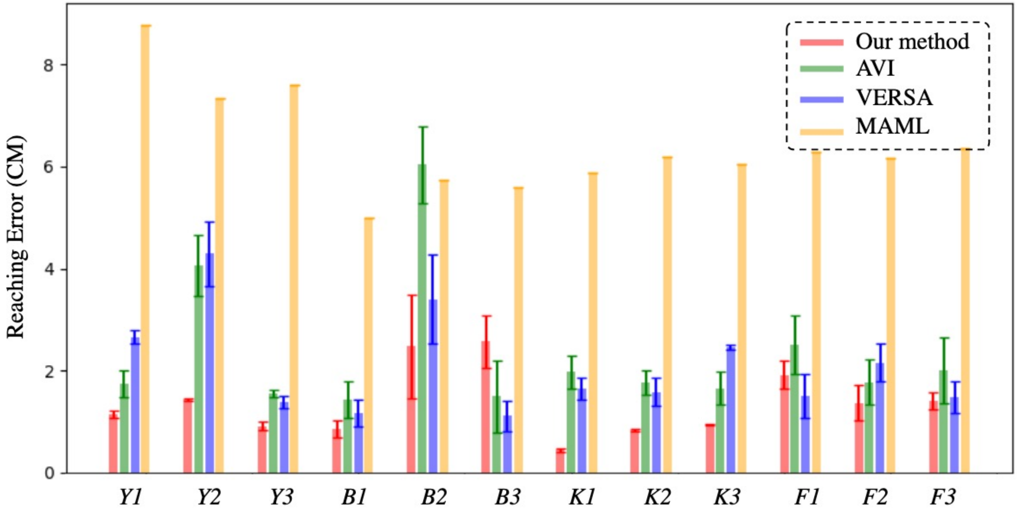


The average reaching error (in cm) of the adapted policies.
Y stands for YuMi, B for Baxter, K for Kinova, and F for Franka.

Experimental Results

Experiments 2:

meta-training and -testing on all platforms (400 robots)



The average reaching error (in cm) of the adapted policies.
Y stands for YuMi, B for Baxter, K for Kinova, and F for Franka.