Smooth Imitation Learning and Reinforcement Learning using Motion Primitives

Talk @ Imperial

Prof. Gerhard Neumann

KIT, Institut für Anthrophomatik und Robotik

Autonomous Learning Robots Lab

Karlsruher Institut für Technologie

Motivation

We will look into robot learning algorithms in terms of:

- **Smoothness**:
  - More robot friendly, more „natural“
  - Manipulate delicate objects
  - Better for exploration

- **Versatility:**
  - Learn more than one solution for a task
  - Inherent to unstructured human demonstrations
  - More robust when things change slightly

Agenda

Consider 2 types of learning…

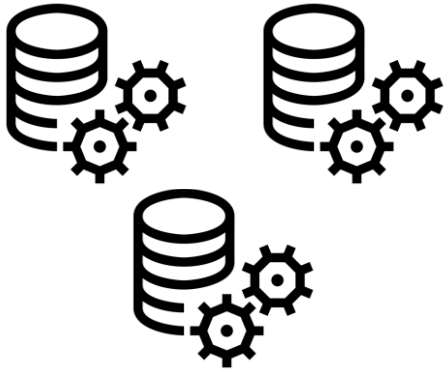
1. Imitation Learning:

- Data collection and Benchmarking using AR
- Movement Primitives and Diffusion

2. Reinforcement Learning

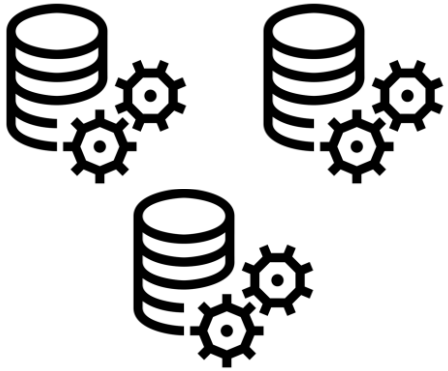
- Exploration with Motion Primitives
- Efficient Off-Policy Updates
- Discover Versatile Behavior

AR Interfaces for Virtual Data Collection

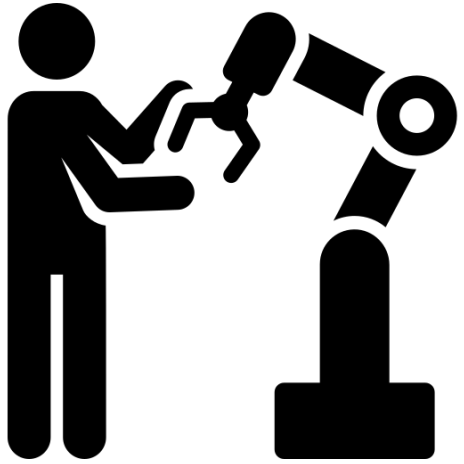


AI models need more and more data

Motivation



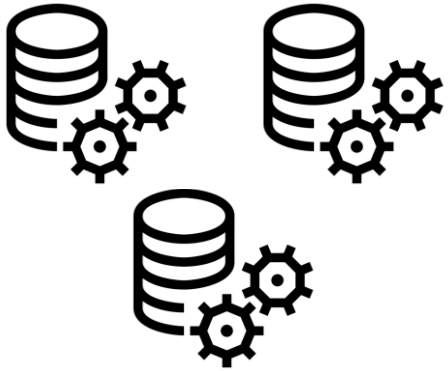
AI models need more and more data



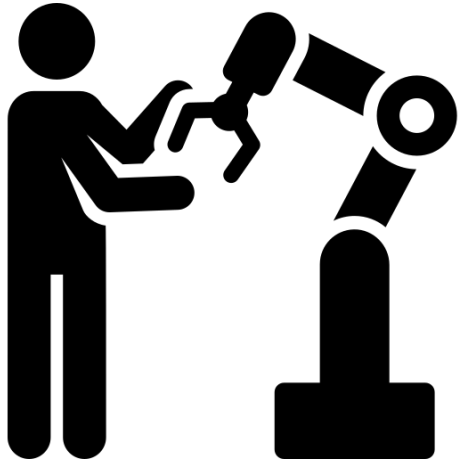
Real world experiments harbor several challenges:

- Less reproducibility
- Cumbersome and slow resetting
- Inaccurate measurement

Motivation

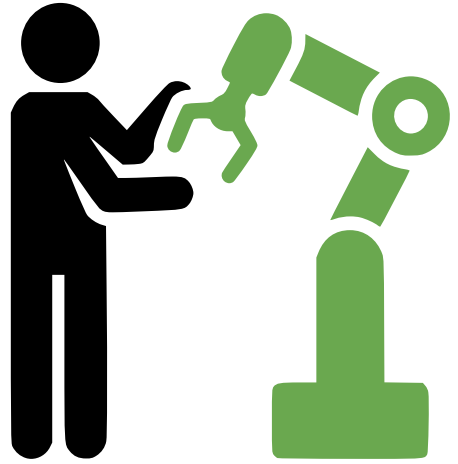


AI models need more and more data



Real world experiments harbor several challenges:

- Less reproducibility
- Cumbersome and slow resetting
- Inaccurate measurement



Virtualized experiments alleviate these challenges:

- High reproducibility
- Easy and fast resetting
- Accurate measurement

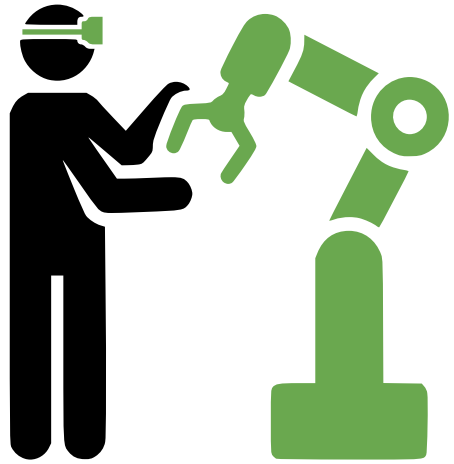
Visualization of Virtualized Experiments

Screens

- Fixed perspective
- Less interaction

vs.

Augmented Reality

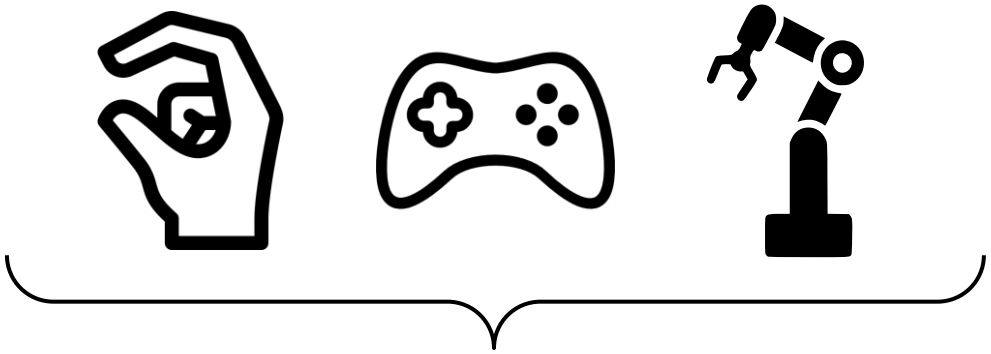


- Immersive visualization
- Interactable engagement

Controlling the virtual Robot

Recording Data by Controlling the Virtual Robot

How to control?

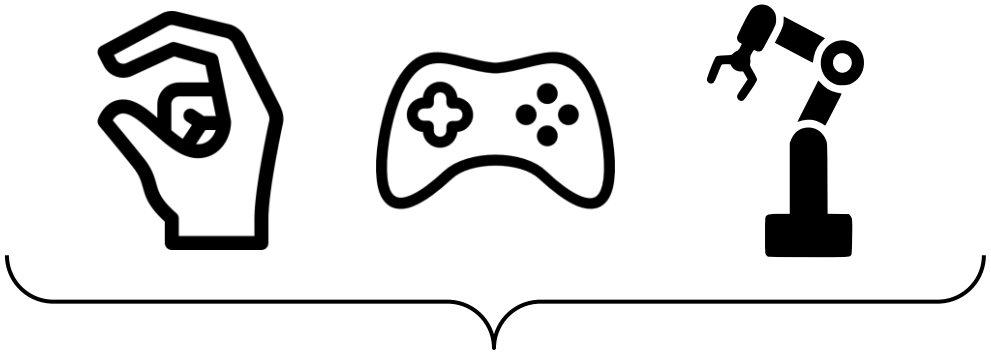


User Control Interfaces

Controlling the virtual Robot

Recording Data by Controlling the Virtual Robot

How to control?



User Control Interfaces

What is the most efficient and intuitive user interface to control the virtual robot?

User Study

User Study

5 different Control Interfaces

3 distinct Tasks

35 Participants
483 Demonstrations

3 Objective Metrics

5 Subjective Metrics

Control Interfaces - Ways to Control the Virtual Robot

Control Interfaces - Ways to Control the Virtual Robot

Hand Tracking (HT)

Least Hardware Requirements ⟷ **Most Hardware Requirements**

Autonomous Learning Robots Lab

Karlsruher Institut für Technologie

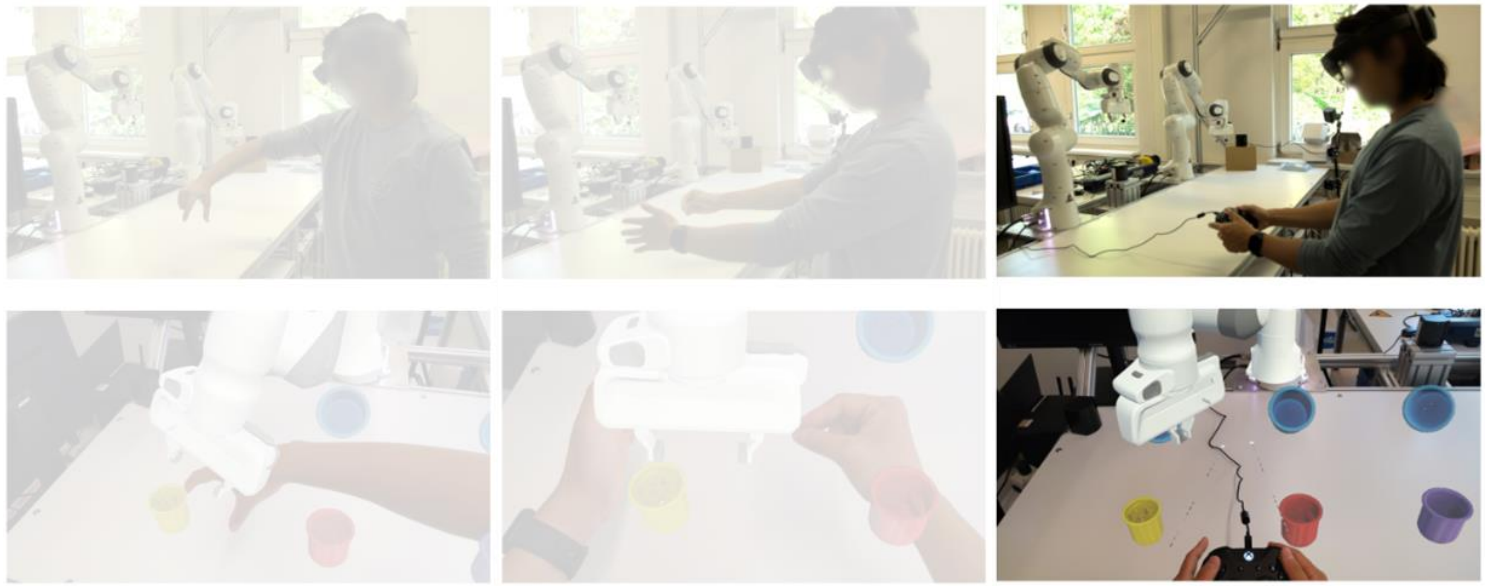
Control Interfaces - Ways to Control the Virtual Robot

Virtual Kinesthetic Teaching (VT)

Least Hardware Requirements ⟷ **Most Hardware Requirements**

ALR Autonomous Learning Robots Lab
KIT Karlsruher Institut für Technologie

Control Interfaces - Ways to Control the Virtual Robot



Gamepad (GP)

Least Hardware Requirements ⟷ **Most Hardware Requirements**

Control Interfaces - Ways to Control the Virtual Robot

Motion Controller (MC)

Least Hardware Requirements ⟷ **Most Hardware Requirements**

Control Interfaces - Ways to Control the Virtual Robot

Kinesthetic Teaching (KT)

Least Hardware Requirements ⟷ **Most Hardware Requirements**

ALR Autonomous Learning Robots Lab
KIT Karlsruher Institut für Technologie

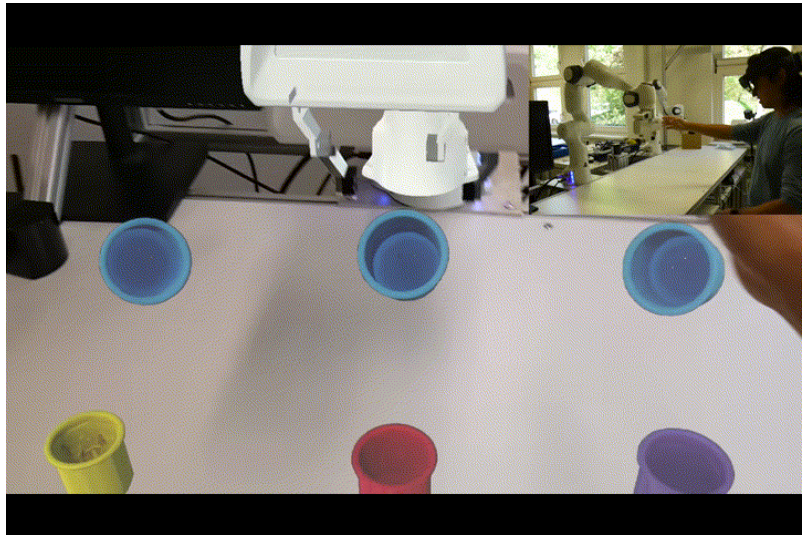
Control Interfaces - Ways to Control the Virtual Robot

Kinesthetic Teaching was never tested in the context of AR before

Kinesthetic Teaching (KT)

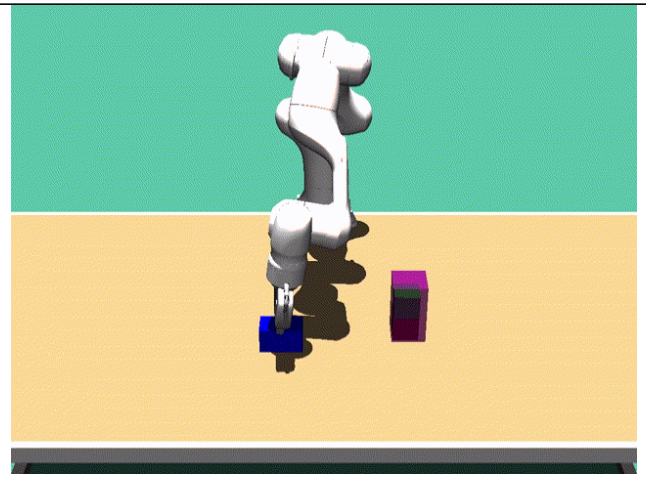
Least Hardware Requirements ⟷ **Most Hardware Requirements**

