Learning Latent Plans from Play

Authors: Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, Pierre Sermanet

Presenter: Ruchira Ray

11th October, 2022

Motivation

What? Multi-task Robotic Skill learning

Why is this important?

- One Robot, Many Tasks
- General-Purpose robots
- Reduced costs of automation - as one robot can handle multiple tasks

Main Problem

Technical challenges arising from the problem:

- ❖ **Lots of Labelled Data** and Segmented **Expert Demonstration** <u>per task</u>
- ❖ Designing **Policies** <u>per task</u>
- ❖ Manually Designing **Reward** <u>per task</u>

Reasons why prior approaches were lacking: **Need lots of human effort**

Key Insights

Things we need to overcome:

- Lack of labelled data
- Lack of demonstration
- hand-engineered reward and policy

We need the ability to reach any reachable goal state from any current state

How?

We consider “task” is no longer discrete, but continuous

Tasks and Skills are not Discrete

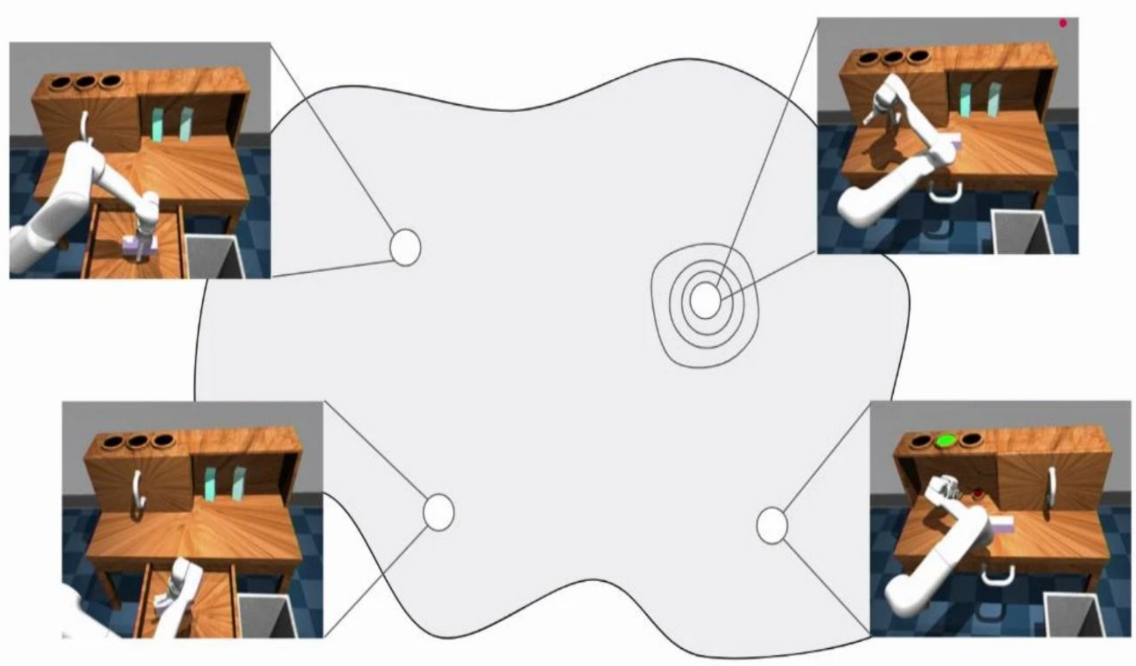
“Grasp fast?”

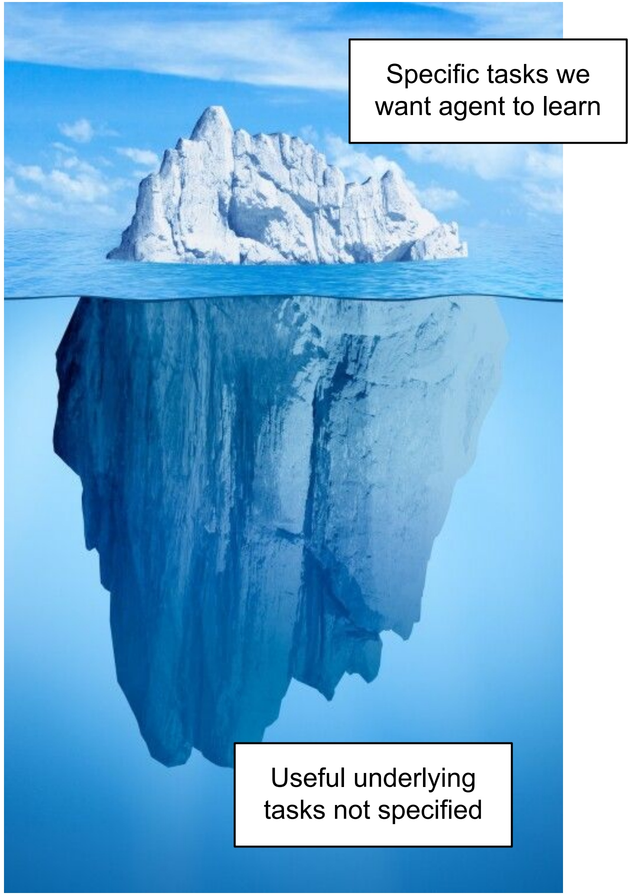
“Nudge + grasp?”

