Closing the sim-to-real loop: Adapting simulation randomization with real world experience

Presenter: **Shakeel** Ahamed Mansoor Shaikna

28th May 2020

Outline

- Introduction
- Related work
- Method
- Experiments
- Conclusion
- Future Work & Limitations

Introduction

- Transferring policies to the real world by training on a distribution of simulated scenarios.

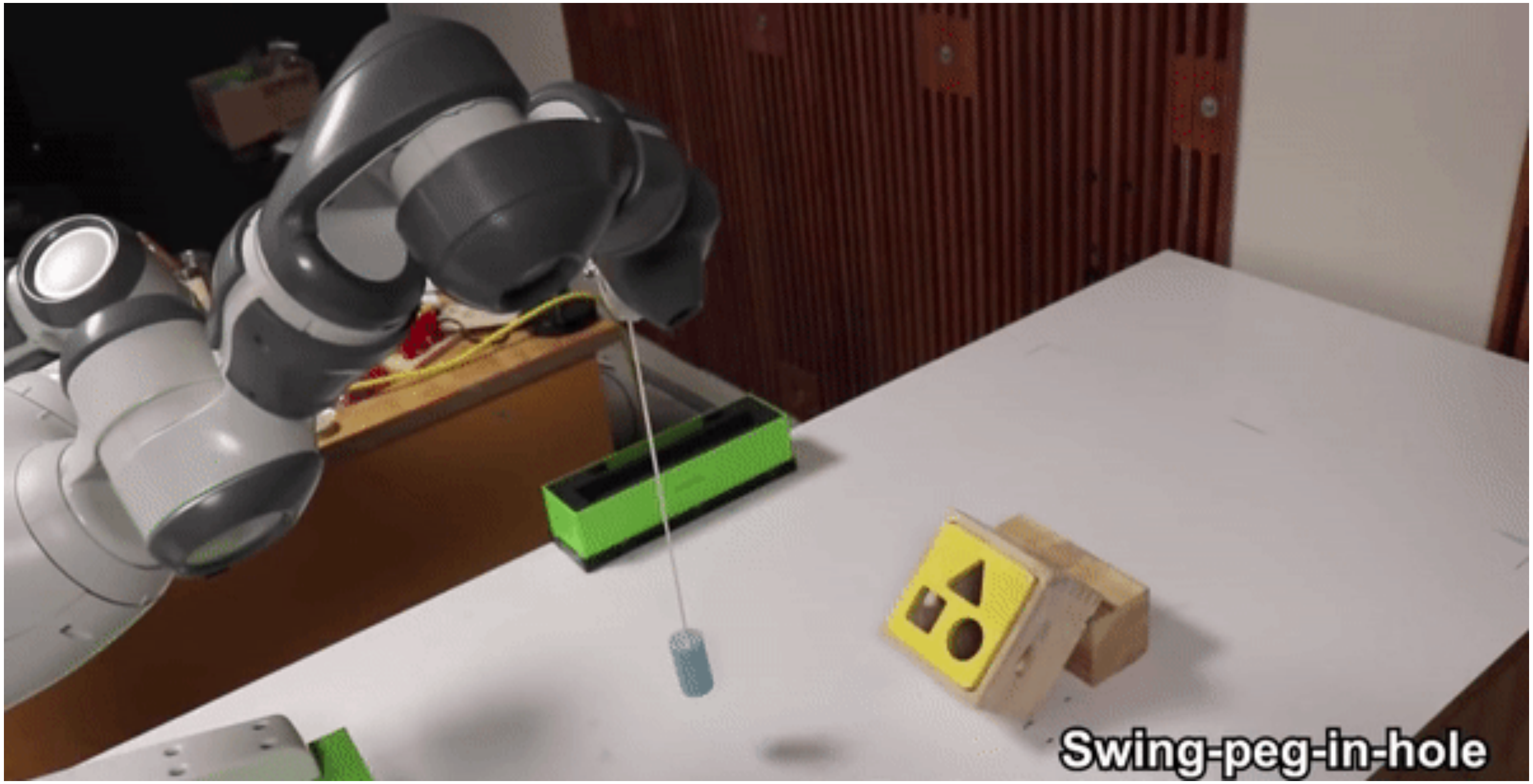
- Learning continuous control in real world complex environments has a wide interest.

