Simulator-Predictive Control: Using Learned Task Representations and MPC for Zero-Shot Generalization and Sequencing

Zhanpeng He*†, Ryan Julian*†, Eric Heiden†, Hejia Zhang†, Joseph J. Lim†, Gaurav Sukhatme†, Stefan Schaal†, and Karol Hausman‡

*Equal Contribution †University of Southern California ‡Google AI

Abstract

We propose a method for zero-shot learning of motion tasks which combines sim2real tranfer, learned task embeddings, and model-predictive control (MPC).

Our method:

- Learns an **embedding space of motion tasks** which can be explored and sampled
- **Explores in a latent task space**, which can be much more efficient than exploring in a high-dimensional action space
- Transfers motion skills from simulation to real **without fine-tuning or explicit alignment**
- Composes **primitive tasks into complex sequences**
- Runs in **real-time on a real robot** with joint-space control

Task Embedding Algorithm

Our method learns the task encoder p_ϕ, policy network π_θ, and trajectory decoder q_ψ simultaneously.

Using the variational inference framework:

- $p_\phi(z|t)$ and $\pi_\theta(a|s)$ can be thought of together as the **encoder** from latent tasks z to trajectories τ
- $q_\psi(z|\tau)$ can be thought of as the **decoder** from trajectories τ to latent tasks z.

The method can be used with *any parametric reinforcement learning algorithm*. This work uses PPO.

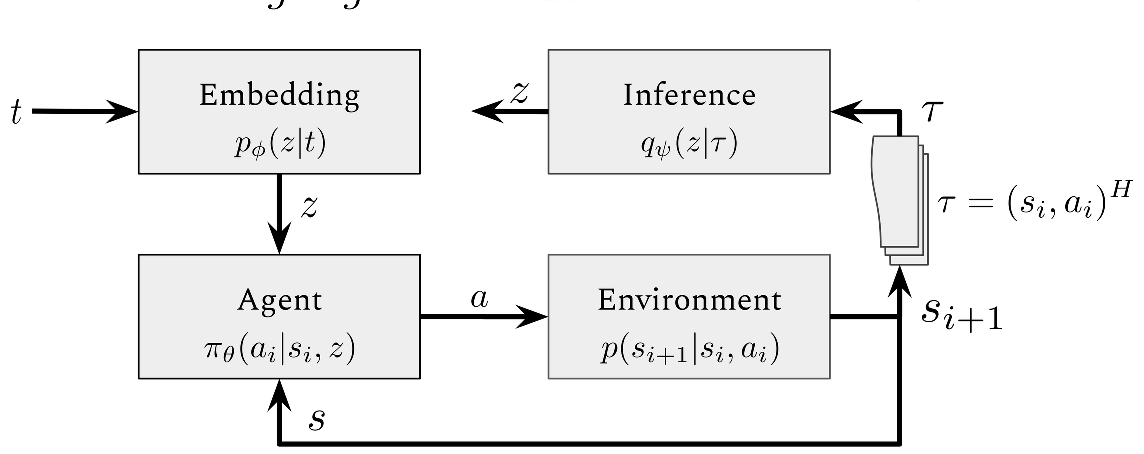

Figure 1: Task Embedding Algorithm Architecture

Augmented RL Loss

$$\mathcal{L}(\theta, \phi, \psi) = \mathbb{E}_{\pi_\theta(a,z|s,t)}\left[\sum_{i=0}^{\infty} \gamma^i \hat{r}(s_i, a_i, z, t)\right] + \alpha_1 \mathbb{E}_{t \in \mathcal{T}} \mathcal{H}\left[p_\phi(z|t)\right]$$

where

$$\hat{r}(s_i, a_i, z, t) = r_t(s_i, a_i) + \alpha_2 \log q_\psi(z|\tau = (s_i, a_i)^H) + \alpha_3 \mathcal{H}\left[\pi_\theta(a|s, z)\right]$$