“Nudge slow?”

Hard to Differentiate + Hard to draw Boundaries between Tasks

Tasks and Skills are Continuous



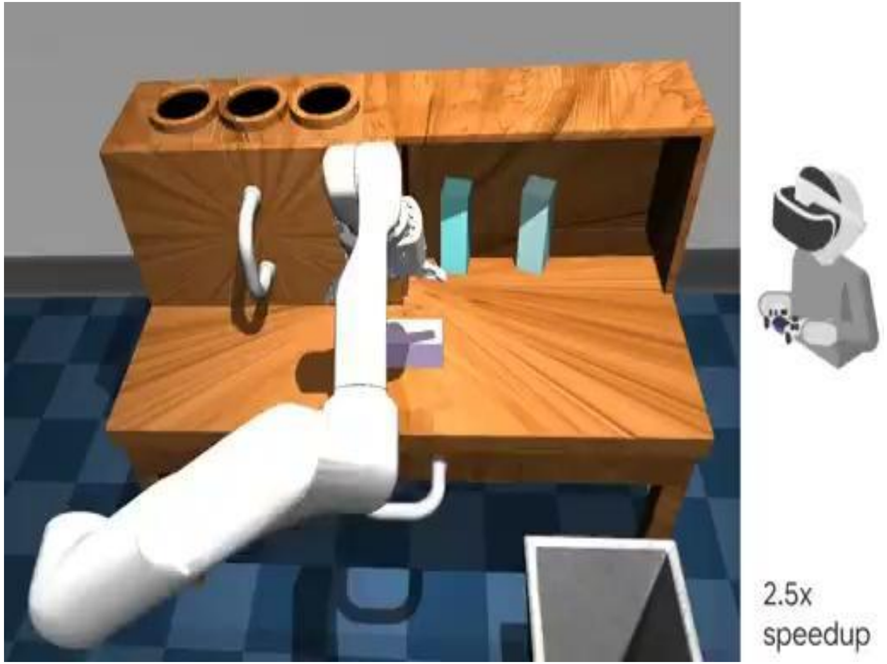


Play Data

The paper proposes to **Self-Supervise** on unlabelled **“Play”** data.

What? Non task-specific data collected from tele-operation.

Why? Cheap, Fast (no scene resets, segmentation, or task labeling), Rich and General



Problem Setting

We consider:

- **For Play data:** s_g (goal state) from s_c (current state) = $p(b|s_c,s_g)$
- Tele-operator samples : $b \sim p(b|s_c,s_g)$
- **Play-Supervised Goal-Conditioned Behavioral Cloning** (Play-GCBC)
  - D = play dataset consists of (O_t, a_t)
  - $O = \{I , p\}$
  - $\Phi = \{E_1 , ..., E_N \}$ (θ_Φ) encoder per sensory channel
  - $\pi_{GCBC}(a_t|s_t,s_g)$ = Goal-conditioned policy
  - actions τ
  - action a_t
  - κ-length sequence of observations

Problem Setting

- **Play-supervised Latent Motor Plans (Play-LMP)**
  - z = latent plan
  - $q_\varphi(z|\tau)$ = Latent Plan Space
  - V_{enc} = Video Encoder
  - Output parameters of a distribution in latent plan space μ_φ , σ_φ
  - π_{LMP}

Related Work + Limitations

Paper uses the concept of Learning from Demonstrations (Off-Policy), no use of RL.

Reasons why prior approaches were lacking:

- Used Meta-learning, Reinforcement Learning, few-shot learning etc.
- Discrete set of tasks
- Need predefined Task Distribution
- Did not cover a large range of skills/task - exploration was low

Proposed Approach

Key idea:

- ❖ Play-Supervised Goal-Conditioned Behavioral Cloning: A random window of (observation, action) pairs retrieved from play depicts how the robot progressed from a certain beginning state to a specific final state.

- ❖ Play-supervised Latent Motor Plans: learning representations of all the different high-level plans (p(b|sc,sg))and condition a policy on a single sampled plan.

Proposed Approach

Algorithm 1 Training Play-GCBC

1: **Input:** Play data $D : \{(s_1, a_1), \cdots, (s_T, a_T)\}$
2: **Input:** Window bounds: $\kappa_{low}, \kappa_{high}$
3: Randomly initialize model parameters $\theta = \{\theta_{GCBC}, \theta_\Phi\}$.
4: **while** not done **do**:
5: Sample a sequence length $\kappa \sim U(\kappa_{low}, \kappa_{high})$
6: Sample a sequence $\tau = \{(O_{t:t+\kappa}, O_{t:t+\kappa})\} \sim D$
7: Set encoded goal state: $s_g \leftarrow \Phi(O_{t+\kappa})$
8: Compute action loss $\mathcal{L}_{GCBC} = -\frac{1}{\kappa}\sum_{t=k}^{k+\kappa} log(\pi_{GCBC}(a_t|\Phi(O_t), s_g))$
9: Update θ by taking the gradient step to minimize $\mathcal{L}_{GCBC}$.