- Policies learned in simulations cannot be directly applied on real world systems – Reality Gap.

- Data-driven approach and real world data to adapt the simulation randomization.

Task

- Swing-peg-in-hole



Task

- Opening a cabinet drawer

Related Work

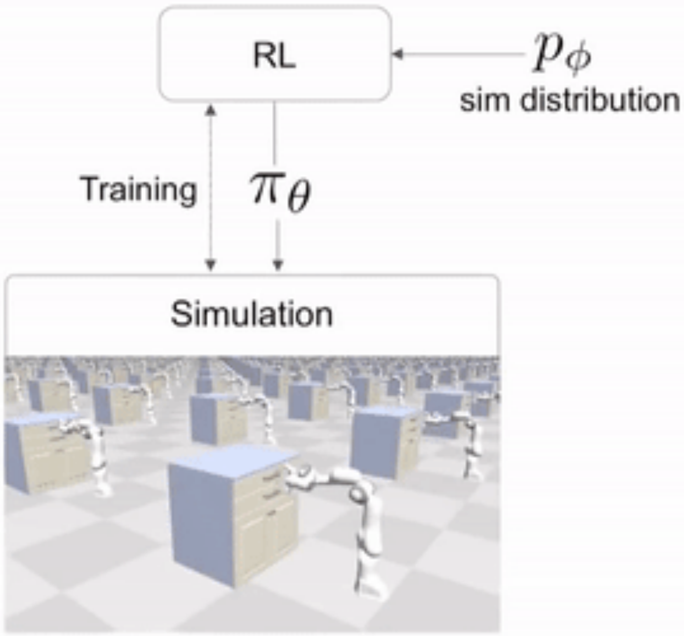
- **Domain randomization**: Training policies on a large diversity of simulated scenarios by randomizing relevant parameters.

- Combination of system identification and domain randomization has been used to learn locomotion for a real quadruped robots.

- **Adaptive EPOpt**: Optimizes a policy over a group of models and adapts the model distribution using data from the target domain.

Method

- Overview



Method

Simulation Randomization

- $\boldsymbol{M} = (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{p_0}, \gamma, \boldsymbol{T})$ a finite-horizon Markov Decision Process (MDP).

- A distribution of simulation parameters $\xi \sim p_\phi(\xi)$ parameterized by ϕ.

$$\max_\theta \mathbb{E}_{P_{\xi \sim p_\phi}} \left[\mathbb{E}_{\pi_\theta}[R(\tau)] \right]$$

Method

Simulation Randomization

Method

Learning simulation randomization

- Optimize the simulation parameter distribution to minimize the following objective,

$$\min_{\phi} \mathbb{E}_{P_{\xi \sim p_\phi}} \left[\mathbb{E}_{\pi_{\theta, p_\phi}} \left[D(\tau_\xi^{ob}, \tau_{real}^{ob}) \right] \right]$$

- Iterative approach is developed to approximate the optimization.

$$\min_{\phi_{i+1}} \mathbb{E}_{P_{\xi_{i+1} \sim p_{\phi_{i+1}}}} \left[\mathbb{E}_{\pi_{\theta, p_{\phi_i}}} \left[D(\tau_{\xi_{i+1}}^{ob}, \tau_{real}^{ob}) \right] \right]$$
$$\text{s.t.} \quad D_{KL}\left(p_{\phi_{i+1}} \| p_{\phi_i}\right) \leq \epsilon,$$