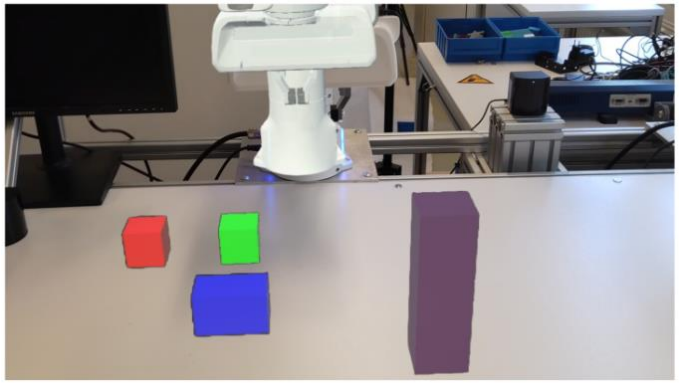
A Mixed Reality-Based Robot Learning Framework



Xinkai Jiang, Paul Mattes, Xiaogang Jia, Nicolas Schreiber, Gerhard Neumann, Rudolf Lioutikov, **A Comprehensive User Study on Augmented Reality-Based Data Collection Interfaces for Robot Learning, HRI 24**

Virtualized Tasks





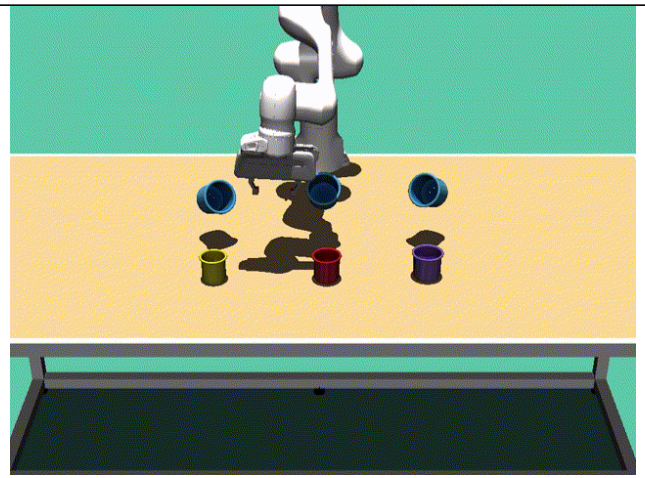
Box Stacking

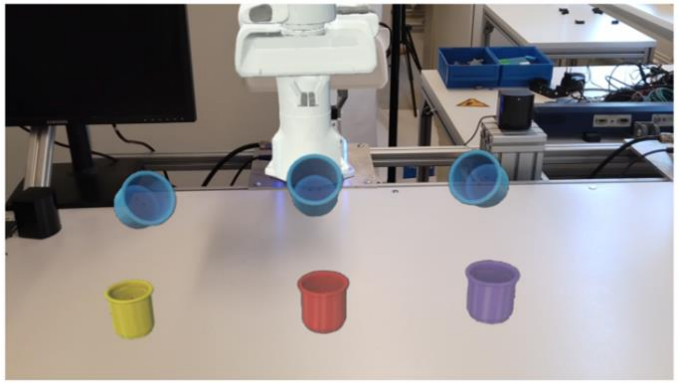
Basic Pick and Place

Autonomous Learning Robots Lab

Karlsruher Institut für Technologie

Virtualized Tasks

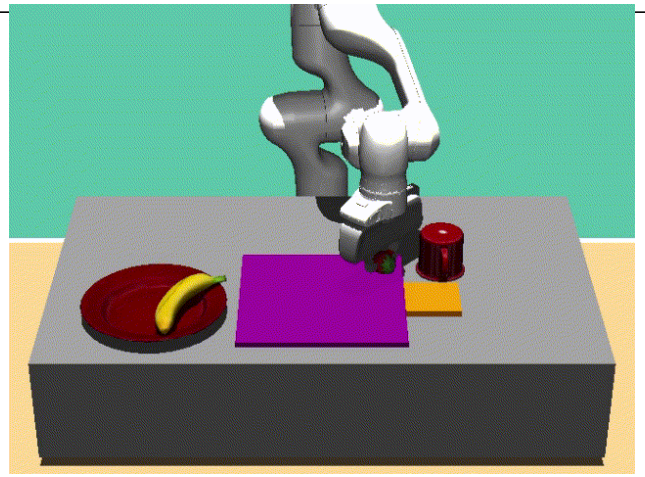


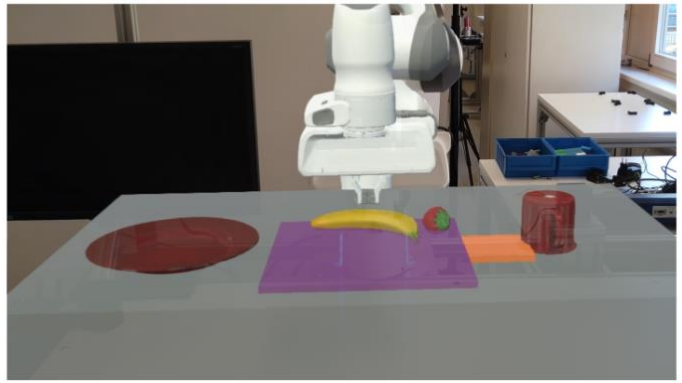


Box Stacking
Basic Pick and Place

Cup Inserting
Flexibility and Precision

Virtualized Tasks





Box Stacking
Basic Pick and Place

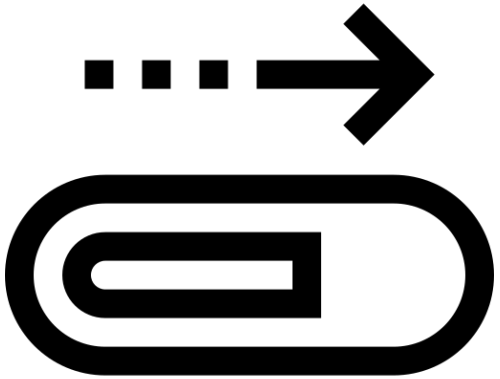
Cup Inserting
Flexibility and Precision

Practical Manipulation
Comprehensive Manipulation

Objective Metrics

Success Rate

Percentage of fully completed tasks



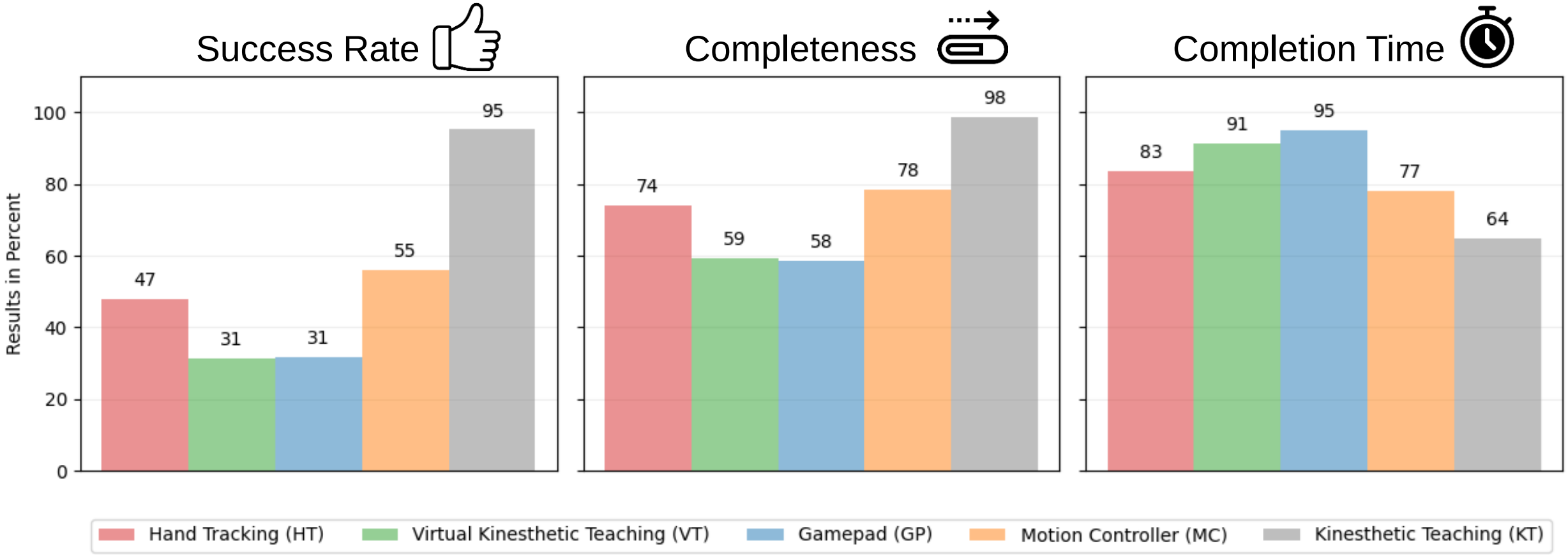
Completeness

Percentage of completed sub-goal per task

Completion Time

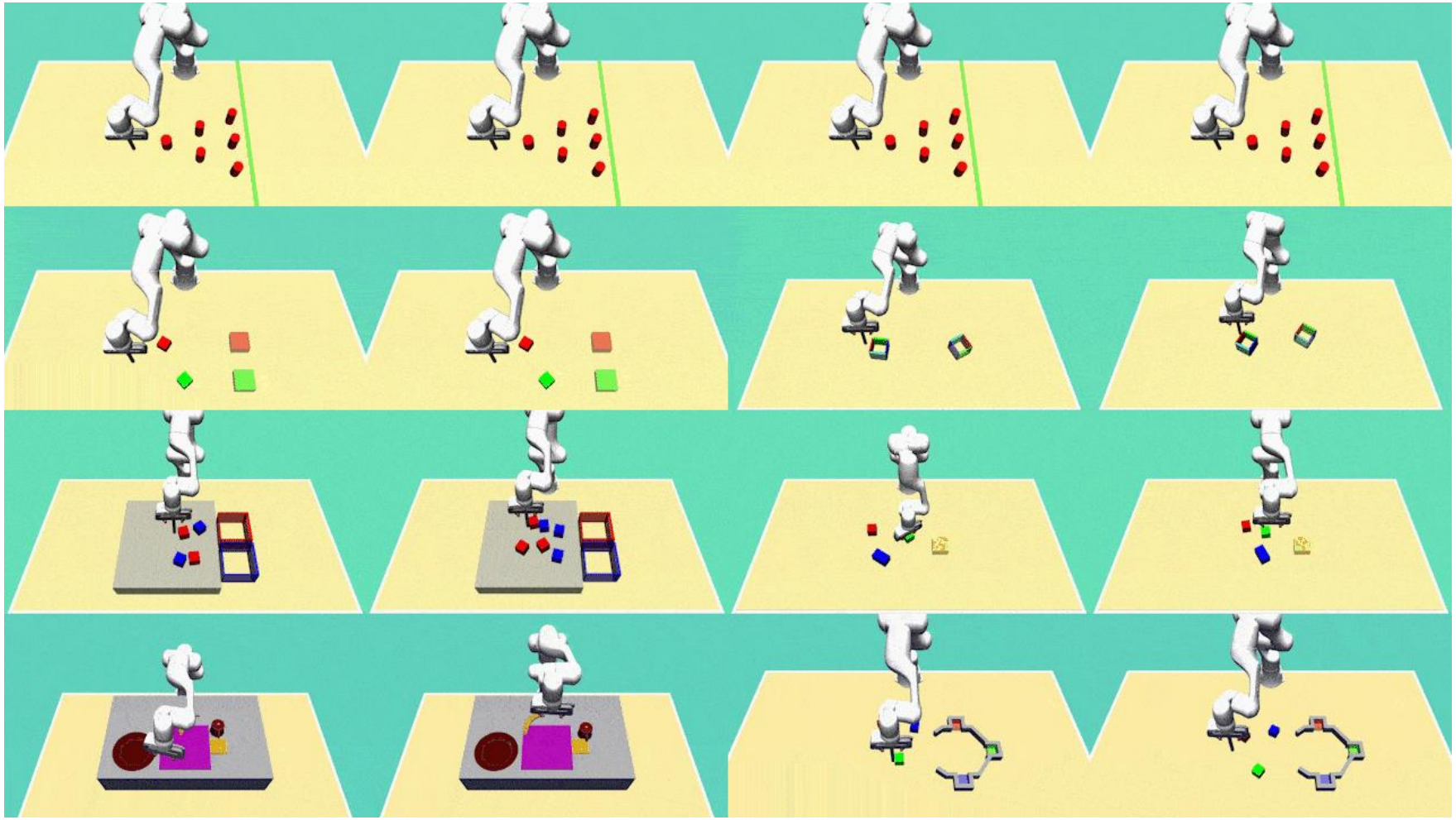
Percentage of maximum time needed per task

Objective Metrics Results



Benchmark datasets for versatile demonstrations

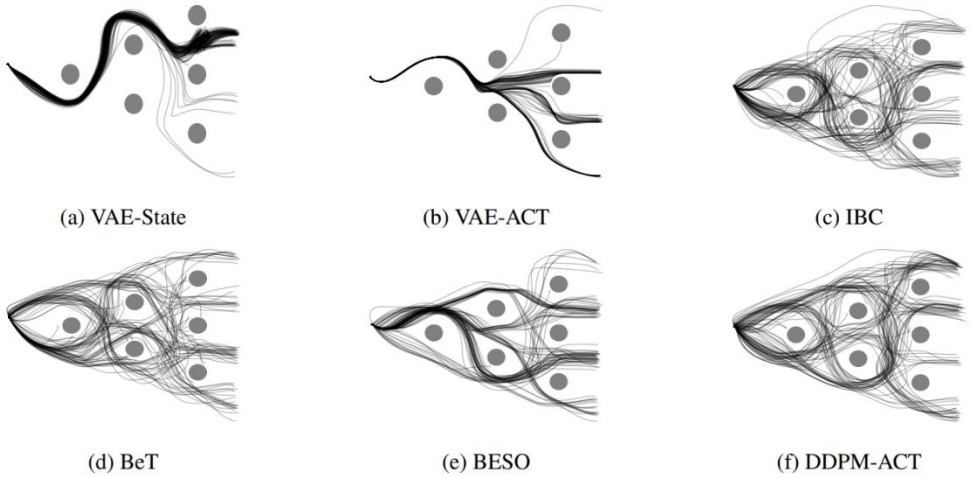
- Seven tasks (2D and 3D)
- Human demonstrations
- All show versatile behaviour
- Versatility can be quantified
- All policies can be executed
- Compare state-based policies with image-based policies

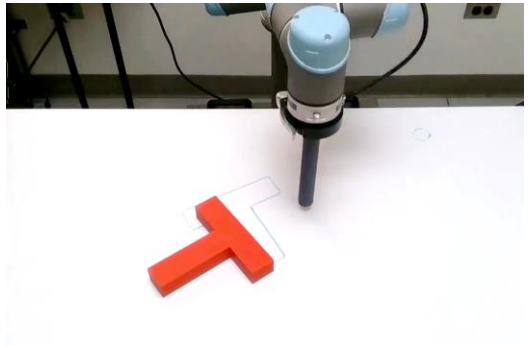


Xiaogang Jia, Denis Blessing, Xinkai Jiang, Moritz Reuss, Atalay Donat, Rudolf Lioutikov, Gerhard Neumann, Towards Diverse Behaviors: A Benchmark for Imitation Learning with Human Demonstrations, ICLR 24

Benchmarking Imitation Learning Algorithms

Can we capture the versatility in human demonstrations?