Algorithm 2 Training Play-LMP

1: **Input:** Play data $\mathcal{D} : \{(s_1, a_1), \cdots, (s_T, a_T)\}$
2: Randomly initialize model parameters $\theta = \{\theta_V, \theta_{CG}, \theta_{\pi LMP}, \theta_\Phi\}$
3: **while** not done **do**:
4: Sample a sequence $\tau = \{(O_{t:t+\kappa}, a_{t:t+\kappa})\} \sim \mathcal{D}$
5: Map raw observations in τ to encoded states: $\tau* = \Phi(\tau)$
6: Map encoded sequence to plan space: $\mu_\phi, \sigma_\phi = V_{enc}(\tau*)$
7: Set current and goal state: $s_i \leftarrow \Phi(O_t),\ s_g \leftarrow \Phi(O_{t+\kappa})$
8: Map encoded (current, goal) to plan space: $\mu_\psi, \sigma_\psi = CG_{enc}(s_t, s_g)$
9: Compute KL loss using Eq. 2.
10: Compute action loss using Eq. 3.
11: Update θ by taking a gradient step to minimize Eq. 4.

2. $$\mathcal{L}_{\text{KL}} = \text{KL}\Big(\mathcal{N}(z|\mu_\phi, \text{diag}(\sigma_\phi^2)) \,||\, \mathcal{N}(z|\mu_\psi, \text{diag}(\sigma_\psi^2))\Big)$$

3. $$\mathcal{L}_\pi = -\frac{1}{\kappa}\sum_{t=k}^{k+\kappa} log(\pi_{LMP}(a_t|s_t, s_g, z))$$

4. $$\mathcal{L}_{LMP} = \mathcal{L}_\pi + \beta\mathcal{L}_{\text{KL}}$$

Play-Supervised Goal-Conditioned Behavioral Cloning

1. Encoding perceptual inputs

$$s_t \leftarrow \Phi(O_t)$$

2. Goal-conditioned policy

$$\mathcal{L}_{GCBC} = -\frac{1}{\kappa}\sum_{t=k}^{k+\kappa} log(\pi_{GCBC}(a_t|s_t, s_g))$$

3. Multimodality problem

Play-supervised Latent Motor Plans

Multimodal policy learning problem -> Unimodal policy learning problem

1. Conditional sequence-to-sequence VAE (seq2seq CVAE)
   a. Plan recognition : $q_{\varphi}(z|\tau)$ latent plan space
   b. Plan proposal: $p_{\theta}(z|s_c, s_g)$
   c. Plan and goal-conditioned policy

2. Plan encoding: μ_{φ} , σ_{φ} = V_{enc} $(\tau*)$

3. Plan prior matching

$$\mathcal{L}_{\mathrm{KL}} = \mathbf{KL}\Big(\mathcal{N}(z|\mu_{\phi}, \mathbf{diag}(\sigma_{\phi}^2)) \;||\; \mathcal{N}(z|\mu_{\psi}, \mathbf{diag}(\sigma_{\psi}^2))\Big)$$

Play-supervised Latent Motor Plans

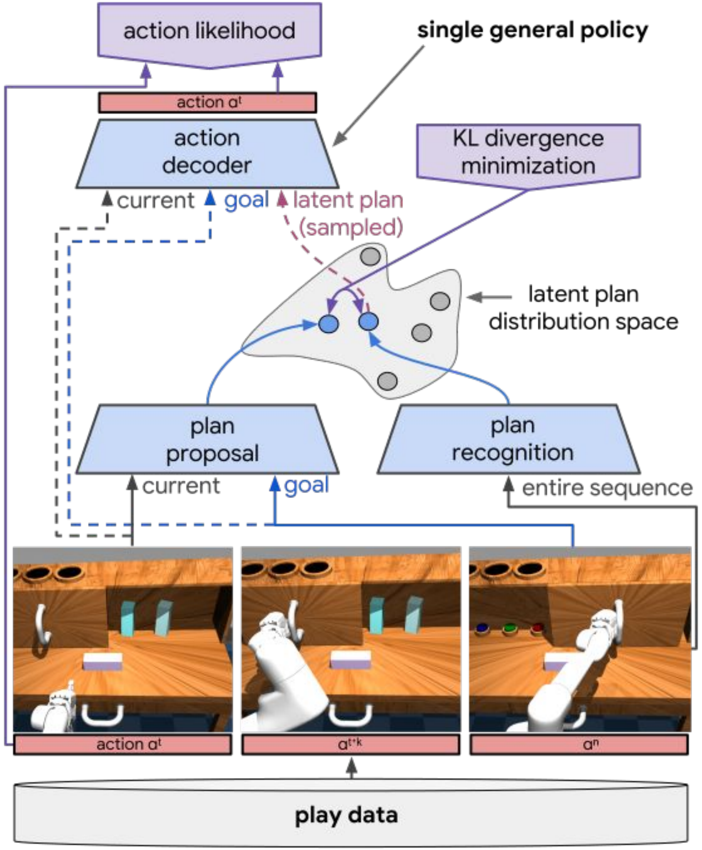
4. Plan decoding:

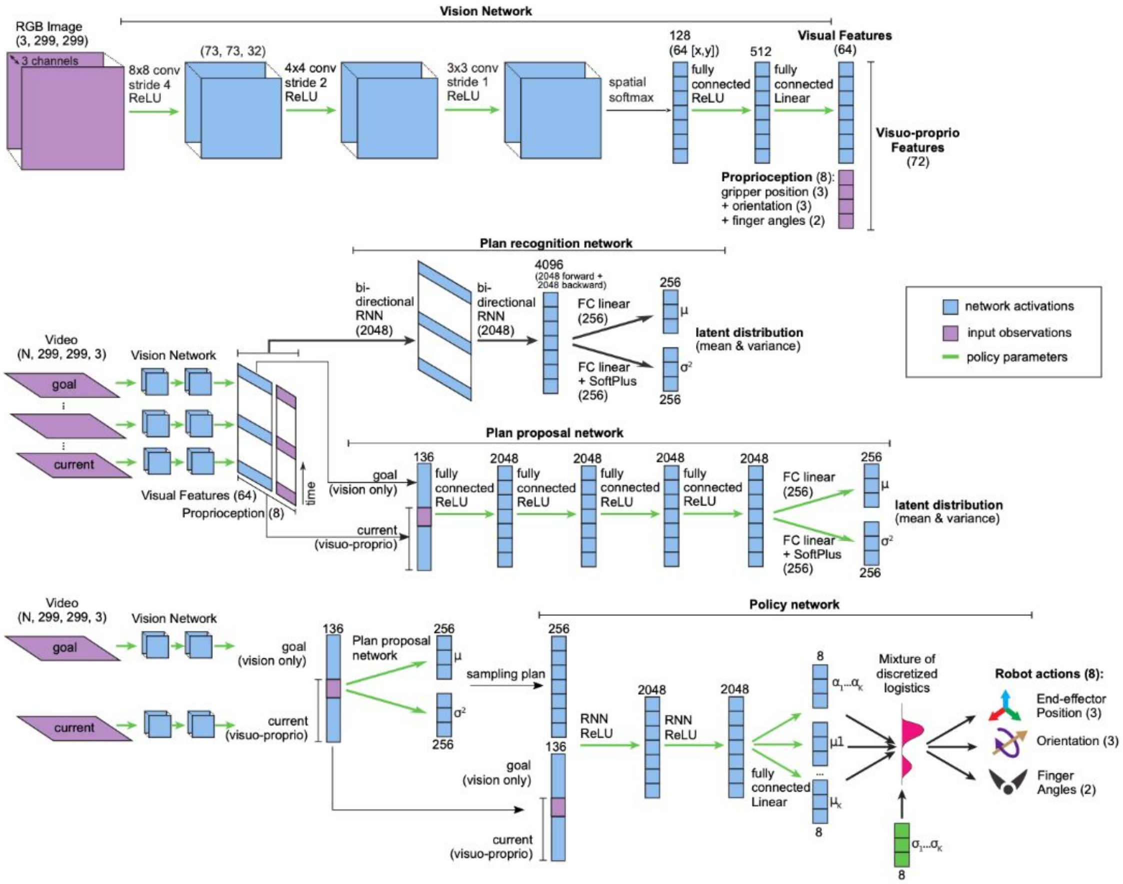
$$\mathcal{L}_{\pi} = -\frac{1}{\kappa}\sum_{t=k}^{k+\kappa} log\big(\pi_{LMP}(a_t|s_t, s_g, z)\big)$$

$$\mathcal{L}_{LMP} = \mathcal{L}_{\pi} + \beta\mathcal{L}_{\text{KL}}$$

5. Task-agnostic control at test time: "replan" by inferring and sampling new latent plans every κ timesteps

κ = 32





Architecture of Play-LMP

Theory

- Unsupervised Representation Learning of Plans and Control from Play
- pdata(x) = the true underlying process generating x ∈ X & D = dataset of i.i.d. samples from pdata(x)
- Consider the joint distribution p(x, z) over (x, z), where x ∈ X = points in the observed data space and z ∈ Z = points in a latent space
- Maximize the marginal log likelihood of the observations: log p_θ (x)
- Use Stochastic Gradient Variational Bayes (SGVB)

$$\log p_\theta(x) \geq -\mathrm{KL}\big(q_\phi(z|x) \,||\, p_\theta(z)\big) + \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]$$

Theory

For each observed window of state action pairs x of size κ sampled from play dataset D:

1) Given an observed context $c \leftarrow (s_c, s_g)$
2) From the conditional prior distribution $z \sim p_\theta(z|c)$ to draw a latent plan z. This is similar to our idea of a "operator drawing a high-level plan in order to reach a goal from a set of behaviors" $b \sim p(b|s_c,s_g)$.
3) Draw $x \sim p_\theta(x|c,z)$, the sequence of intervening states and actions between s_c and s_g according to context and plan-conditioned distribution.