Method

Iterative Algorithm

Algorithm 1 SimOpt framework

$p_{\phi_0} \leftarrow$ Initial simulation parameter distribution
$\epsilon \leftarrow$ KL-divergence step for updating p_ϕ
for iteration $i \in \{0, \dots, N\}$ **do**
    env $\leftarrow$ Simulation(p_{ϕ_i})
    $\pi_{\theta, p_{\phi_i}} \leftarrow$ RL(env)
    $\tau^{ob}_{real} \sim$ RealRollout$(\pi_{\theta, p_{\phi_i}})$
    $\xi \sim$ Sample(p_{ϕ_i})
    $\tau^{ob}_{\xi} \sim$ SimRollout$(\pi_{\theta, p_{\phi_i}}, \xi)$
    $c(\xi) \leftarrow D(\tau^{ob}_{\xi}, \tau^{ob}_{real})$
    $p_{\phi_{i+1}} \leftarrow$ UpdateDistribution$(p_{\phi_i}, \xi, c(\xi), \epsilon)$

Method

Implementation

- RL training is performed on a GPU based simulator using a parallelized version of proximal policy optimization (PPO) on a multi-GPU cluster.

- Parameterized the simulation parameter distribution as a Gaussian, $p_\phi(\xi) \sim \mathcal{N}(\mu, \Sigma)$

- Weighted L1 and L2 norms is used for discrepancy function D,

$$D(\tau_\xi^{ob}, \tau_{real}^{ob}) = w_{\ell_1} \sum_{i=0}^{T} |W(o_{i,\xi} - o_{i,real})| + w_{\ell_2} \sum_{i=0}^{T} \|W(o_{i,\xi} - o_{i,real})\|_2^2$$

Experiments

- How does the method compare to standard domain randomization?
- How many SimOpt iterations and real world trials are required for a successful transfer of robotic manipulation policies?
- Does our method work for different real world tasks and robots?

Tasks

Swing-peg-in-hole:

- Task set up in the simulation and real world using a 7-DoF Yumi robot from ABB.

- Observation space consists of 7-DoF arm joint configurations and 3D position of the peg.

- Reward function for the RL training in simulation includes the distance, angle alignment with the hole and a binary reward for solving the task.

Tasks

Drawer opening:

- Task involves an ability to handle contact dynamics when grasping the drawer handle. Used 7-DoF Panda arm from Franka Emika for this task.

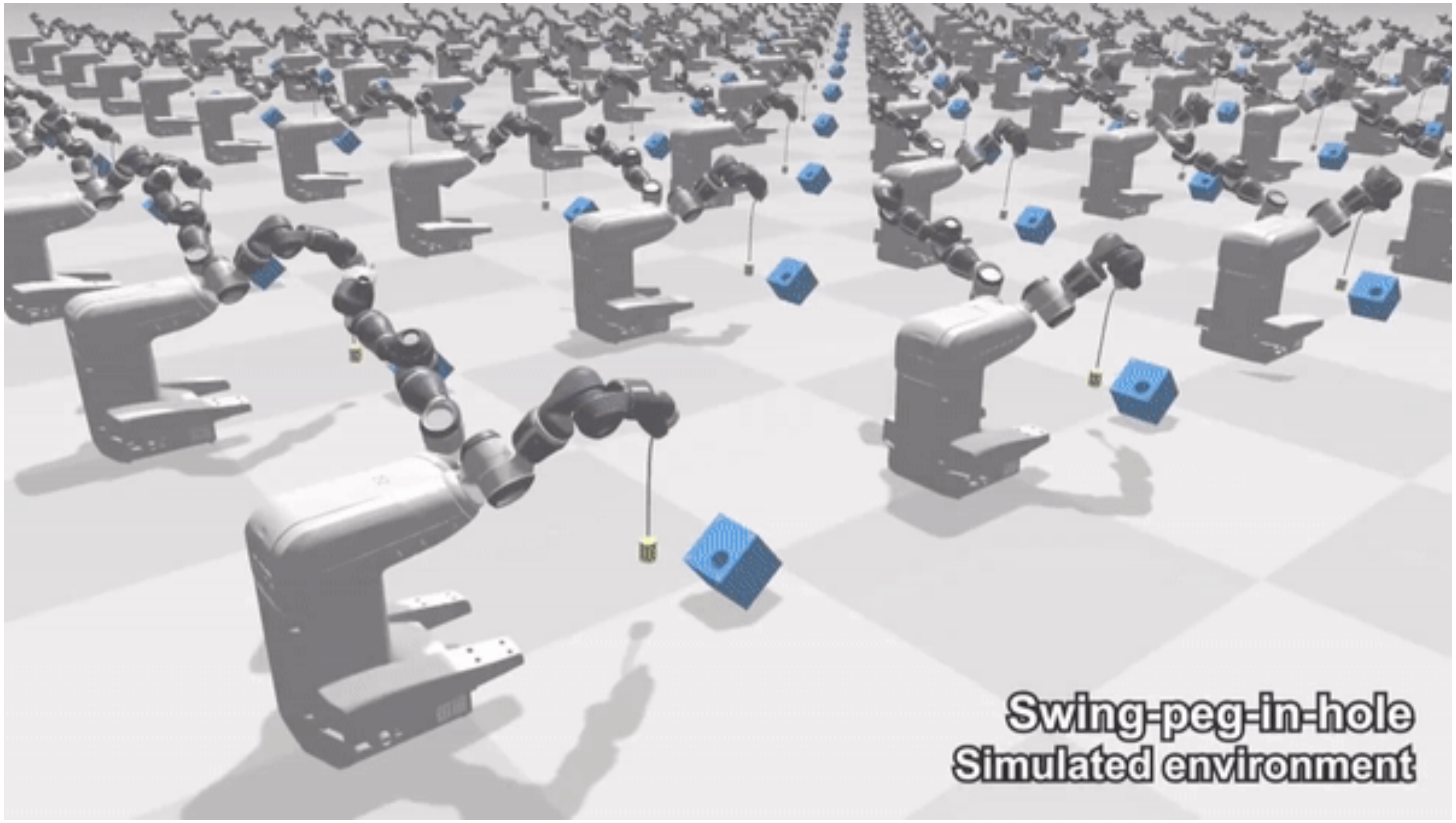
- Operated on a 10D observation space: 7D robot joint angles and 3D position of the cabinet drawer handle.