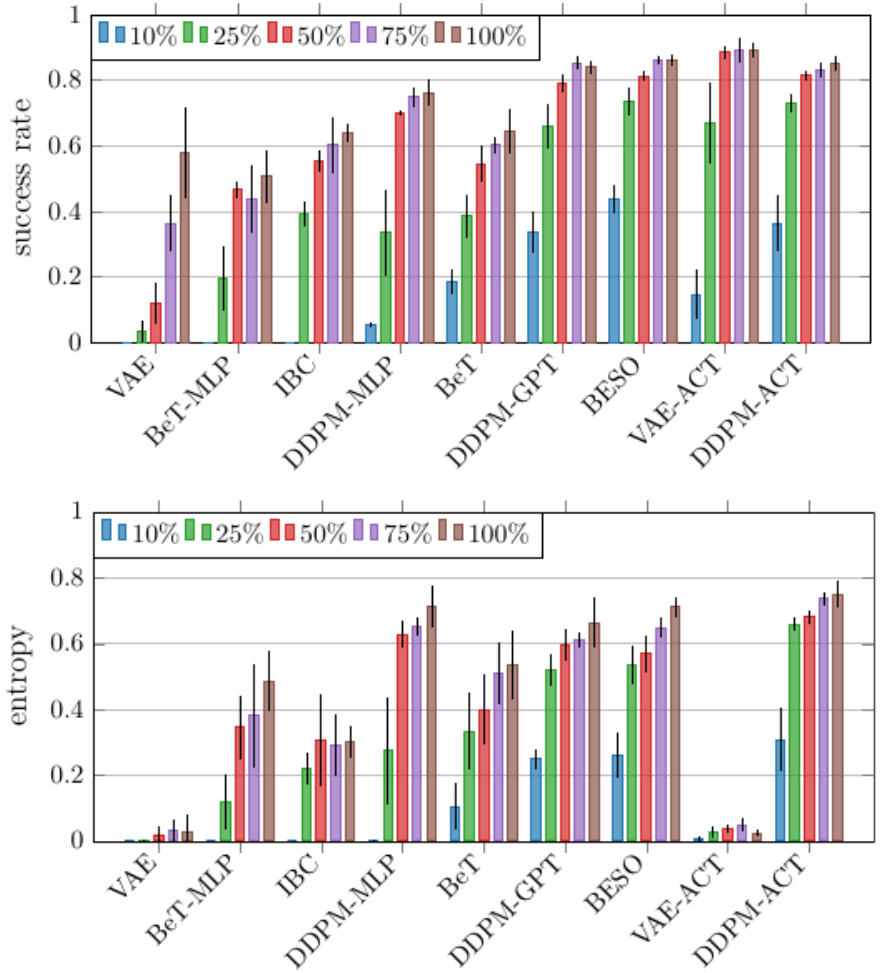




Chi, Cheng and Feng, Siyuan, ..., Song, Shuran, Diffusion Policy: Visuomotor Policy Learning via Action Diffusion, RSS 23

- Yes, most modes can be recovered (entropy)
- Algorithms that cover most modes (diffusion) are competitive,
  - But sometimes show slightly decreased performance
  - Work well with small datasets
  - Produce quite “jerky” behavior

Results on Object Alignment



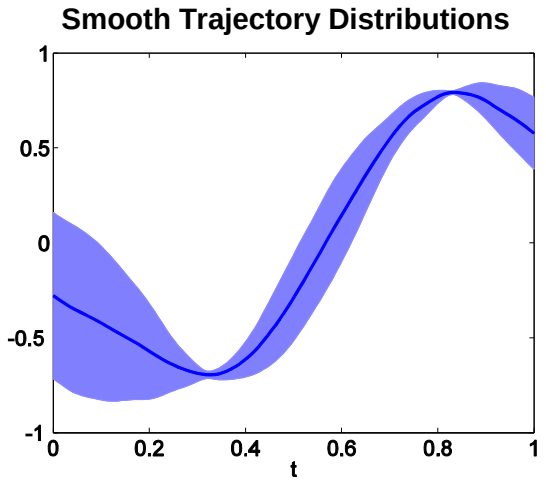
Autonomous Learning Robots Lab

Karlsruher Institut für Technologie

Movement Primitives

Movement primitives specify smooth behavior

- **Imitation learning:** Smooth execution
- **Reinforcement learning:** Smooth exploration
- Captures long-term behavior instead of just single actions



- Actions are generated by using trajectory tracking controllers

Dynamic Movement Primitives (DMPs):

$$\ddot{y} = \alpha(\beta(g - y) - \dot{y}) + \boldsymbol{\psi}_t^T \boldsymbol{w}$$

✓ Generates smooth trajectory from any initial position + velocity

× Hard to generate higher-order statistics due to integration

× Hard to integrate in neural network

Probabilistic Movement Primitives (ProMPs) :

- Directly represent trajectory via basis functions

$$y_t = \boldsymbol{\phi}_t^T \boldsymbol{w}, \quad \boldsymbol{\tau} = y_{1:T} = \boldsymbol{\Phi} \boldsymbol{w}$$

- Learn trajectory distribution

$$\boldsymbol{\tau} \sim p(\boldsymbol{\tau}) \Leftrightarrow \boldsymbol{w} \sim p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$$

✓ Captures higher order statistics: variance profile, temporal correlation,...

✓ Easy to integrate in neural networks

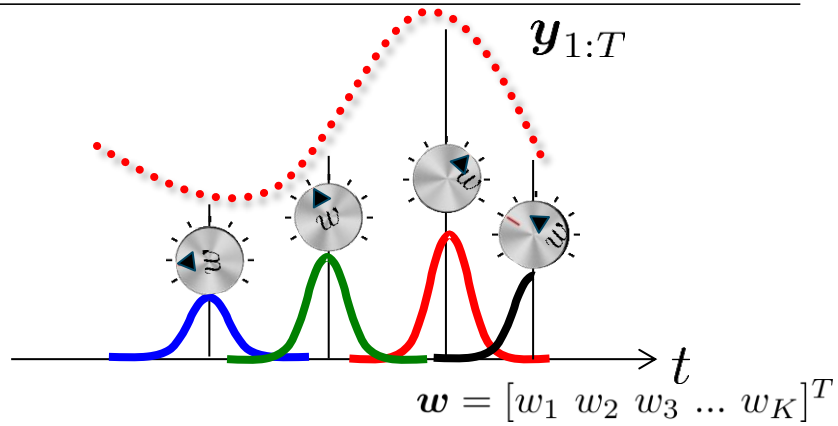
× Initial time-step or replanning causes jumps in trajectory

A. Paraschos, …, **G. Neumann,** *Probabilistic Movement Primitives*, NIPS, 2013

Probabilistic Movement Primitives

Representation of a single trajectory

$$y_t = \boldsymbol{\psi}_t^T \boldsymbol{w} + \epsilon_y \qquad \epsilon_y \sim \mathcal{N}(0, \sigma^2)$$



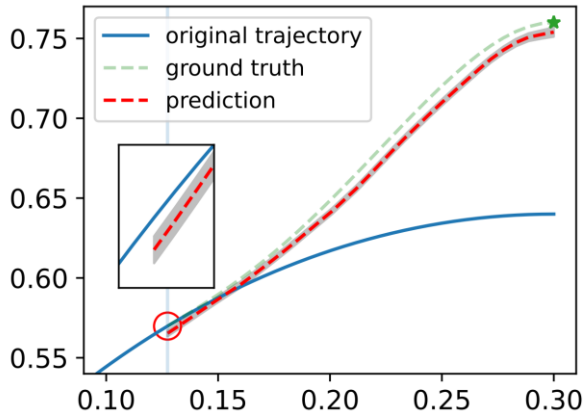
- Approximation in position instead of acceleration space

Phase-dependent basis: $\boldsymbol{\psi}_t = \boldsymbol{\psi}(z_t)$

- For example, normalized Gaussian basis functions

$$\psi_i(z) = \frac{\phi_i(z)}{\sum_{j=1}^{K} \phi_j(z)}, \qquad \phi_i(z) = \exp(-0.5(z - c_i)^2 / h_i)$$

ProMP Replaning



Autonomous Learning Robots Lab
Karlsruher Institut für Technologie

Unified perspective on DMPs and ProMPs

Probabilistic Dynamic Movement Primitives (ProDMPs):

- Solve ODE of DMPs in closed form

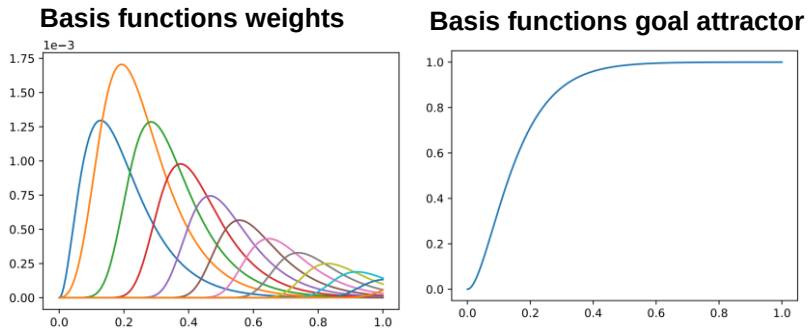
$$\ddot{y} = \alpha(\beta(g - y) - \dot{y}) + \boldsymbol{\psi}_t^T \boldsymbol{w}$$

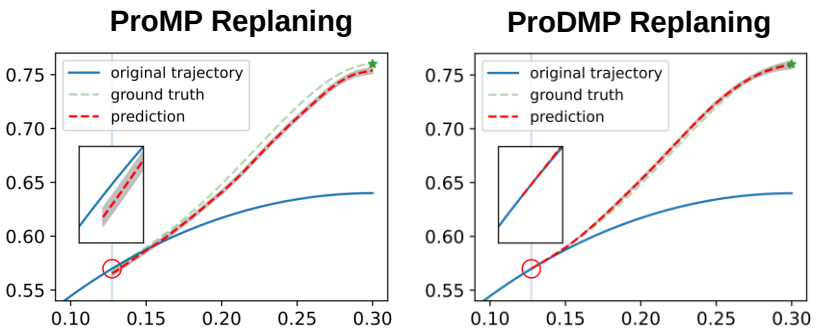
- Again, yields linear basis function representation, as in ProMPs

$$y_t = c_1 y_{1,t} + c_2 y_{2,t} + \boldsymbol{\varphi}_t^T \boldsymbol{w}_g,$$

  - Basis φ_t are obtained via numerical offline integration
  - $y_{1,t}$ and $y_{2,t}$ are homogenous solution of the ODE
  - c_1 and c_2 are constant specified by the boundary conditions (initial state)

- Boundary conditions (initial pos + vel) can always be satisfied
  - Resulting replanning trajectory is always smooth, as in DMPs