Note that this is equivalent to a goal and plan-conditioned policy $\pi_\theta(a_t|s_c,s_g,z)$.

Theory

Three modules to implement:

1. Recognition network $q_\varphi(z|x,c)$
2. Conditional prior network $p_\theta(z|c)$
3. Generation network $p_\theta(x|z,c)$

Substitute back data variables obtained by self-supervised mining of windows from play to define each of Play- LMP's modules:

1. $q_\varphi(z|\tau) \leftarrow q_\varphi(z|x,c)$
2. $p_\theta(z|s_c,s_g) \leftarrow p_\theta(z|c)$
3. $\pi(a_t|s_c,s_g,z) \leftarrow p_\theta(x|z,c)$

Experimental Setup

(1) Can a single play-supervised policy be used for a wide range of user-specified visual manipulation tasks even though it wasn't trained on task-specific data?

2) Are play-supervised models trained on cheap to collect play data (LfP) as good as specialist models trained on expensive expert demonstrations for each task (LfD)?

3) Does Play-LMP improve performance over goal-conditioned behavioral cloning (Play-GCBC), which doesn't do explicit latent plan inference, by separating latent plan inference and plan decoding into separate problems?

Experimental Setup

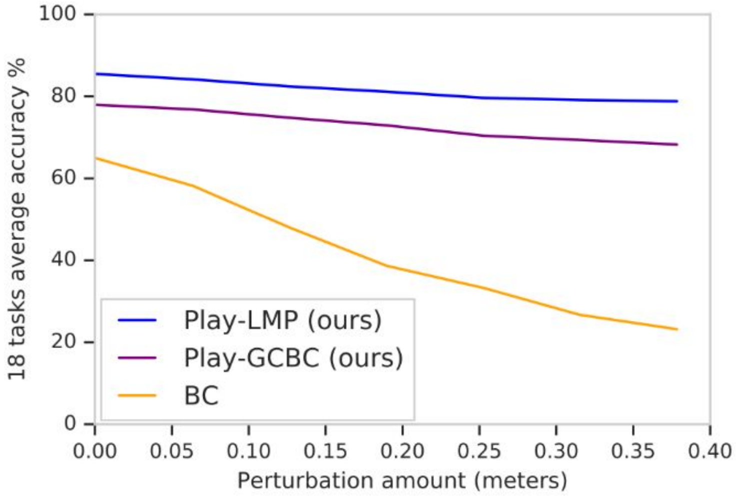
1. Mujoco HAPTIX system to collect teleoperation demonstration data
2. Simulation: 8-DOF agent (arm and gripper)
3. 18 visual manipulation tasks
4. 3 hours total of playground data and 100 positive demonstrations each of 18 tasks (1800 demonstrations total)
5. Train behavioral cloning policy, BC: 100 expert demonstrations per task
6. Train single multi-task behavioral cloning baseline, Multitask BC: same
7. Play-LMP and Play-GCBC : ~7 hours total Play Data
8. Metrics Used: Accuracy and Success

Results

We conduct :

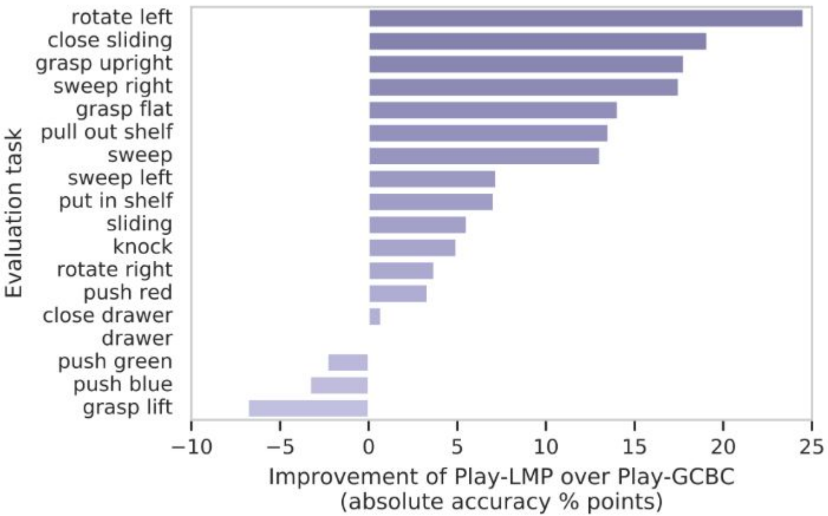
- Pixel experiments
- State experiments

Method	training data labels	input	success %
BC	labeled	pixels	66.5% ± 12.1
Play-GCBC (ours)	unlabeled	pixels	58.7% ± 11.6
Play-LMP (ours)	unlabeled	pixels	**69.4%** ± 10.8
BC	labeled	states	70.3%
Multitask BC	labeled	states	66.2%
Play-GCBC (ours)	unlabeled	states	77.9%
Play-LMP (ours)	unlabeled	states	**85.5%**