- Reward function consists of the distance penalty, angle alignment of the end effector and handle, the opening distance of the drawer.

Robots

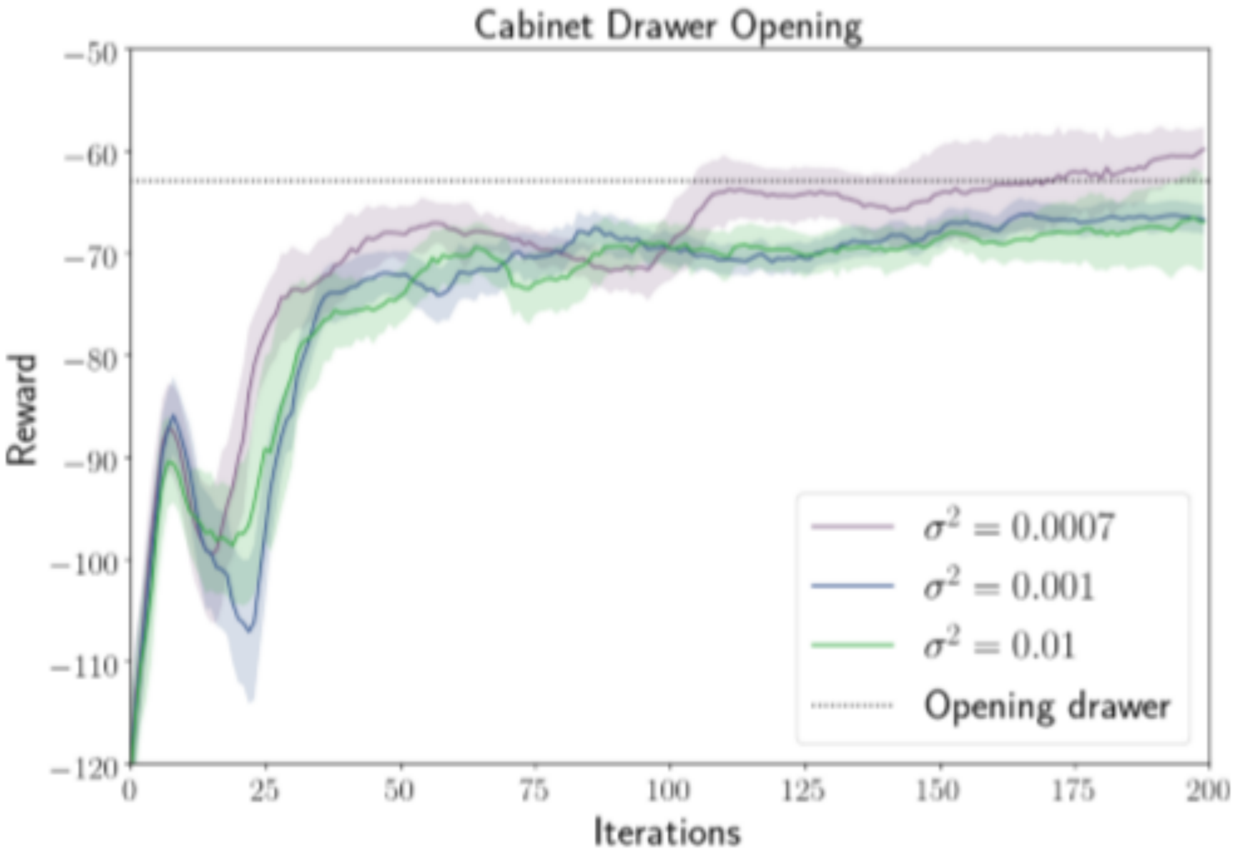
Simulation Engine

- NVIDIA Flex as a high-fidelity GPU based physics simulator that uses maximal coordinate representation to simulate rigid body dynamics.