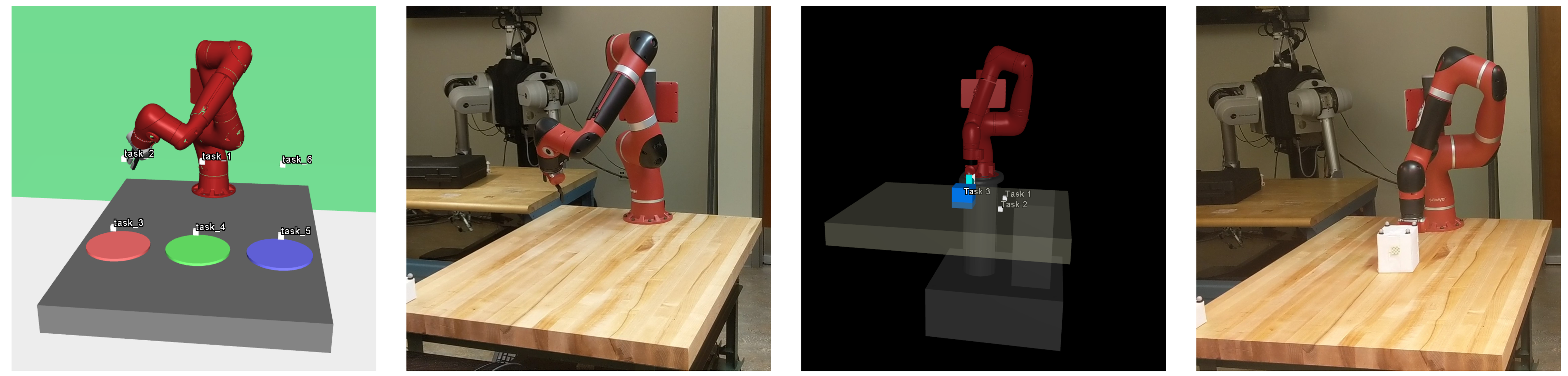

Figure 2: The Sawyer robot performing the reaching (left) and pushing (right) tasks in simulation and real world

Takeaway

We show how to use the simulation from the pre-training step of sim2real methods as a **tool for foresight**, allowing an embedded task to policy **zero-shot adapt to unseen tasks**.

MPC in the Latent Space

We generalize to new tasks by performing MPC on the latent space input of the pre-trained policy. Importantly, we use MPC to **search in the simulation environment** from pre-training, but use those actions to **execute in the real environment**.

