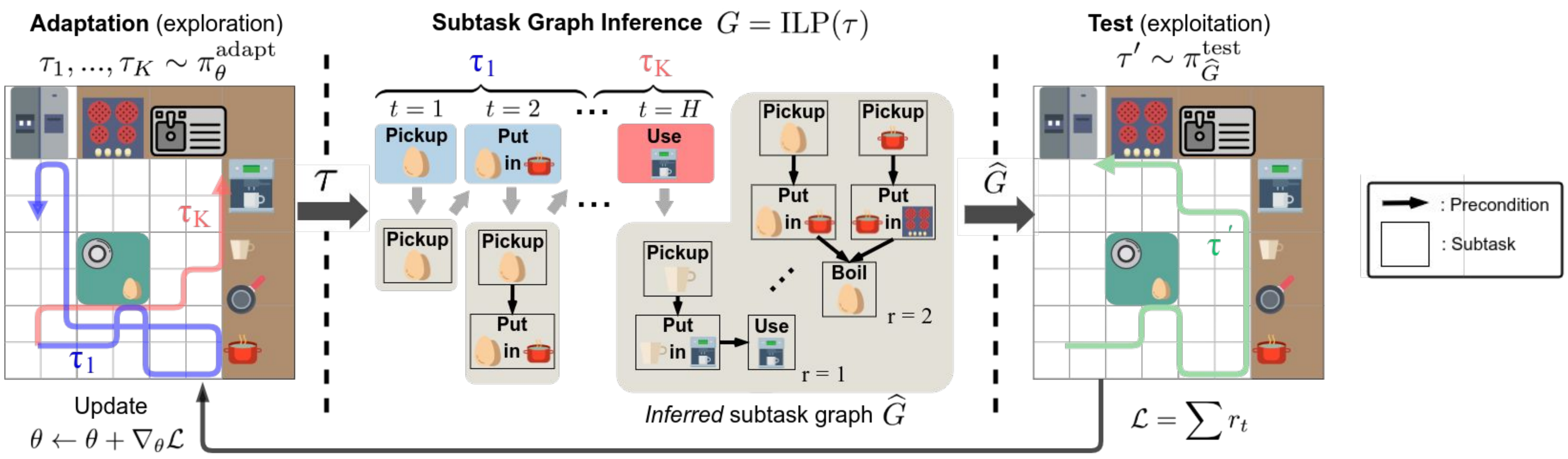
Meta Reinforcement Learning with Autonomous Inference of Subtask Dependencies

Sungryull Sohn[1] Hyunjae Woo[1] Jongwook Choi[1] Honglak Lee[1, 2] srsohn@umich.edu

Introduction

- **Summary**
  - **Goal**: Solve a meta RL problem by *explicitly inferring the underlying task structure* in order to quickly adapt to the task
  - **Idea**: Infer the subtask reward and dependencies in adaptation phase, and solve the task in test phase
  - **Result**: Better generalization and performance on complex tasks
- **Subtask graph inference problem**
  - A task is characterized by the **subtask graph** which defines the subtasks, their *rewards* and *preconditions*.
  - The subtask graph is **unknown** to the agent
  - Few-shot RL: after K episodes of adaptation, the agent needs to maximize the return in test phase.



Method

- Meta-learner with Subtask Graph Inferencer (MSGI)
  - **Adaptation**: Adaptation policy ($\pi_\theta^{\text{adapt}}$) learns to efficiently collect experience to accurately infer the task
  - **Inference**: ILP algorithm infers the subtask reward and precondition in the form of subtask graph $\widehat{G} = (\widehat{G}_{\mathbf{r}}, \widehat{G}_{\mathbf{c}})$
  - **Execution**: execution policy ($\pi_{\widehat{G}}^{\text{test}}$) executes the inferred subtask graph $\widehat{G}$