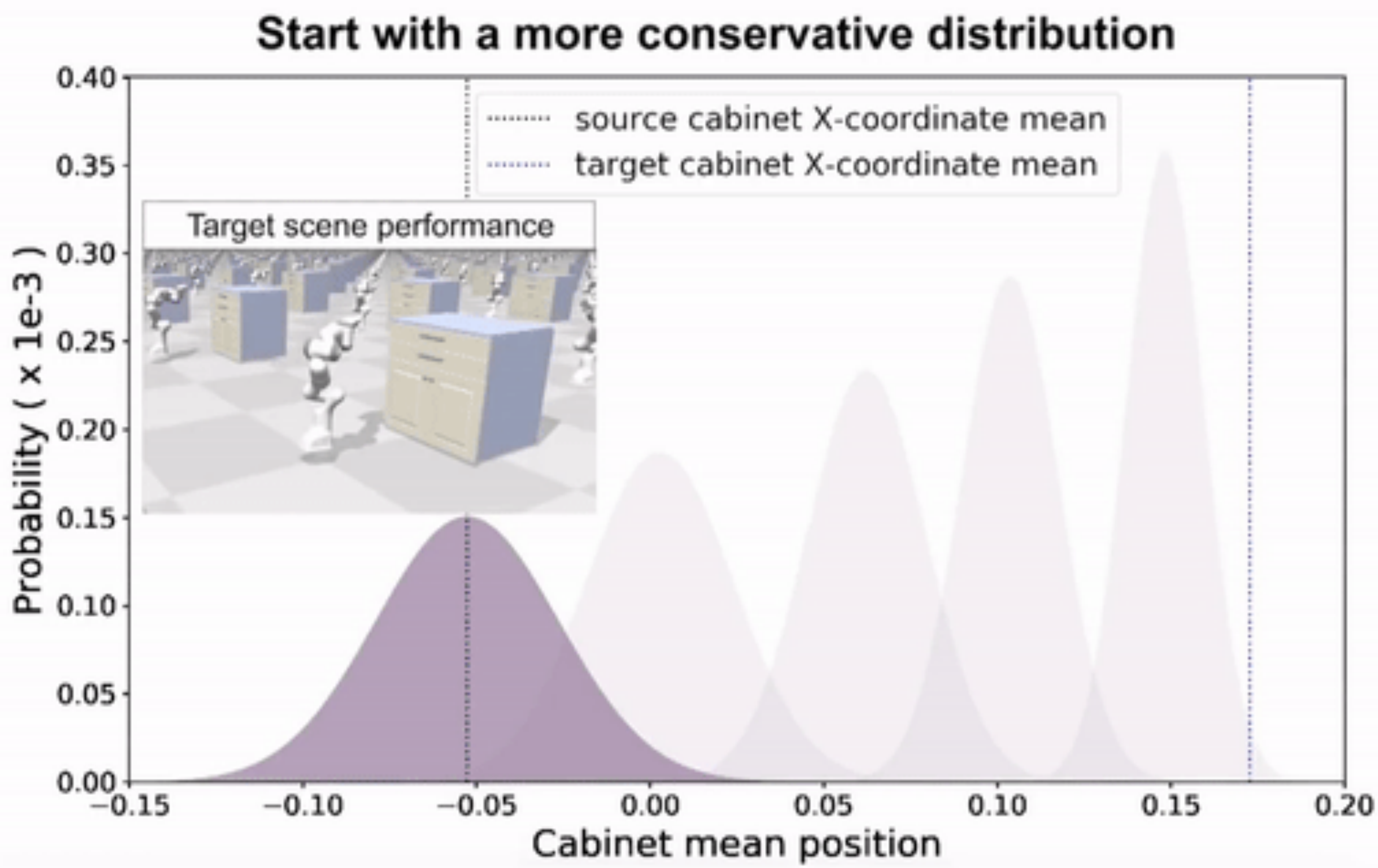


Comparison: Domain randomization

- Randomize the position of the cabinet along the lateral direction (X-coordinate) while keeping all other simulation parameters constant.