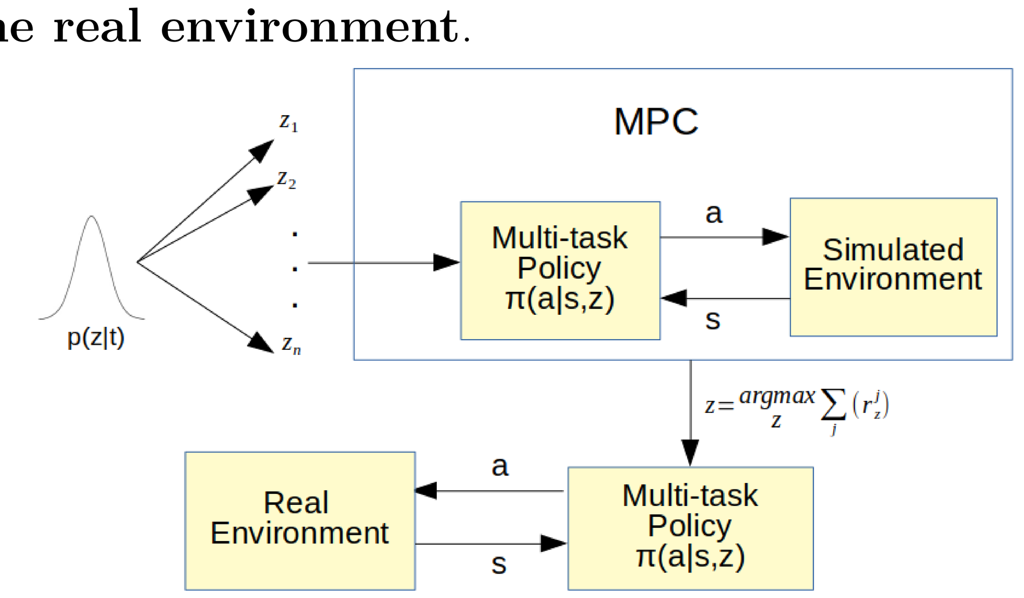

Figure 3: Simulator-Predictive Control

Algorithm 1 MPC in Task Latent Space

while t^{new} is not complete **do**
  Sample $\mathcal{Z} = \{z_1, \dots, z_k\} \sim \mathbb{E}_{t \sim p(t)} p_\phi(z|t)$
  Observe state s_{real} from real environment $\mathcal{R}$
  {1. Search in simulated environment $\mathcal{S}$ with horizon T}
  for $z_i \in \mathcal{Z}$ **do**
    Set initial state of $\mathcal{S}$ to s_{real}
    $(s_j, a_j)^T$ = rollout($\mathcal{S}$, $\pi_\theta(\cdot|\cdot, z_i)$, T)
    Calculate $R_i^{\text{new}} = \Sigma_{j=0}^{T} \gamma^j r^{\text{new}}(s_j, a_j)$
  end for
  Choose $z^* = \text{argmax}_{z_i} R_i^{\text{new}}$
  {2. Execute in real environment $\mathcal{R}$ for N timesteps}
  rollout($\mathcal{R}$, $\pi_\theta(\cdot|\cdot, z^*)$, N)
end while

Experiments

We demonstrate our method with three experiments which challenge SPC and the Sawyer robot to adapt to unseen tasks in real-time.

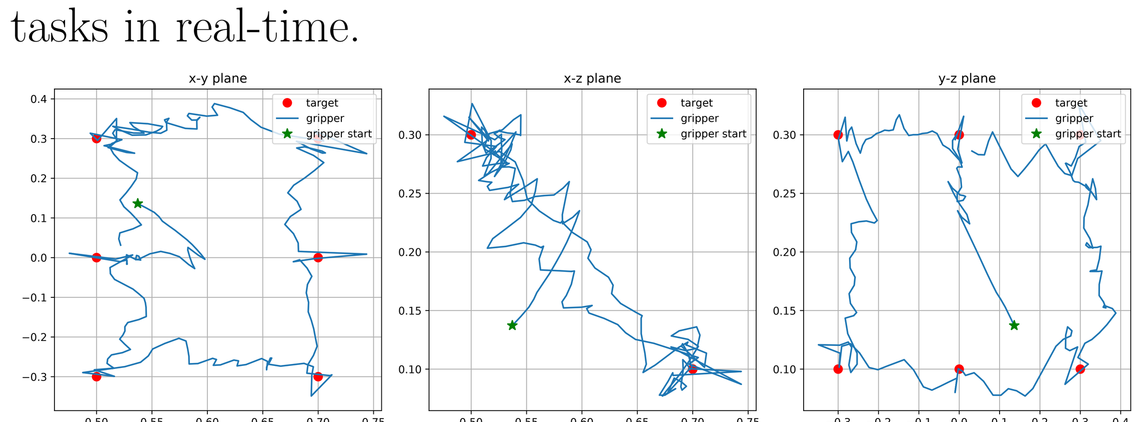

Figure 4: Gripper position plots for the rectangle-drawing experiment in simulation. The pre-trained embedded policy for the triangle- and rectangle-drawing experiments were pre-trained on only 8 reaching tasks, and uses joint-space control.

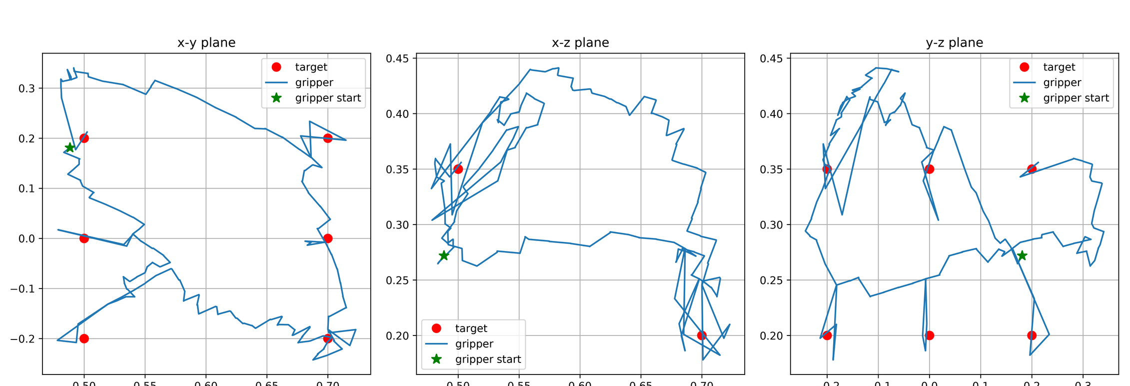

Figure 5: Gripper position plots for the rectangle-drawing experiment on a Sawyer robot.