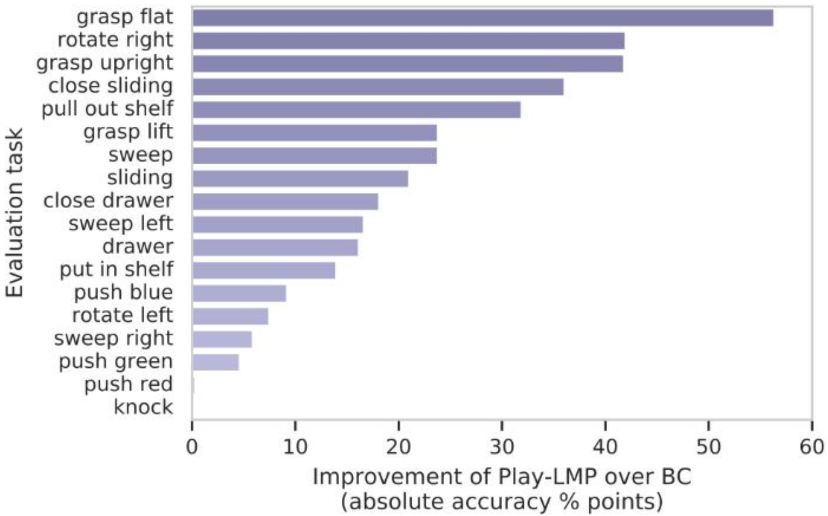


Perturbation Theory: a small change in a system which can be as a result of a third object interacting with the system

Results

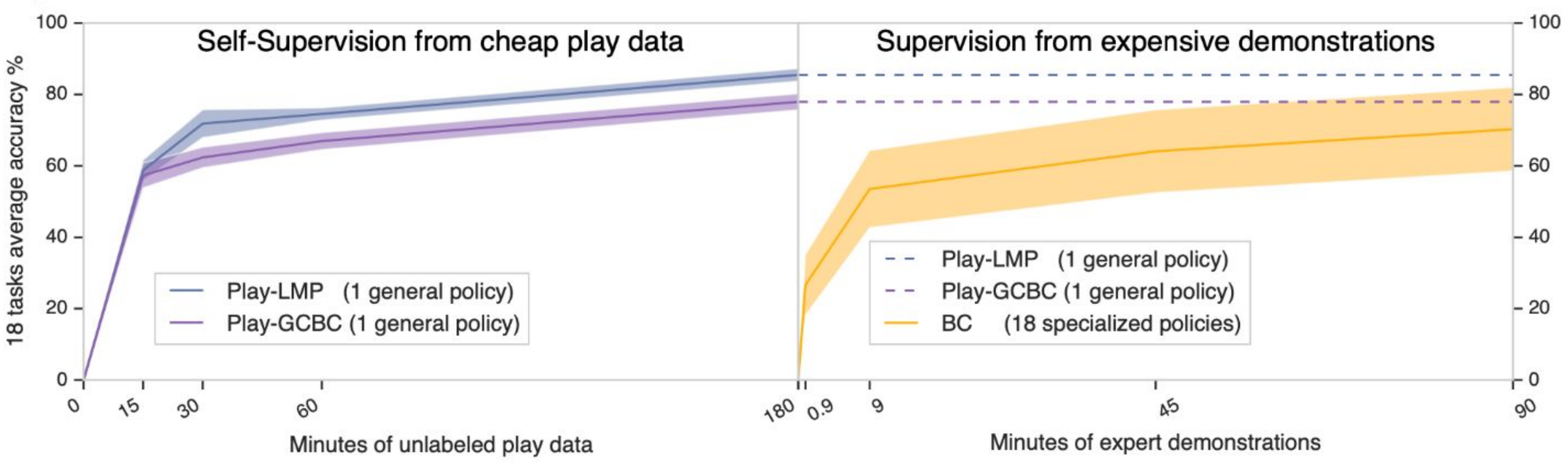


rotate left
close sliding
grasp upright
sweep right
grasp flat
pull out shelf
sweep
sweep left
put in shelf
sliding
knock
rotate right
push red
close drawer
drawer
push green
push blue
grasp lift
Evaluation task
−10
−5
0
5
10
15
20
25
Improvement of Play-LMP over Play-GCBC
(absolute accuracy % points)