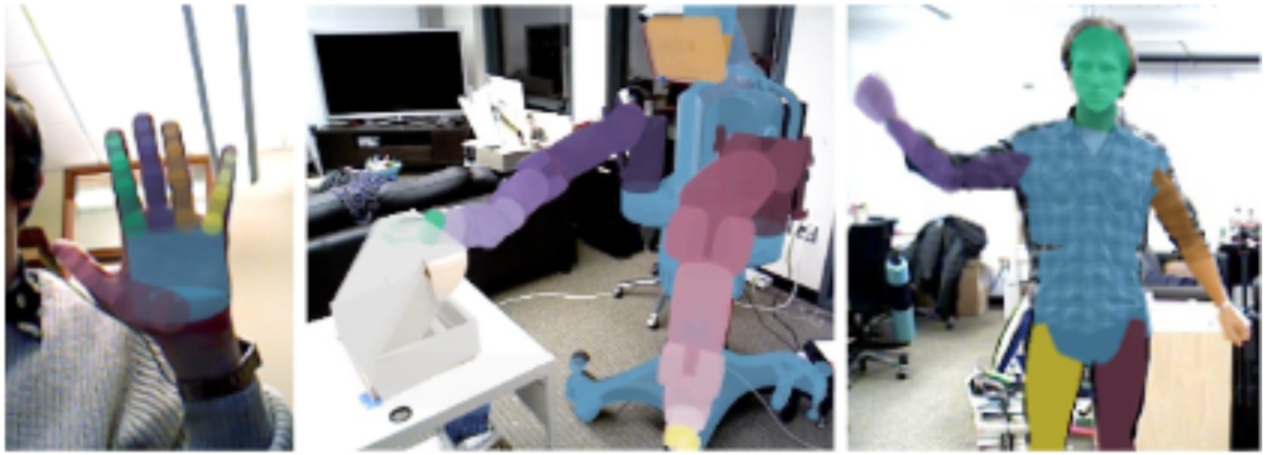
Comparison: Domain randomization



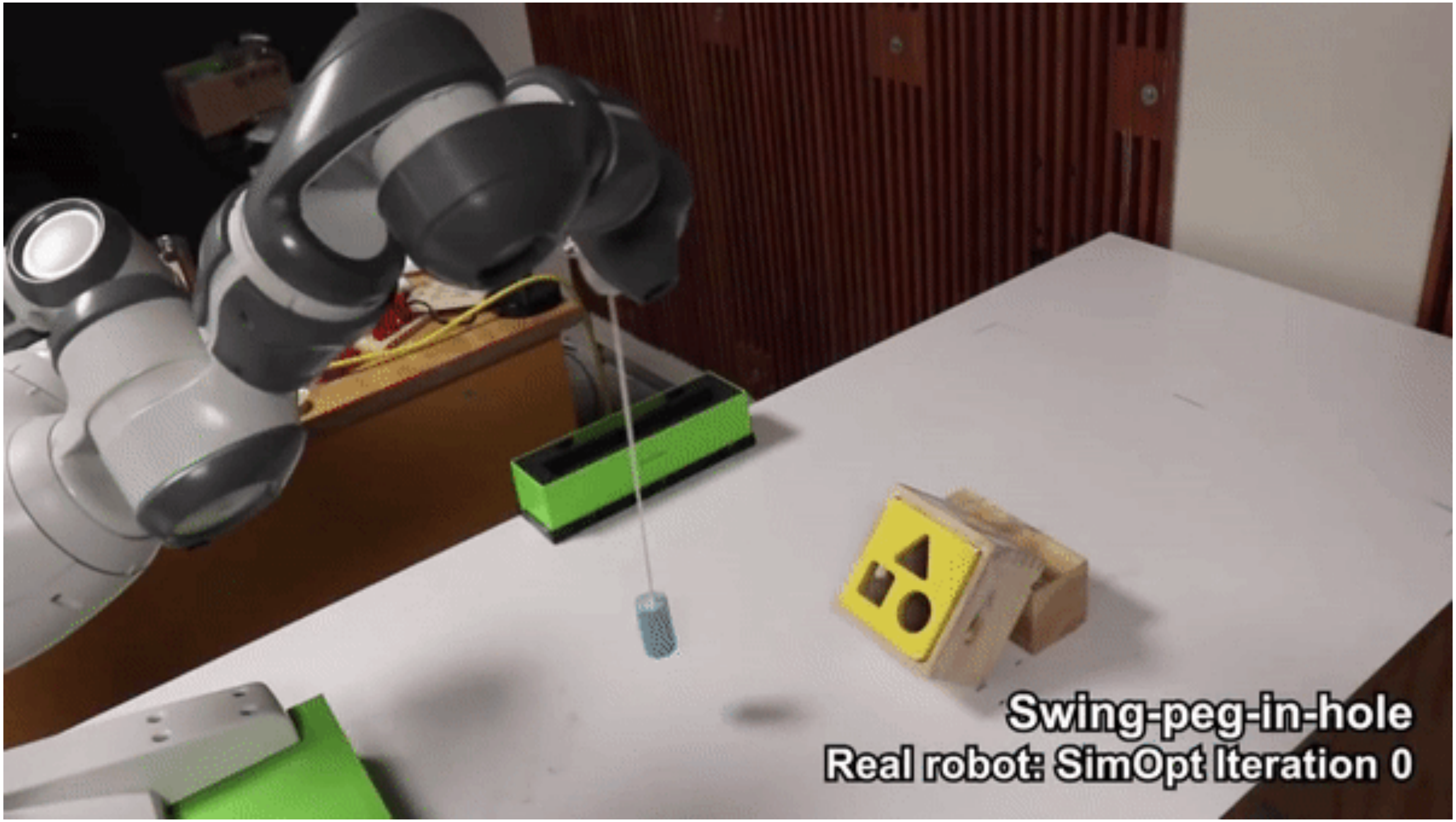
Real Robot Experiments

- Object Tracking: To continuously track the 3D positions of the peg and the handle of the cabinet drawer DART is used.

- DART operates on depth images and requires 3D articulated models of the objects.



Swing-peg-in-hole



Drawer opening

Conclusion

- Adapting simulation randomization using real world data can help in learning simulation parameter distributions.

- Updating simulation distributions is possible using partial observations of the real world.

- Evaluated on two real world robotic tasks and policies can be transferred with only a few iterations of simulation updates.

Future Work & Limitations

- Extend the framework to multi-modal distributions and more complex generative simulation models.

- Incorporate higher-dimensional sensor modalities, such as vision and touch, for both policy observations and factors of simulation randomization.

- Generalization of the method.

- Model may be overfitted in the drawer opening task.

- Initial simulation distribution calculation.

Thank You ☺