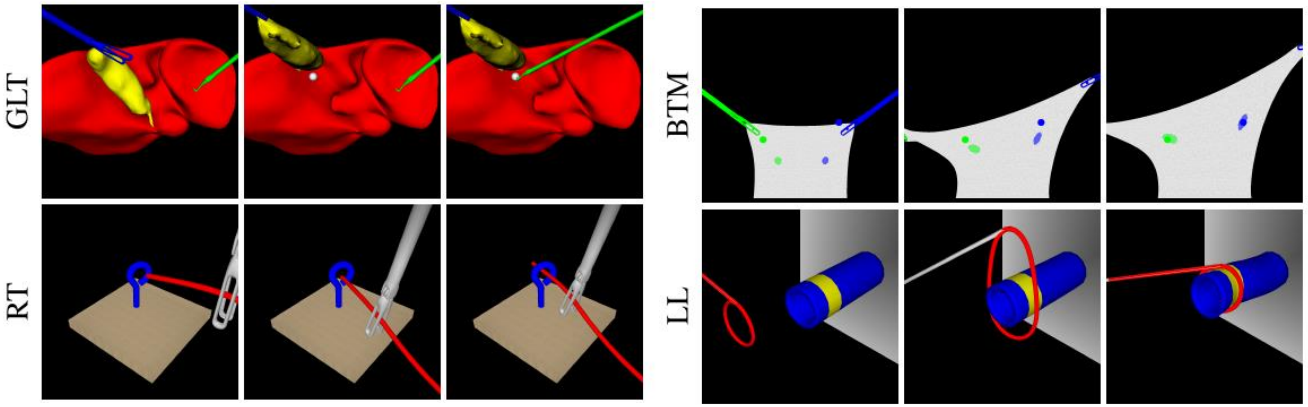


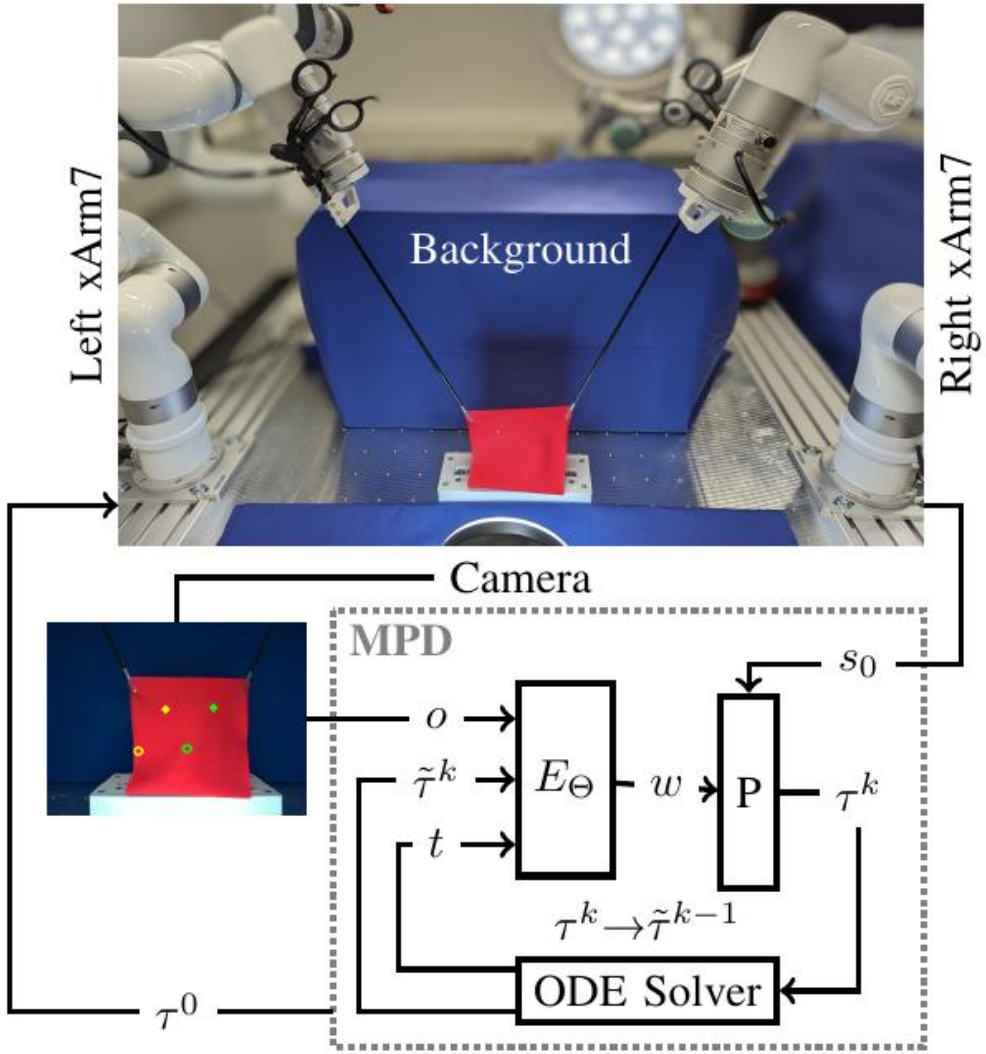
Li, G., Zeqi Q., Otto F., Lioutikov R., **Neumann, G.**; *A unified perspective on Dynamic and Probabilistic Motion Primitives, RAL 2023*

Can we learn smooth versatile behavior with MPs?

Motion Primitive Diffusion (MPD):

- Diffusion decoder predict ProDMP weights w
  - Which is translated into smooth trajectories
  - Use boundary conditions to enforce smoothness for replanning
- Evaluated on medical robotic domains
  - Smoothness is crucial for safety!

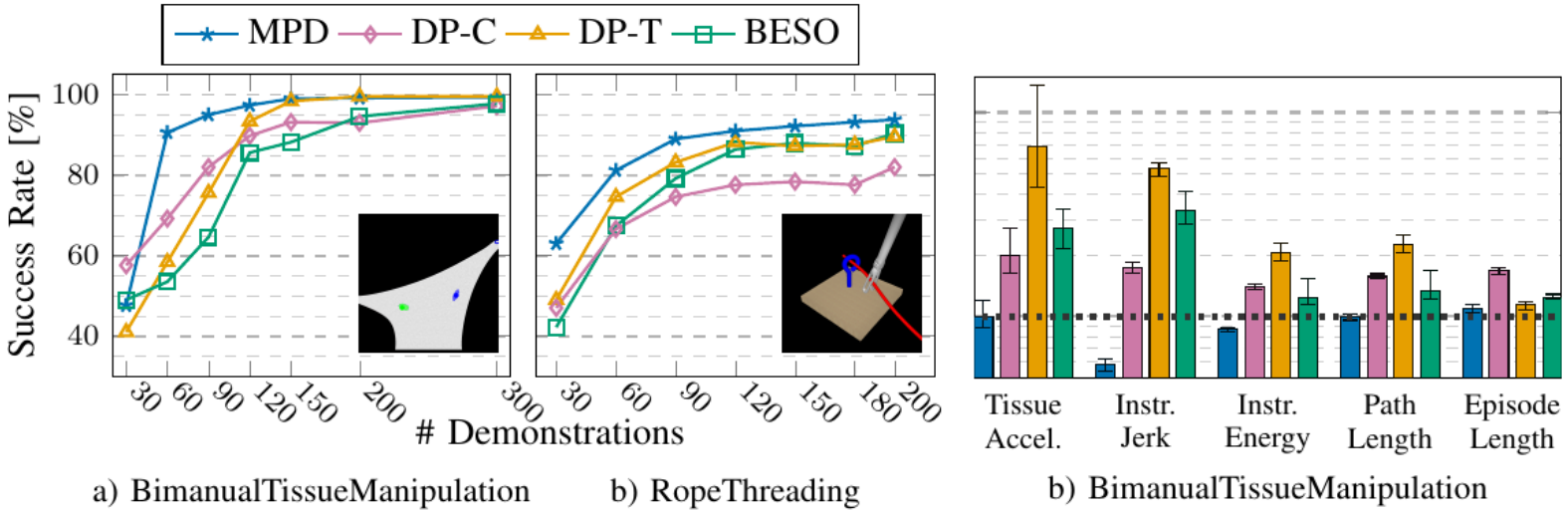




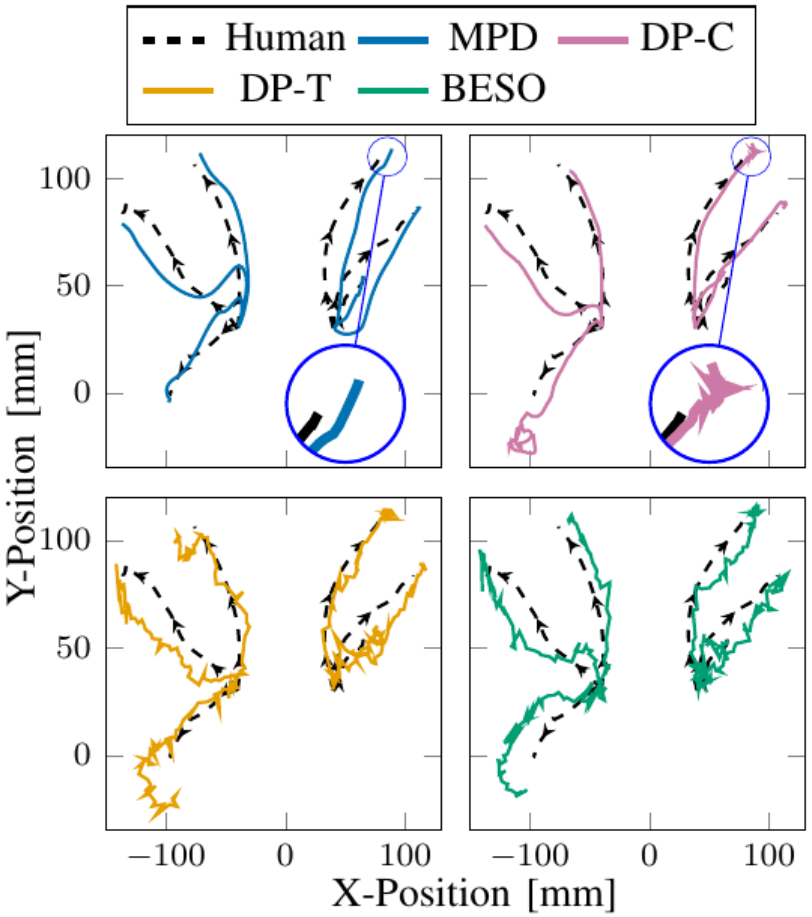
Paul Maria Scheikl, Nicolas Schreiber, Christoph Haas, Niklas Freymuth, **Gerhard Neumann**, Rudolf Lioutikov, and Franziska Mathis-Ullrich, Movement Primitive Diffusion: Learning Gentle Robotic Manipulation of Deformable Objects, RAL 2024

Motion Primitive Diffusion

Diffusion is now done on smooth MP trajectories



a) BimanualTissueManipulation b) RopeThreading b) BimanualTissueManipulation



- Outperforms Diffusion Policies in terms of success
- Always produces smooth / gentle policies
- Avoids tissue damage

Autonomous Learning Robots Lab
Karlsruher Institut für Technologie

Motion Primitive Diffusion - Results

Friedrich-Alexander-Universität Erlangen-Nürnberg

Karlsruher Institut für Technologie

Movement Primitive Diffusion: Learning Gentle Robotic Manipulation of Deformable Objects

Paul Maria Scheikl[1], Nicolas Schreiber[2], Christoph Haas[2], Niklas Freymuth[2], Gerhard Neumann[2], Rudolf Lioutikov[2], and Franziska Mathis-Ullrich[1]

1 Department Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander-University Erlangen-Nürnberg, Germany
2 Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany

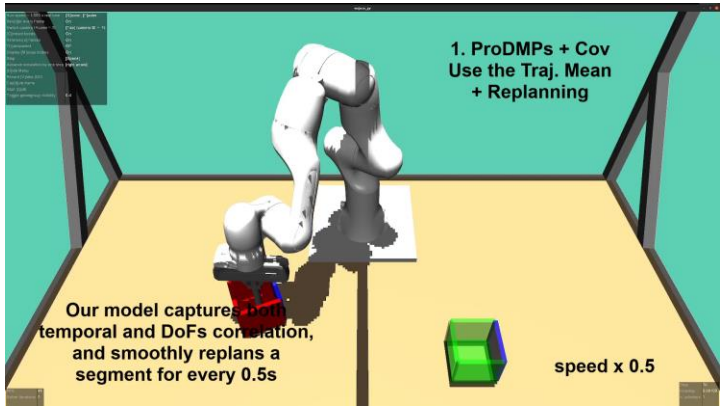
Conclusion and Future Work in Imitation Learning

Conclusion:

- AR interfaces can hugely simplify data-genaration
- Versatility can be captured well by diffusion
- Movement primitives allow for smooth behavior generation

Future work:

- AR interfaces for real bimanual robots
- Hierarchical policy representations for object centric learning



Agenda

In this talk…

1. **Imitation Learning:**
   - Data collection and Benchmarking using AR
   - Movement Primitives and Diffusion

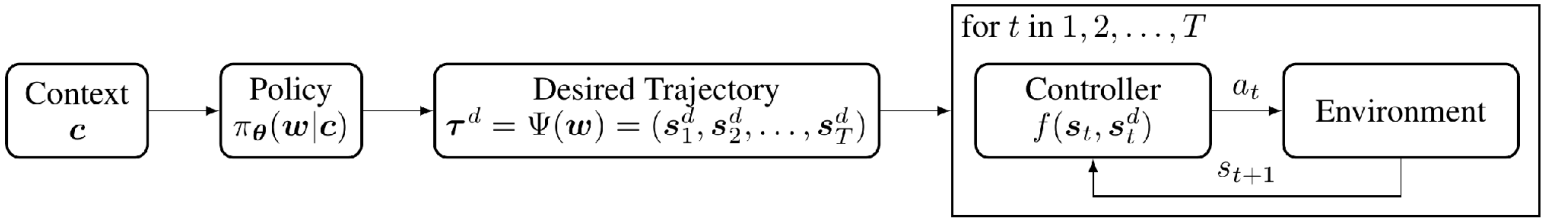
2. **Reinforcement Learning**
   - Exploration with Motion Primitives
   - Efficient Off-Policy Updates
   - Discover Versatile Behavior

Learning MP Parameters using Black-Box RL

Black-Box Reinforcement Learning with DNNs

$$\arg\max_{\pi} \mathbb{E}_{p(\boldsymbol{c})} \left[\int \pi_{\boldsymbol{\omega}}(\boldsymbol{w}|\boldsymbol{c}) g(\boldsymbol{c}, \boldsymbol{w}) d\boldsymbol{w} \right]$$

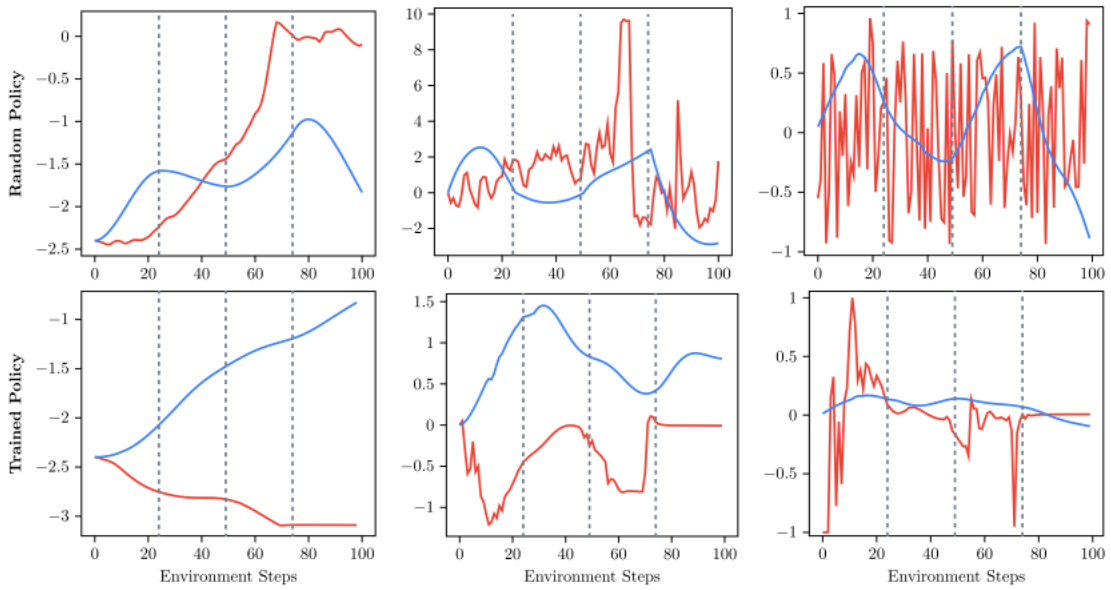
- Context c describes the task (e.g. goal location, object pose, etc…)
- Same “action” $\boldsymbol{w}$ is used for entire trajectory
- No random walk – better exploration behavior...
- ... But requires learning highly accurate action selection (no chance for correction)
- Standard deep RL algorithms (PPO, SAC) do not perform well



Autonomous Learning Robots Lab
Karlsruher Institut für Technologie

Parameter-based exploration is smooth

Red: PPO
Blue: MPs



(a) Position (rad) (b) Velocity (rad/s) (c) Torque (Nm)

Autonomous Learning Robots Lab
Karlsruher Institut für Technologie

Learning MP Parameters using Black-Box RL

Problem: Gradient descent on Gaussian distributions can get unstable, inprecise, etc...