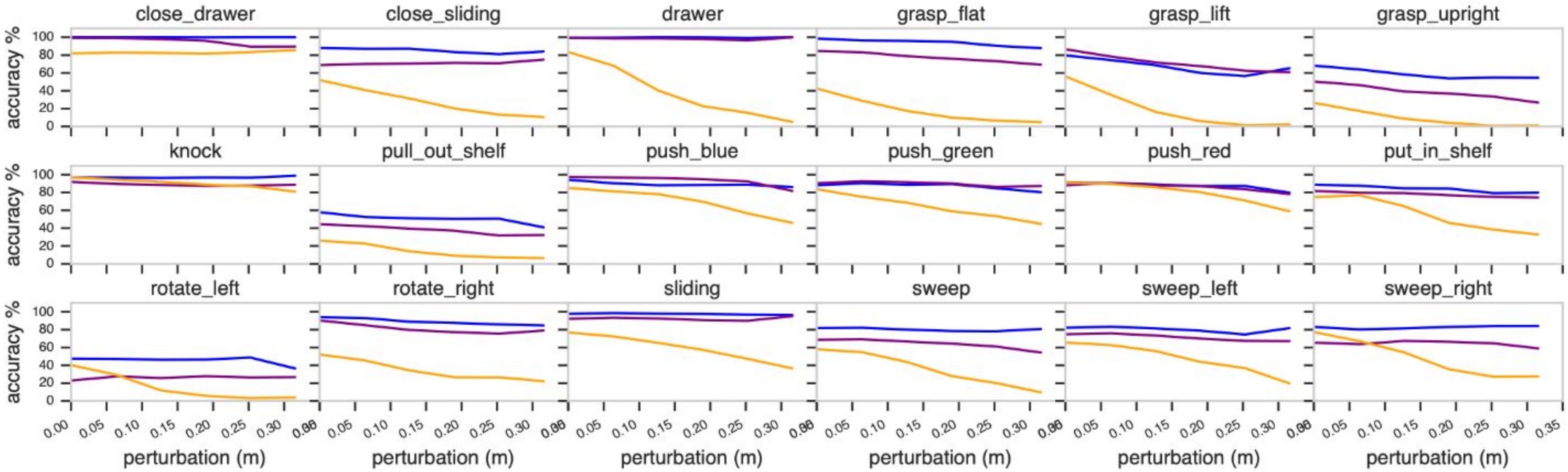


grasp flat
rotate right
grasp upright
close sliding
pull out shelf
grasp lift
sweep
sliding
close drawer
sweep left
drawer
put in shelf
push blue
rotate left
sweep right
push green
push red
knock
Evaluation task
0
10
20
30
40
50
60
Improvement of Play-LMP over BC
(absolute accuracy % points)

Results

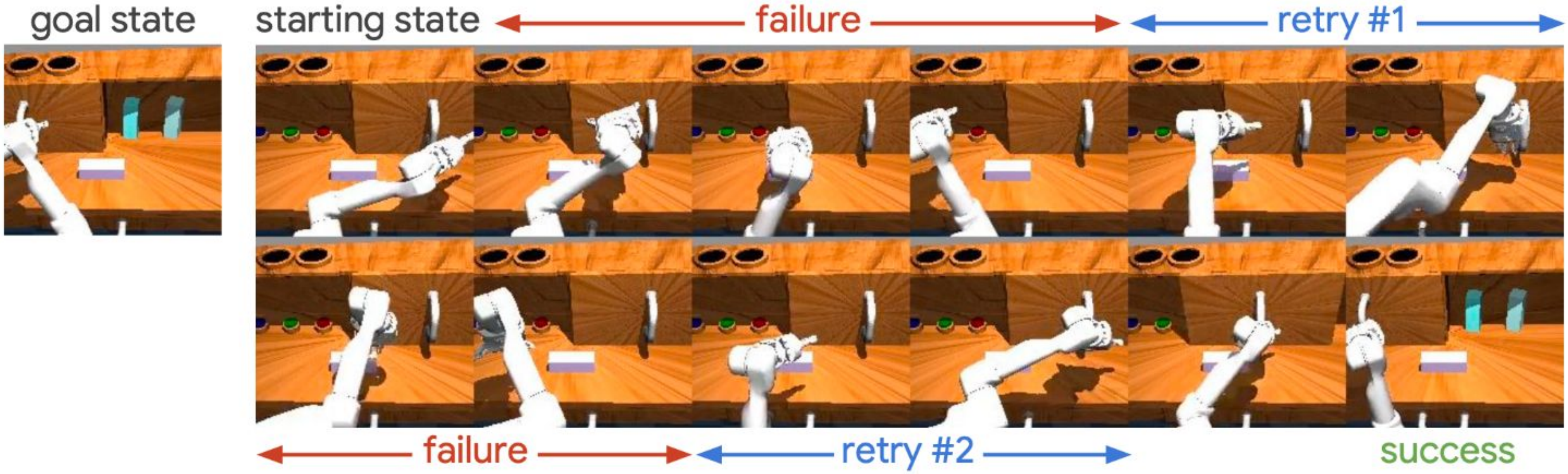

Self-Supervision from cheap play data
Supervision from expensive demonstrations
18 tasks average accuracy %
100
80
60
40
20
0
Play-LMP (1 general policy)
Play-GCBC (1 general policy)
BC (18 specialized policies)
0
15
30
60
180
0.9
9
45
90
Minutes of unlabeled play data
Minutes of expert demonstrations