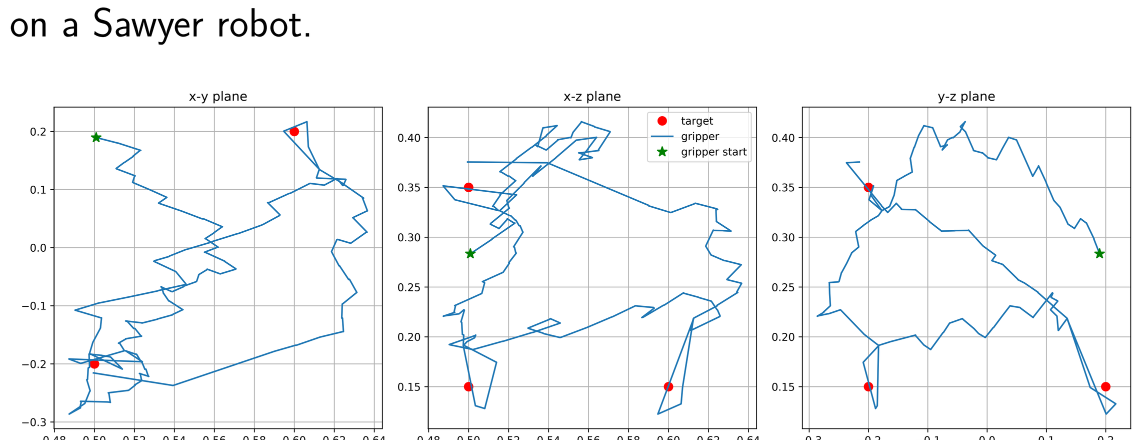

Figure 6: Gripper position plots for the triangle-drawing experiment on a Sawyer robot.

Experiments

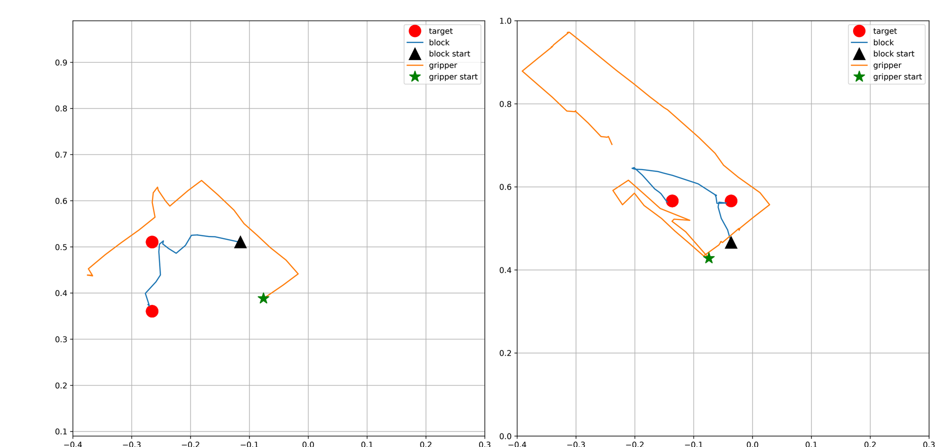

Figure 7: Block position plots for the block-pushing experiment on a Sawyer robot. (Left) the robot pushes the box left-then-down. (Right) the robot push the box up-then-left. The embedded policy is pre-trained only to push {up, down, left, right} from a single starting position, and uses task-space control.

Conclusion

Our results show:

- We can use SPC to **achieve unseen tasks by composing and sequencing in the latent space**
- The method is efficient-enough to **adapt to new tasks in real-time while executing on a real robot**
- **SPC results in intelligent behaviors** (e.g. the SPC pusher recovers from a mistake not encountered during pre-training)

References

[1] K. Hausman, J. Springenberg, Z. Wang, N. Heess, and M. Riedmiller, "Learning an embedding space for transferable robot skills," in *ICLR*, 2018.

[2] R. Julian, E. Heiden, Z. He, H. Zhang, S. Schaal, J. J. Lim, G. S. Sukhatme, and K. Hausman, "Scaling simulation-to-real transfer by learning composable robot skills," in *ISER*, 2018.

[3] J. D. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach., P. Abbeel, and S. Levine, "Self-Consistent Trajectory Autoencoder: Hierarchical reinforcement learning with trajectory embeddings," in *ICML*, 2018.

More Information

- arXiv: arxiv.org/abs/1810.02422
- Code: github.com/ryanjulian/embed2learn
- Supplemental Video: youtu.be/te4JWe7LPKw
- Email: {zhanpenh, rjulian}@usc.edu

USC University of Southern California Google AI