- In particular for high dimensional action spaces (20 – 40 dimensions)

Solution: Differentiable Trust Region Layers [1]

- Onpolicy policy gradient algorithm for deep RL

Fabian Otto

Philipp Becker

[1] Fabian Otto, Philipp Becker, Vien Ngo, Hanna Ziesche, Gerhard Neuman, Differentiable Trust Region Layers for Deep Reinforcement Learning, ICLR 2021

Trust regions in policy optimization

Keep the new policy close to the old policy:

$$\boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})} \left[\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})} \hat{A}^{\pi}(\boldsymbol{s},\boldsymbol{a}) \right],$$
$$\text{s.t. } D\big(\pi(\boldsymbol{a}|\boldsymbol{s}), \pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})\big) \leq \epsilon, \quad \forall \boldsymbol{s} \in \mathcal{S}$$

- … where D is a dissimilarity measure between distributions
- I.e. we want to maximize our objective while “moving maximally epsilon away” from old policy

Benefits:

✓ Stabilizes the learning process

✓ Also avoids too fast decline of variance

× How do achieve a trust region per state?

Differentiable Trust Region Layers

If we can **construct a policy** $\tilde{\pi}_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})$ that always **satisfies the constraints**

$$D\left(\tilde{\pi}_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}), \pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})\right) \leq \epsilon, \quad \forall \boldsymbol{s} \in \mathcal{S}$$

... then the optimization reduces to:

$$\boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})} \left[\frac{\tilde{\pi}_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})} \hat{A}^{\pi_{\text{old}}}(\boldsymbol{s},\boldsymbol{a}) \right]$$

Can be implemented using **Differentiable Convex Optimization Layers** [Agrarwal2019]

$$\begin{aligned} \min \quad & f(\boldsymbol{x};\boldsymbol{\theta}) \\ \text{s.t.} \quad & h(\boldsymbol{x};\boldsymbol{\theta}) \leq 0 \\ & g(\boldsymbol{x};\boldsymbol{\theta}) = 0 \end{aligned}$$

- Convex in input ***x*** (objective and convex constraints)
- Non-convex in DNN-parameters
- Layer is fully differentiable

Differientable Trust Region Layers

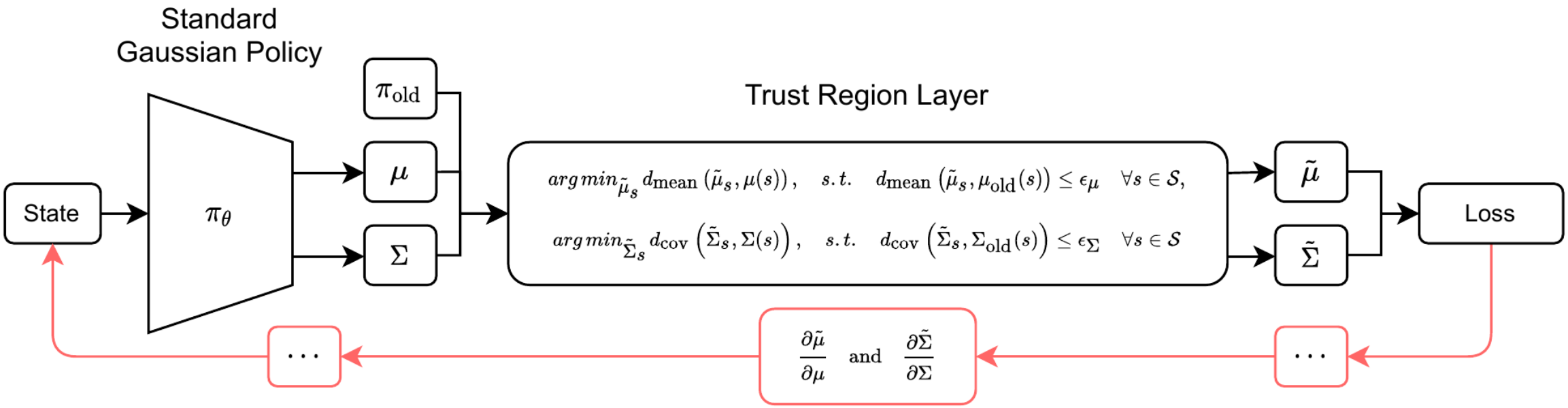
For Gaussian policies we can split the constraints into mean and variance constraint

$$D_{\text{mean}}\left(\tilde{\pi}_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}), \pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})\right) \leq \epsilon_{\mu}, \quad \forall \boldsymbol{s} \in \mathcal{S}$$

$$D_{\text{cov}}\left(\tilde{\pi}_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}), \pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})\right) \leq \epsilon_{\Sigma}, \quad \forall \boldsymbol{s} \in \mathcal{S}$$

- Mean and covariance typically require different step-sizes
- Can achieve faster convergence

Differentiable trust region layers

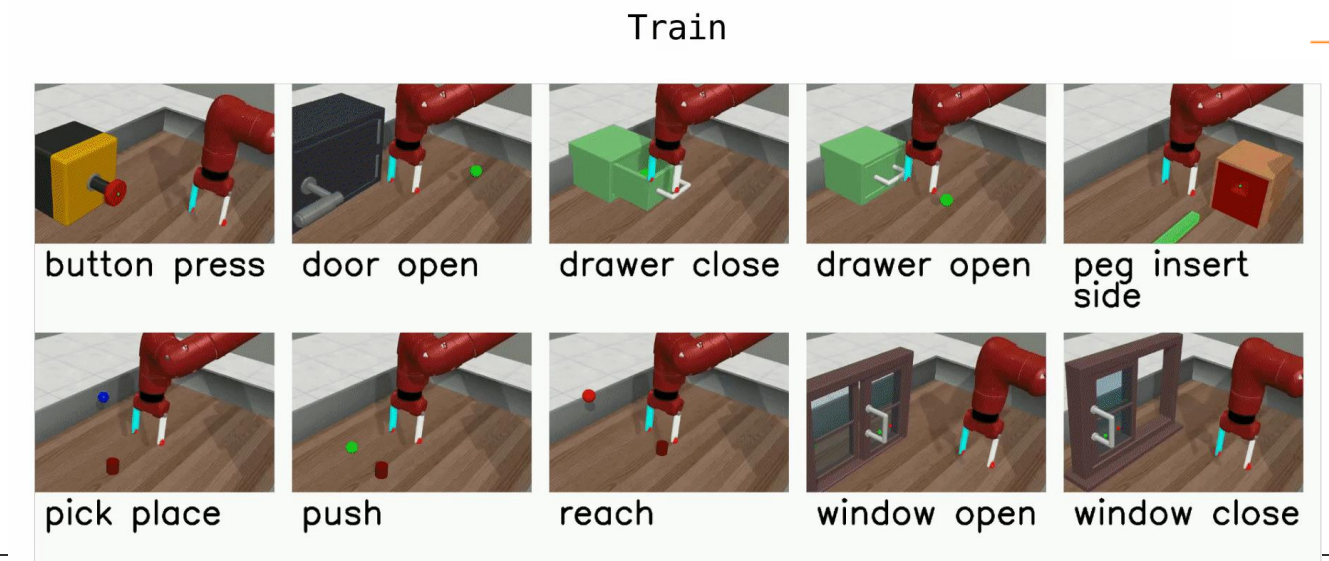


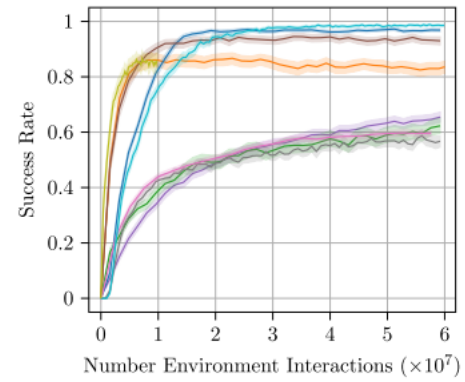
- Allows to project π_θ into the trust region by finding parameters $\tilde{\mu}$ and $\tilde{\Sigma}$ that are closest to the old mean $\mu(s)$ and covariance $\Sigma(s)$ while satisfying the trust region constraints
- Projection is done for every evaluation of the policy
- Is fully differentiable!

Experiments with Meta-World

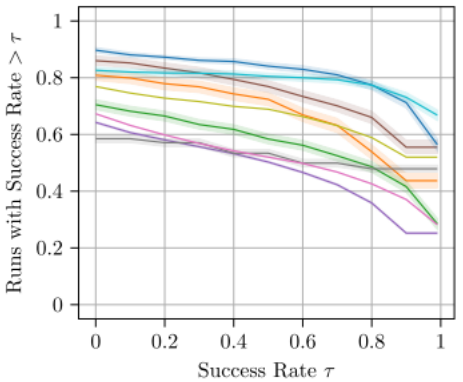
Tested on 50 Tasks from the Meta-World benchmark

- Simple to hard manipulation tasks using a simulated Sawyer robot
- TRPL outperforms PPO
- RL with MPs (MP3) achieves unpredecented performance



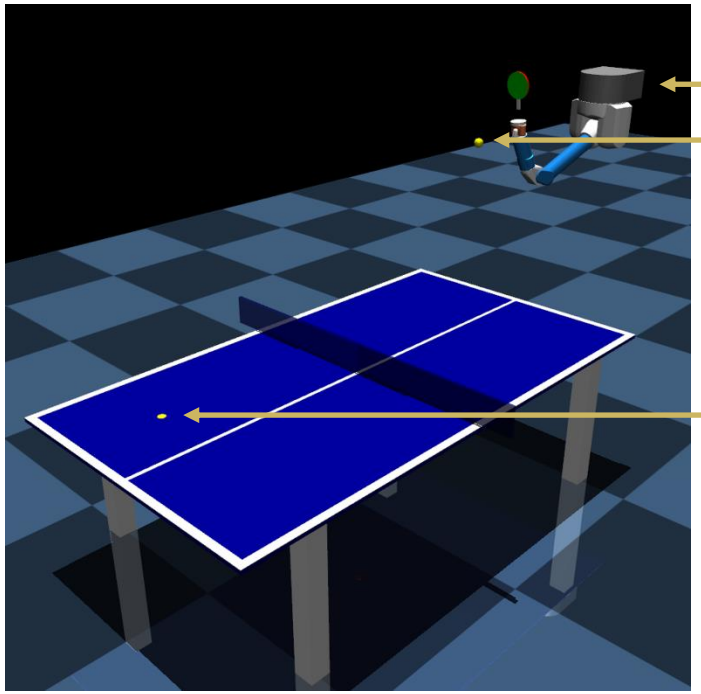


(a) MetaWorld - Sample Efficiency



(b) MetaWorld - Perf. Profile

Table Tennis



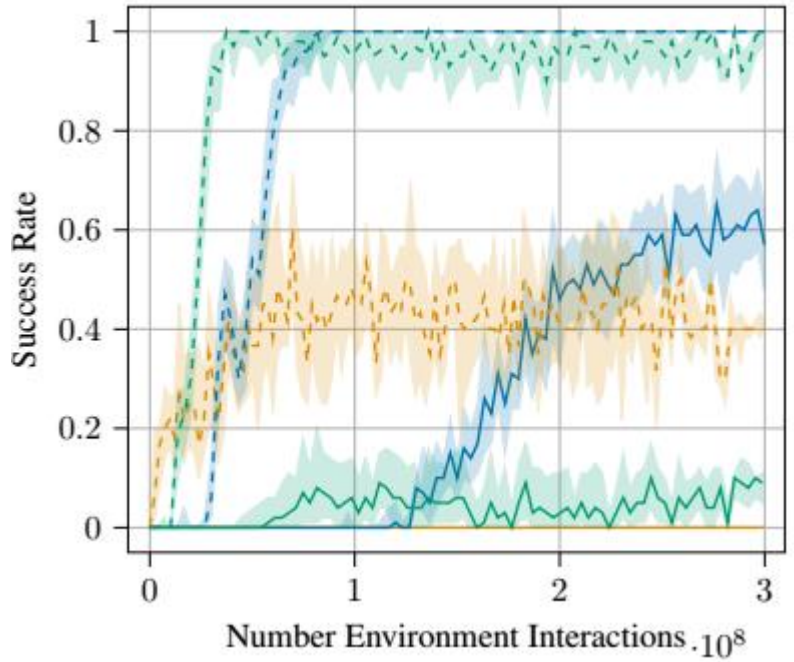
Rewards:

- **Dense:** We found no real good solution, give non-Markov reward in the end of the episode
- **Non-Markov:** Minimal distance of racket to ball, Distance of impact to desired point

Table Tennis: Results

Dashed lines: hitting rate

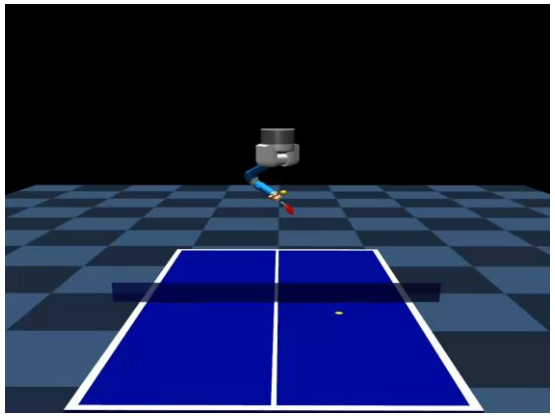
Solid lines: return to desired location rate



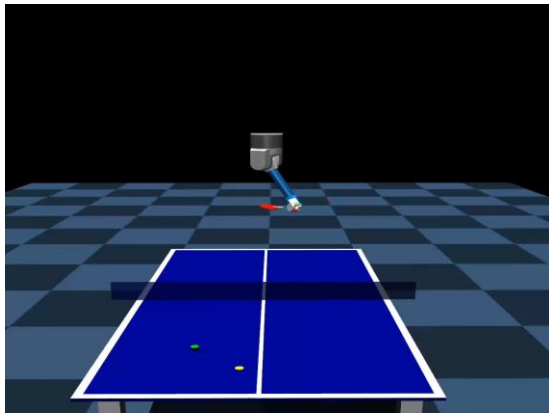
Results:

- **Step-based (orange):** Struggles with hitting the ball
- **BBRL-PPO (green):** Hits the ball, can't return
- **BBRL-TRPL (blue):** More reliable in hitting, 60% success in returning