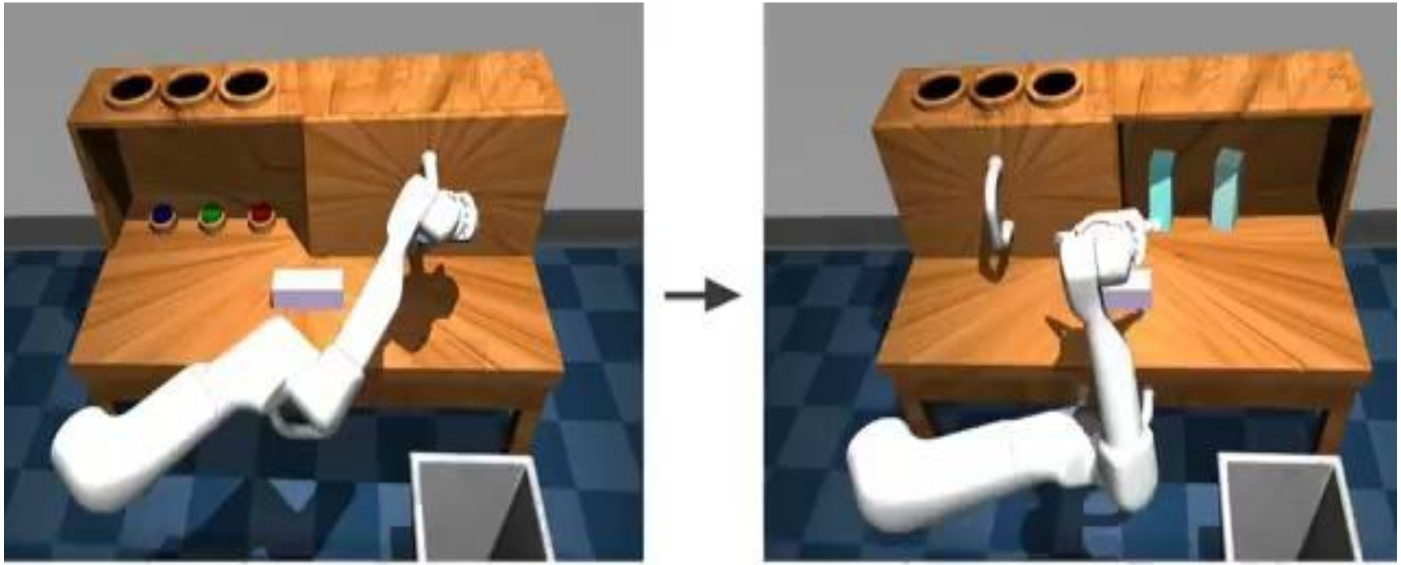
Robustness to Perturbations



Perturbation Theory: a small change in a system which can be as a result of a third object interacting with the system

Naturally Emerging Retrying Behaviour





Goal

Single Play-LMP policy

Critiques

Play data is super cool !

Follow ups:

1. Intra-task and Inter-task generalization?
2. Sim to Real Gap
3. Minutes of Unlabelled Time Data vs Expert Demonstration - Graph is not very accurate
4. How can we ensure that the present state to objective did not include any extra/unnecessary actions?
5. “Our model can in principle use any past experience for training, but the particular data collection approach we used is based on human-provided play data”. Would be interesting to see how well the model performs on existing datasets.

Future Work

- Visual grounding
- Leveraging cross-modal retrieval on play data
- Reducing human effort further - Augment data (Learning to Play by Imitating Humans)
- Learn object and action from play data for better learning

Extended Readings

- Learning and generalization of motor skills by learning from demonstration
- Unsupervised control through non-parametric discriminative rewards
- Playful Interactions for Representation Learning
- PLATO: Predicting Latent Affordances Through Object-Centric Play
- Learning to Play by Imitating Humans
- GTI: Learning to Generalize Across Long-Horizon Tasks from Human Demonstrations
- BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning

Summary

- One Robot, Many Tasks
- But then we need many policies, lots of expert demonstrations, handcrafted reward functions per task
- We consider tasks/skills are not discrete, but continuous.
- Use Play data
- Learn using demonstration in a self supervised manner
- Outperforms individual expert-trained policies on 18 user-specified visual manipulation tasks
- Robust to perturbations and retrying-till-success behaviors