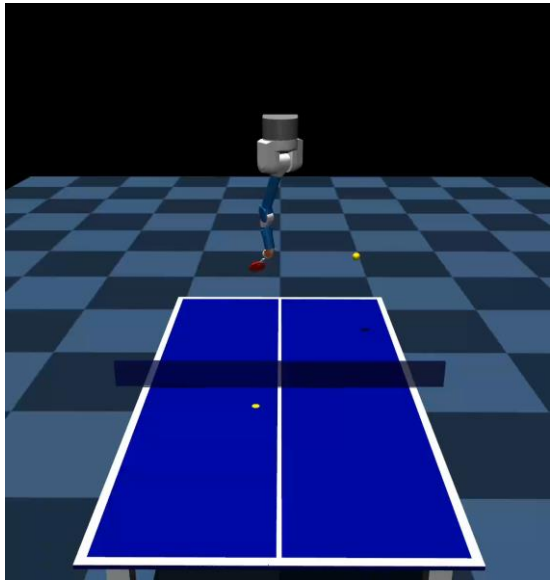
Table Tennis: Results



PPO



BBRL- PPO



BBRL-TRPL

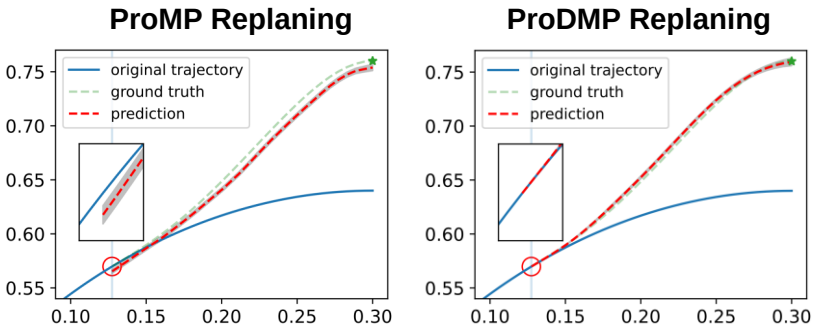
From open-loop to closed loop: Replanning

So far: Black-Box

- Trajectory generation purely open-loop (but depends on context)

But: Replanning is easy using Pro-DMPs

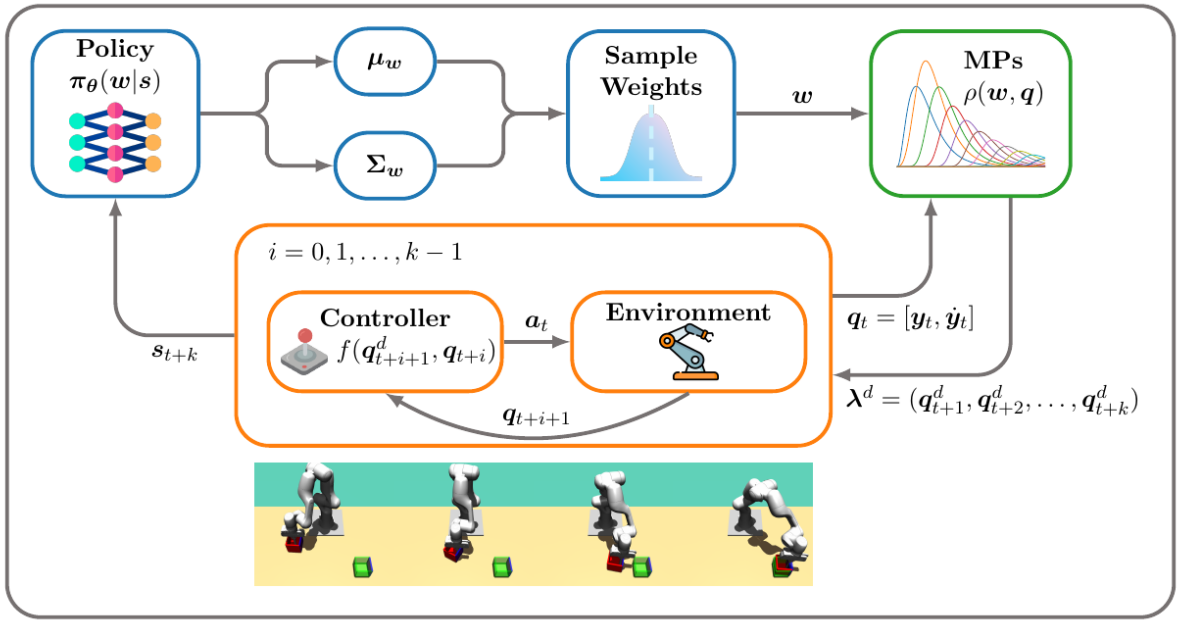
- DNN Policy can re-select parameters of MP during execution
  - still for a small number K of time steps
- Pro-DMP guarantees smoothness via boundary conditions
- 2 feedback loops on different time scales



[1] Hongyi Zhou, Fabian Otto, Onur Celik, Ge Li, Rudolf Lioutikov, Gerhard Neuman, "MP3: Movement Primitive-Based (Re-)Planning Policy", submitted to JMLR

× 1 data point per episode

✓ K data points per episode



Autonomous Learning Robots Lab

Karlsruher Institut für Technologie

Table Tennis: Wind as Dynamics

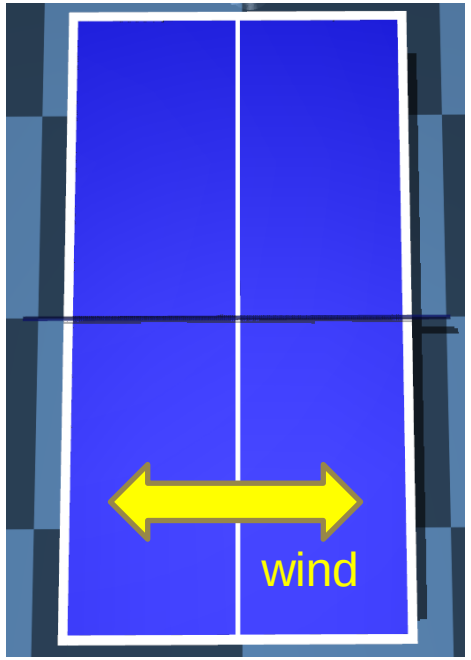
Environment Dynamics

- Wind in y-direction (not observable)
- Wind velocity [-10, 10] m/s

Replanning Times: 1 init + 3 replanning

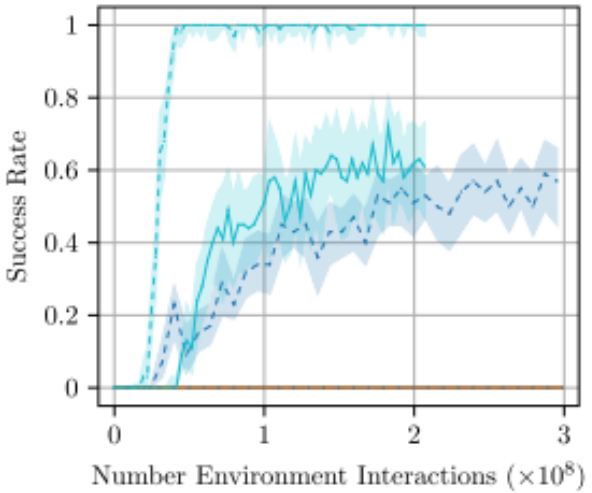
Observation Space

- Position of the ball
- Velocity of the ball
- Robot state

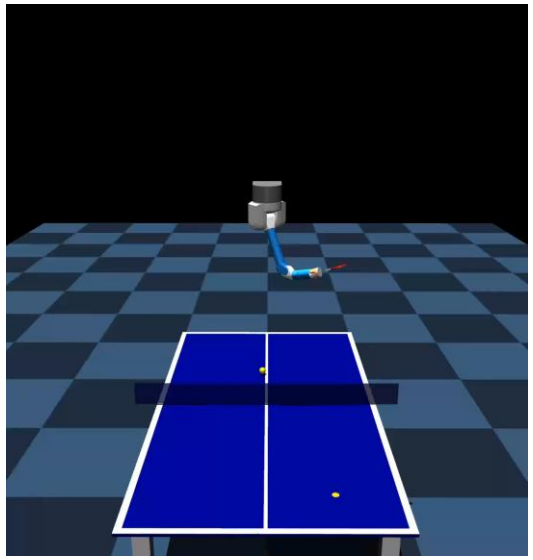


Wind direction

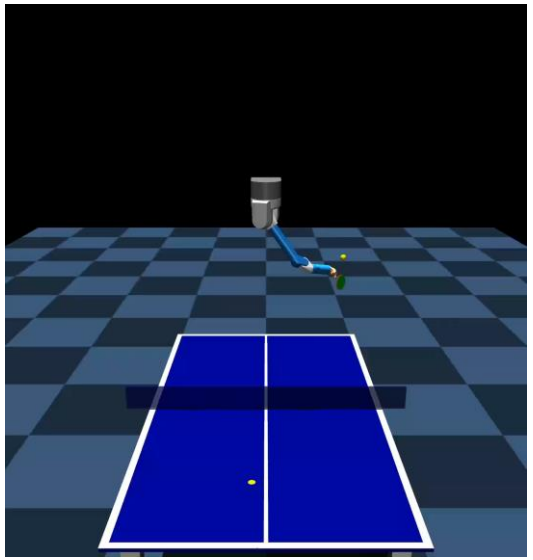
Table Tennis: Wind as Dynamics



(c) Table Tennis - Wind



BBRL

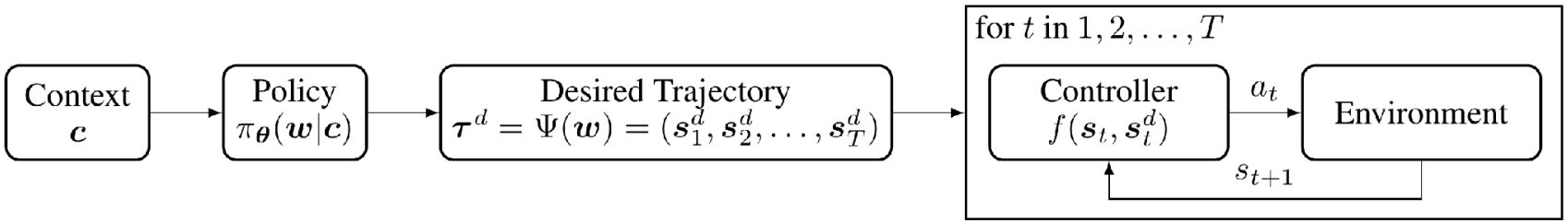


MP3 (ours)

Parameter-based exploration is sample-inefficient

Treat problem as Black Box RL:

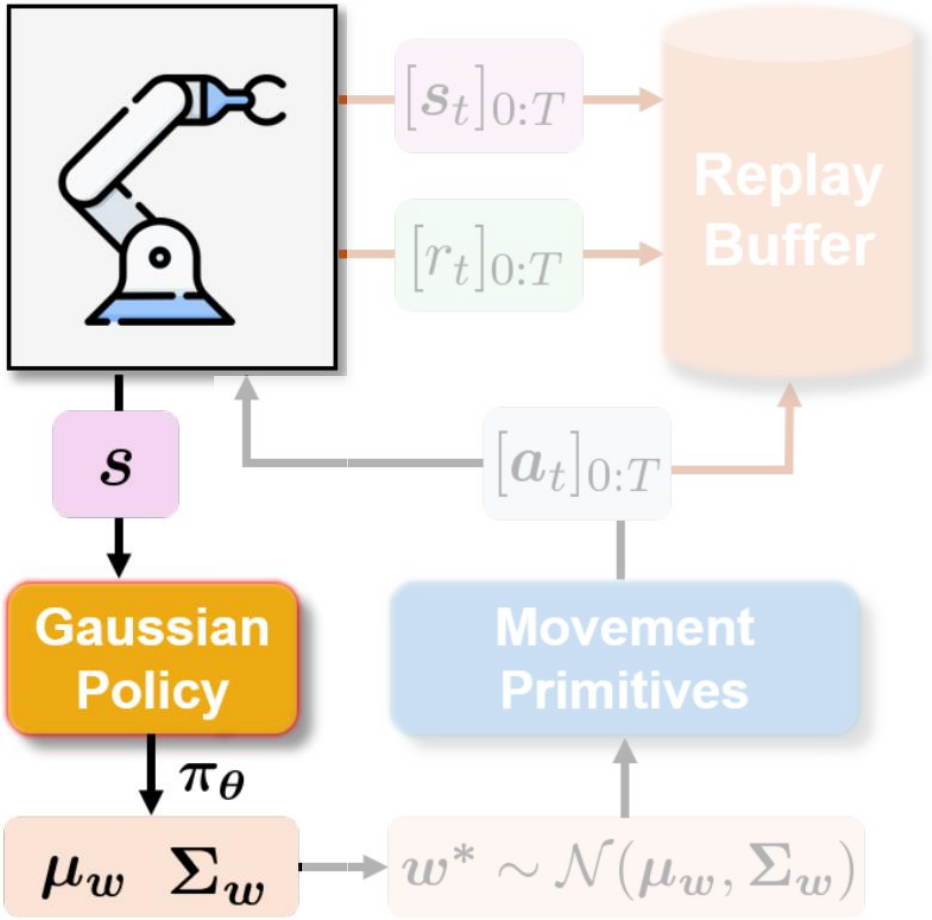
- Each trajectory is a single data point in parameter space
  - Gets a bit better with replanning, but still...
- Use black box optimization methods
- Does not exploit temporal structure of RL



Can we use parameter-based exploration with efficient off-policy updates?

Autonomous Learning Robots Lab
Karlsruher Institut für Technologie

Transformer-based Off-Policy Episodic Reinforcement Learning



Karlsruhe Institute of Technology · Autonomous Learning Robots Lab · Intuitive Robots Lab · Eberhard Karls Universität Tübingen · BOSCH

TOP-ERL: Critic Transformer

Critic Loss

For each predicted state-action value $Q(\boldsymbol{s}_0, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)$ we utilize the N-step return as its target. The objective to update the parameters ϕ of the critic is the N-step squared TD error[2]

$$\textbf{Critic loss:} \quad \mathcal{L}(\phi) = \frac{1}{L} \sum_{N=1}^{L-1} \left[\underbrace{Q_\phi(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)}_{\text{Predicted value of N actions}} - \underbrace{G^{(N)}(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)}_{\text{Target using \textbf{N-step return}}} \right]^2$$

$$+ \left[\underbrace{V_\phi(\boldsymbol{s}_0^k)}_{\text{Predicted state value}} - \mathbb{E}_{\tilde{\boldsymbol{w}} \sim \pi_\theta(\cdot|s)} [\underbrace{Q_{\phi_{\text{tar}}}(\boldsymbol{s}_0^k, \tilde{\boldsymbol{a}}_0^k, ..., \tilde{\boldsymbol{a}}_{L-1}^k)}_{\text{Target of new actions using } \tilde{\boldsymbol{w}}}] \right]^2, \qquad (7)$$

$$\textbf{N-step return:} \quad G^{(N)}(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k) = \underbrace{\sum_{i=0}^{N-1} \gamma^i r_i}_{\text{N-step rewards}} + \underbrace{\gamma^N V_{\phi_{\text{tar}}}(\boldsymbol{s}_N).}_{\text{Future return after N-step}} \qquad (8)$$

KIT Karlsruhe Institute of Technology · ALR Autonomous Learning Robots Lab · Intuitive Robots Lab · Eberhard Karls Universität Tübingen · BOSCH

Critic Loss

For each predicted state-action value $Q(\boldsymbol{s}_0, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)$ we utilize the N-step return as its target. The objective to update the parameters ϕ of the critic is the N-step squared TD error[2]

Critic loss:

$$\mathcal{L}(\phi) = \frac{1}{L}\sum_{N=1}^{L-1}\left[\underbrace{Q_\phi(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)}_{\text{Predicted value of N actions}} - \underbrace{G^{(N)}(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)}_{\text{Target using N-step return}}\right]^2$$

$$+\left[\underbrace{V_\phi(\boldsymbol{s}_0^k)}_{\text{Predicted state value}} - \mathbb{E}_{\tilde{\boldsymbol{w}}\sim\pi_\theta(\cdot|s)}[\underbrace{Q_{\phi_{\text{tar}}}(\boldsymbol{s}_0^k, \tilde{\boldsymbol{a}}_0^k, ..., \tilde{\boldsymbol{a}}_{L-1}^k)}_{\text{Target of new actions using } \tilde{\boldsymbol{w}}}]\right]^2, \quad (7)$$

N-step return:

$$G^{(N)}(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k) = \underbrace{\sum_{i=0}^{N-1}\gamma^i r_i}_{\text{N-step rewards}} + \underbrace{\gamma^N V_{\phi_{\text{tar}}}(\boldsymbol{s}_N).}_{\text{Future return after N-step}} \quad (8)$$

KIT Karlsruhe Institute of Technology · ALR Autonomous Learning Robots Lab · Intuitive Robots Lab · Eberhard Karls Universität Tübingen · BOSCH

Critic Loss

For each predicted state-action value $Q(\boldsymbol{s}_0, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)$ we utilize the N-step return as its target. The objective to update the parameters ϕ of the critic is the N-step squared TD error[2]

Critic loss:

$$\mathcal{L}(\phi) = \frac{1}{L} \sum_{N=1}^{L-1} \Bigg[\underbrace{Q_\phi(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)}_{\text{Predicted value of N actions}} - \underbrace{G^{(N)}(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k)}_{\text{Target using N-step return}} \Bigg]^2$$

$$+ \Bigg[\underbrace{V_\phi(\boldsymbol{s}_0^k)}_{\text{Predicted state value}} - \mathbb{E}_{\tilde{\boldsymbol{w}} \sim \pi_\theta(\cdot|s)} \underbrace{[Q_{\phi_{\text{tar}}}(\boldsymbol{s}_0^k, \tilde{\boldsymbol{a}}_0^k, ..., \tilde{\boldsymbol{a}}_{L-1}^k)]}_{\text{Target of new actions using } \tilde{\boldsymbol{w}}} \Bigg]^2 , \quad (7)$$

N-step return:

$$G^{(N)}(\boldsymbol{s}_0^k, \boldsymbol{a}_0^k, ..., \boldsymbol{a}_{N-1}^k) = \underbrace{\sum_{i=0}^{N-1} \gamma^i r_i}_{\text{N-step rewards}} + \underbrace{\gamma^N V_{\phi_{\text{tar}}}(\boldsymbol{s}_N).}_{\text{Future return after N-step}} \quad (8)$$

KIT Karlsruhe Institute of Technology · ALR Autonomous Learning Robots Lab · Intuitive Robots Lab · Eberhard Karls Universität Tübingen · BOSCH

Policy Learning Objective

$$J(\theta) = \mathbb{E}_{\boldsymbol{s}\sim B}\mathbb{E}_{\tilde{\boldsymbol{w}}\sim\pi_\theta(\cdot|\boldsymbol{s})}\left[\frac{1}{KL}\sum_{k=1}^{K}\sum_{N=0}^{L-1} Q_\phi(\boldsymbol{s}_0^k, \left[\tilde{\boldsymbol{a}}_t^k\right]_{t=0:N})\right]$$

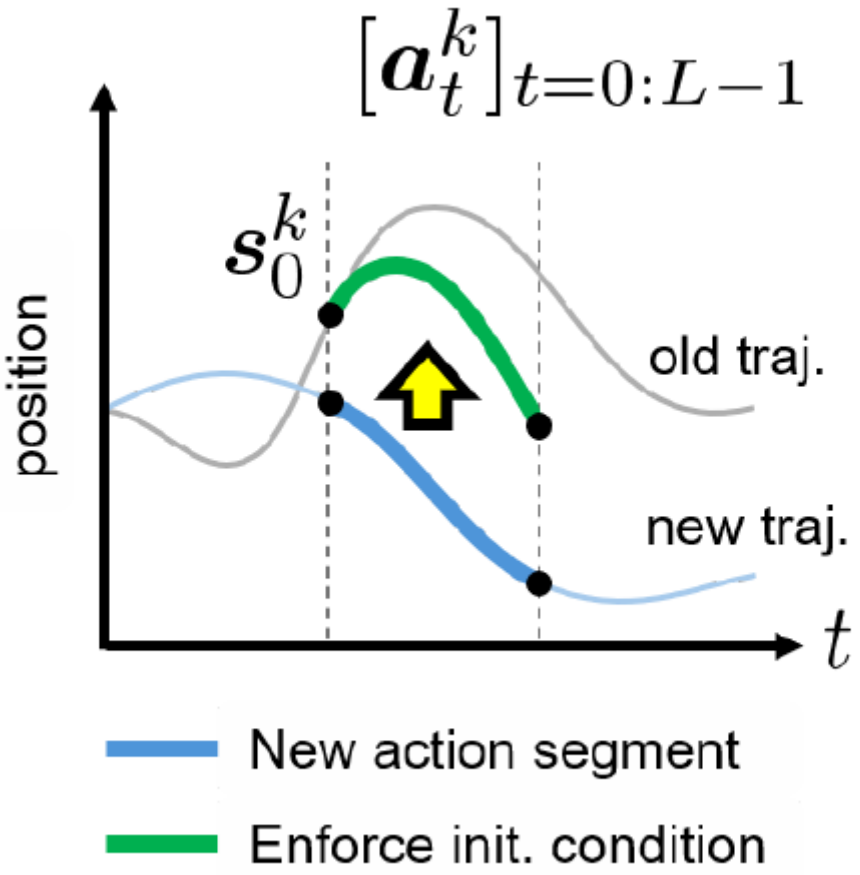
$$\left[\tilde{\boldsymbol{a}}_t^k\right]_{t=0:N} = \left[\tilde{\boldsymbol{a}}_t^k(\tilde{\boldsymbol{w}}_\theta, \boldsymbol{s}_0^k)\right]_{t=0:N}$$

$$\tilde{\boldsymbol{w}}_\theta \sim \pi_\theta(\cdot|\boldsymbol{s})$$

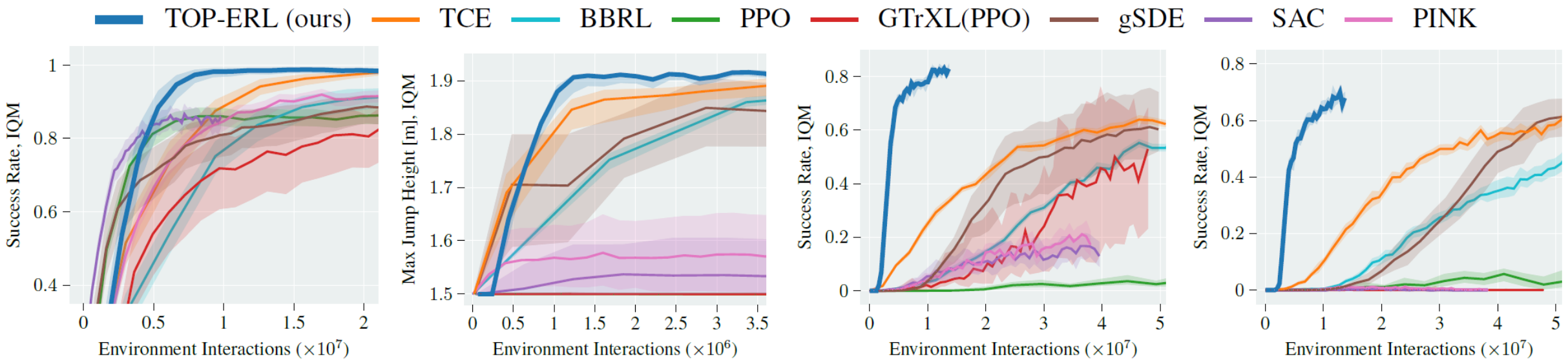
Re-param Trick, as SAC

KIT Karlsruhe Institute of Technology
ALR Autonomous Learning Robots Lab
Intuitive Robots Lab
Eberhard Karls Universität Tübingen
BOSCH

Enforce Init Condition for State and Actions



Empirical Results



(a) Metaworld Agg. of 50 tasks

(b) Hopper Jump Max Height

(c) Box Pushing Dense Reward

(d) Box Pushing Sparse Reward

Karlsruhe Institute of Technology
Autonomous Learning Robots Lab
Intuitive Robots Lab
Eberhard Karls Universität Tübingen
BOSCH

How can we make the policies more versatile?

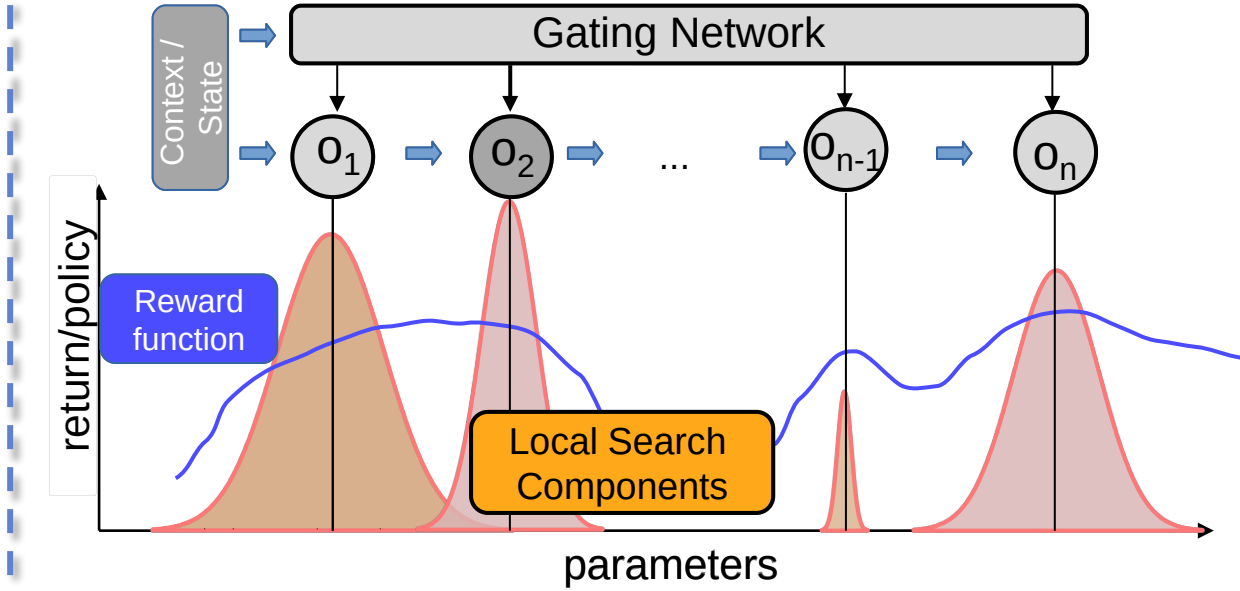
Suitable model: Mixture of Experts (MoE)

$$p(\boldsymbol{w}|\boldsymbol{c}) = \sum_o p(o|\boldsymbol{c}) p(\boldsymbol{w}|\boldsymbol{c}, o)$$

- **Gating Policy:** $p(o|\boldsymbol{c})$
  - Select component *o* given context **c**
- **Component policies**:

$$p(\boldsymbol{w}|\boldsymbol{c}, o) = \mathcal{N}\left(\boldsymbol{w}|\boldsymbol{\mu}_o(\boldsymbol{c}), \boldsymbol{\Sigma}_o\right)$$

  - Select parameters $\boldsymbol{\theta}$ given context $\boldsymbol{c}$
  - Deep NN policies per component



Powerful models, but hard to learn!

- **Components "die" – often converges to a single mode!**

Curriculum Learning for Mixture of Experts

Foster Versatility: Maximum-Entropy RL

$$\text{argmax}_{p(\boldsymbol{w}|\boldsymbol{c})} \mathbb{E}_{\mu(\boldsymbol{c})} \left[\int p(\boldsymbol{w}|\boldsymbol{c}) R(\boldsymbol{c}, \boldsymbol{w}) d\boldsymbol{w} + \alpha H\big(\pi(\boldsymbol{w}|\boldsymbol{c})\big) \right]$$

- α : scaling factor
- $H\big(\pi(\boldsymbol{\theta}|\boldsymbol{c})\big)$: Entropy of mixture model

Avoid dieing components: Curriculum Learning

$$\text{argmax}_{p(\boldsymbol{w}|\boldsymbol{c})\pi(\boldsymbol{c})} \mathbb{E}_{\pi(\boldsymbol{c})} \left[\int p(\boldsymbol{w}|\boldsymbol{c}) R(\boldsymbol{c}, \boldsymbol{\theta}) d\boldsymbol{w} + \alpha H\big(\pi(\boldsymbol{w}|\boldsymbol{c})\big) \right] - \beta \text{KL}\big(\pi(\boldsymbol{c}) || \mu(\boldsymbol{c})\big)$$

- Introduce **learned context distribution**
- Should be close to given context distribution
- **Each component will learn ist own context distribution**

Curriculum Learning for Mixture of Experts

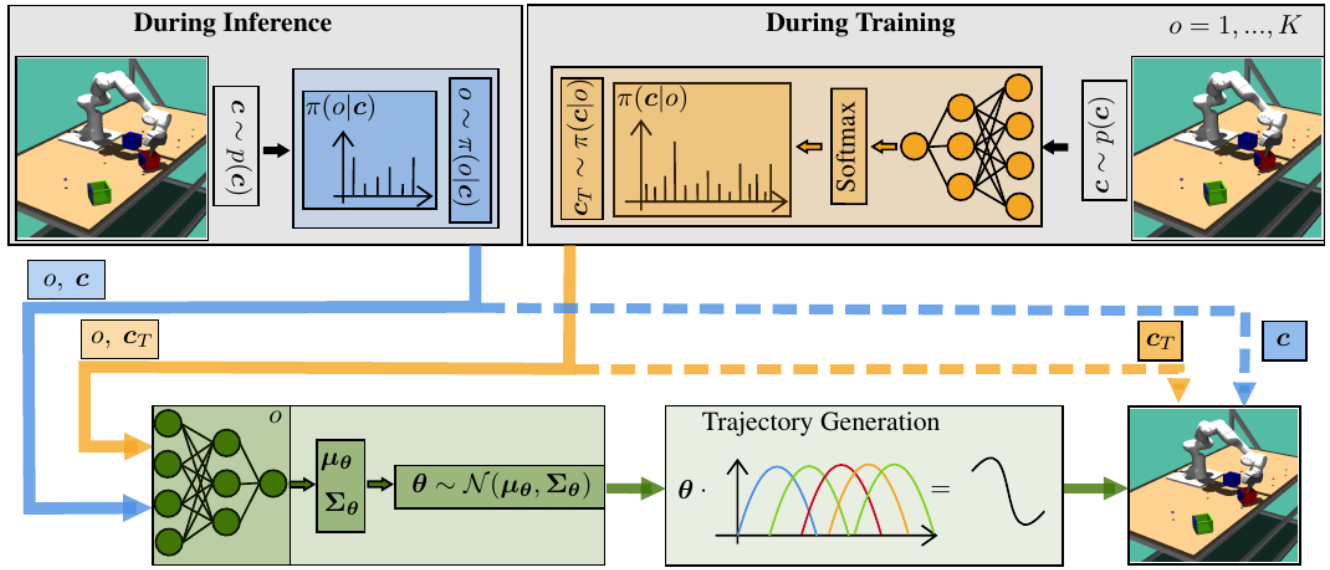
Curriculum Learning:

$$\text{argmax}_{p(\boldsymbol{w}|\boldsymbol{c})\pi(\boldsymbol{c})} \; \mathbb{E}_{\pi(\boldsymbol{c})} \left[\int p(\boldsymbol{w}|\boldsymbol{c}) R(\boldsymbol{c}, \boldsymbol{\theta}) d\boldsymbol{w} + \alpha H\big(\pi(\boldsymbol{w}|\boldsymbol{c})\big) \right] - \beta \text{KL}\big(\pi(\boldsymbol{c}) || \mu(\boldsymbol{c})\big)$$

- Introduce learned context distribution $\pi(\boldsymbol{c})$ as GMM

$$\pi(\boldsymbol{c}) = \sum_o \pi(o)\pi(\boldsymbol{c}|o)$$

  should be close to given context distribution $\mu(\boldsymbol{c})$
- Each component learns ist own curriculum / context distribution $\pi(\boldsymbol{c}|o)$
- Each component gets the same amount of samples
  - No dying of components
- Derivation of tight lower bound that allows for individual optimization of components
- Again use TRPL for component optimization

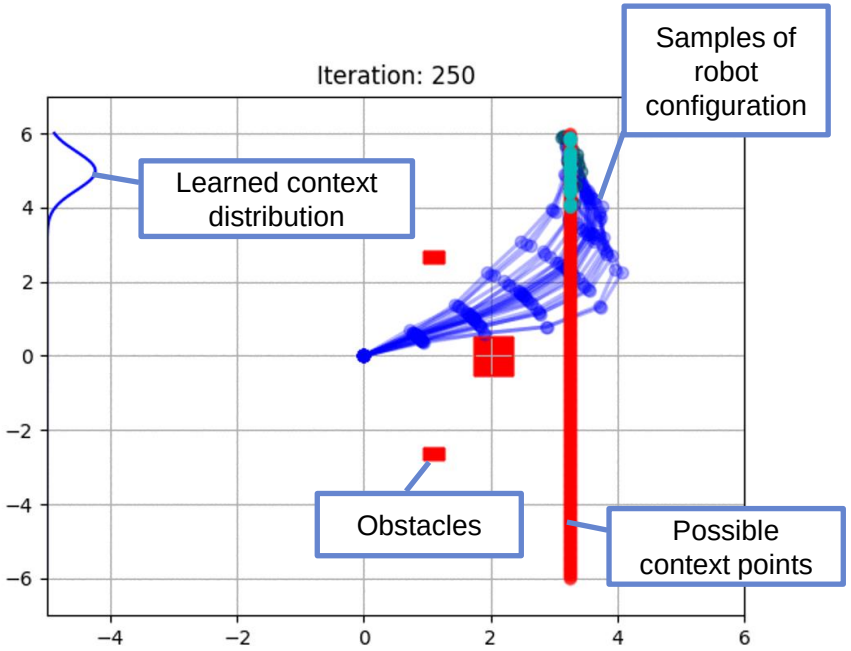


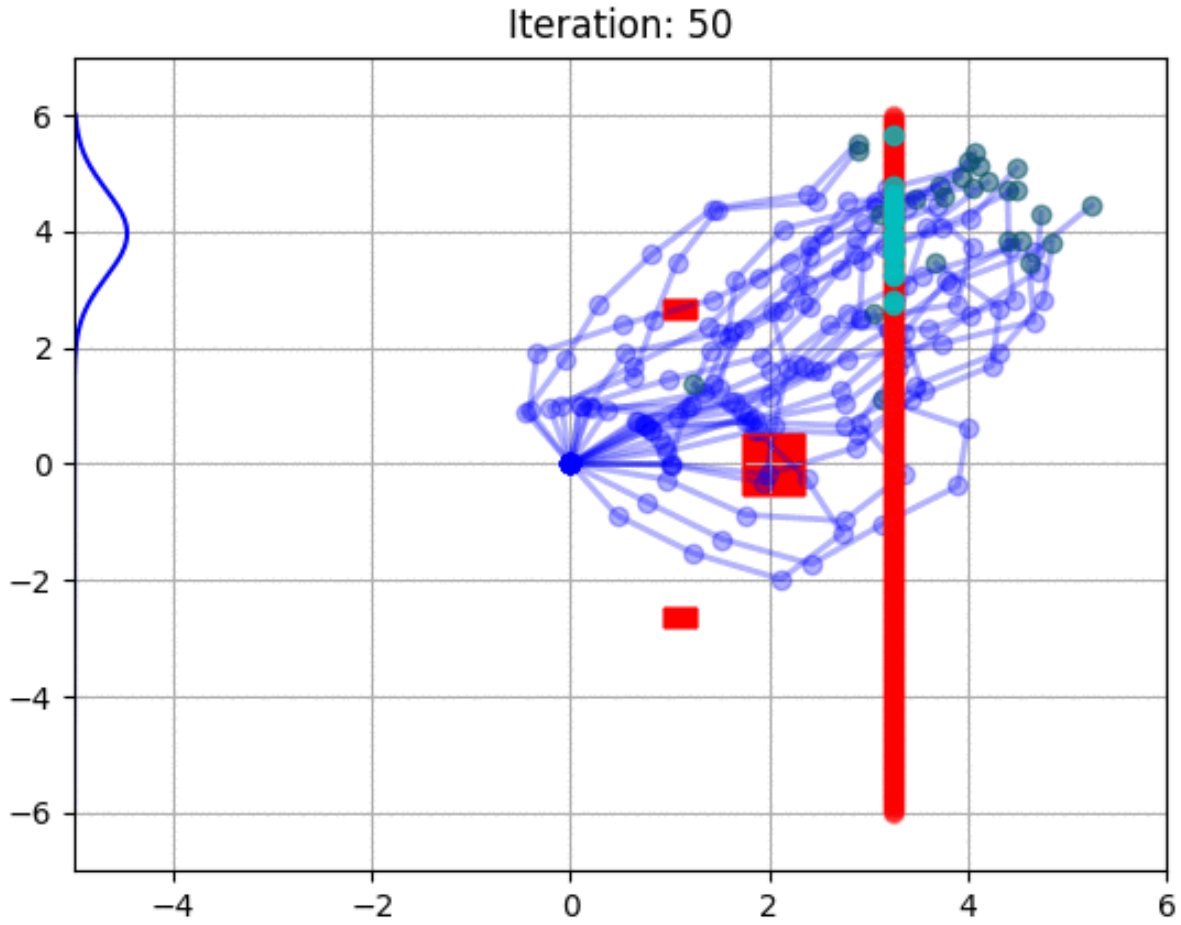
Autonomous Learning Robots Lab
Karlsruher Institut für Technologie

Local Specialization of Skills

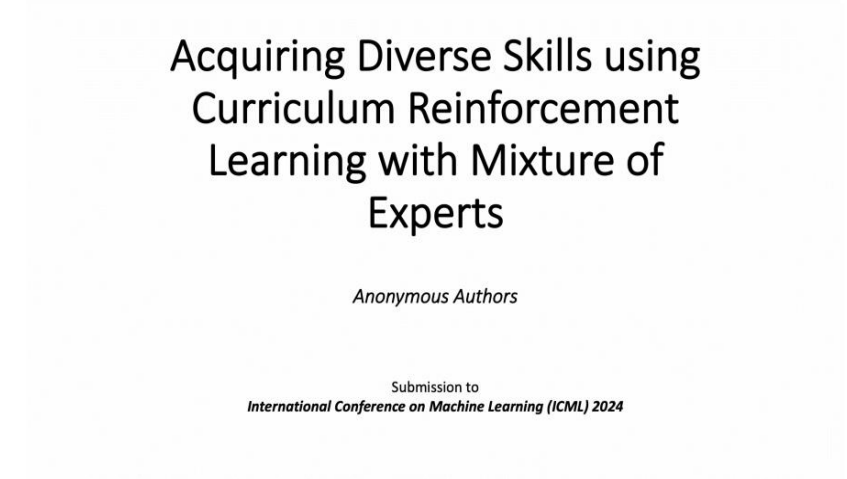
Illustration: 7-link planar robot needs to touch point on red line

- Incremental building up the mixture model by adding more components
- Context distribution will search for a good “free area”



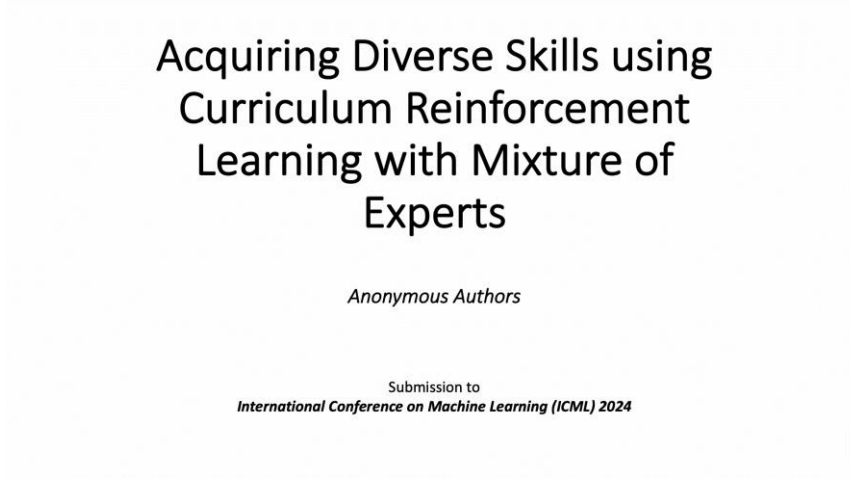


Examples: Versatile Skill Libraries


Acquiring Diverse Skills using Curriculum Reinforcement Learning with Mixture of Experts

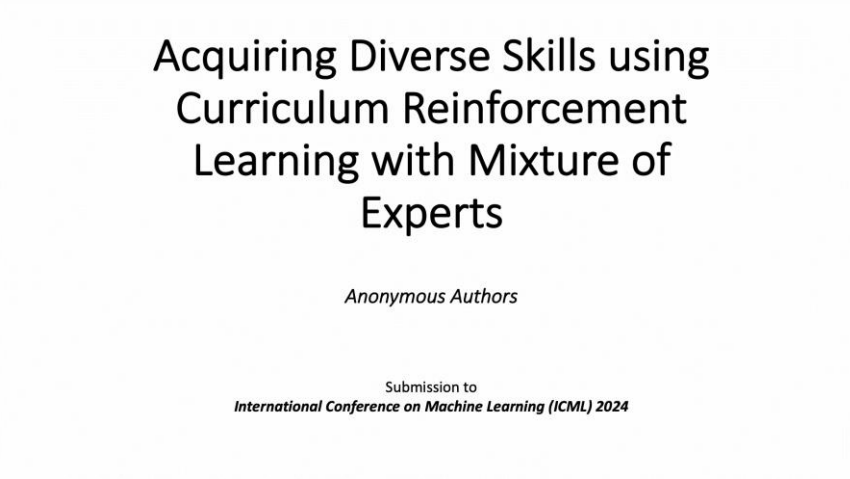
Anonymous Authors

Submission to
International Conference on Machine Learning (ICML) 2024


Acquiring Diverse Skills using Curriculum Reinforcement Learning with Mixture of Experts

Anonymous Authors

Submission to
International Conference on Machine Learning (ICML) 2024

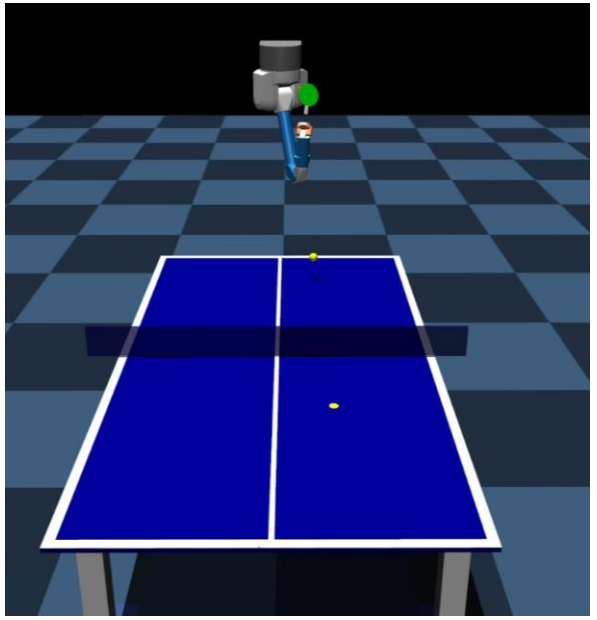

Acquiring Diverse Skills using Curriculum Reinforcement Learning with Mixture of Experts

Anonymous Authors

Submission to
International Conference on Machine Learning (ICML) 2024

Celik, O.;..., **Neumann, G**. *Specializing Versatile Skill Libraries using Local Mixture of Experts,* CoRL, 2021
Celik, O; ..., **Neumann, G**. *Acquiring Diverse Skills using Curriculum Reinforcement Learning with Mixture of Experts, ICML 24*

Wrap-Up – Reinforcemen Learning

Learning with MPs …

- Can often even be done as a black-box
- Induces better exploration (correlated in time, smooth) using MPs
- Requires principled trust region techniques
- Sample efficiency due to step-based off-policy updates
- Versatility can be achieved in trajectory space