Training

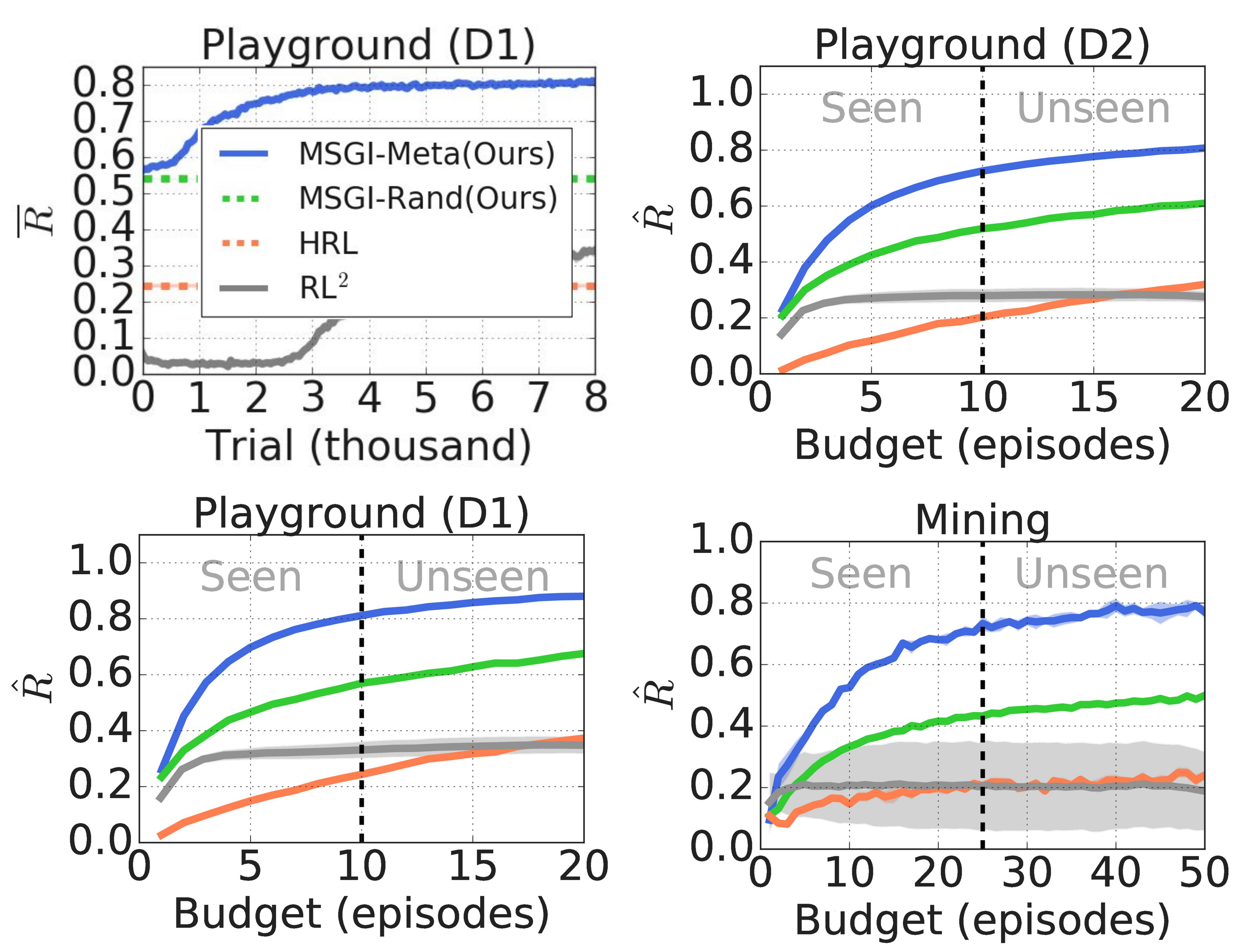
- Adaptation policy
  - **UCB**-like intrinsic bonus: $r_t^{\text{UCB}} = w_{\text{UCB}} \cdot \mathbb{I}(\mathbf{x}_t \text{ is novel})$
  - Trained via actor-critic method with GAE [1]
- Subtask graph inference

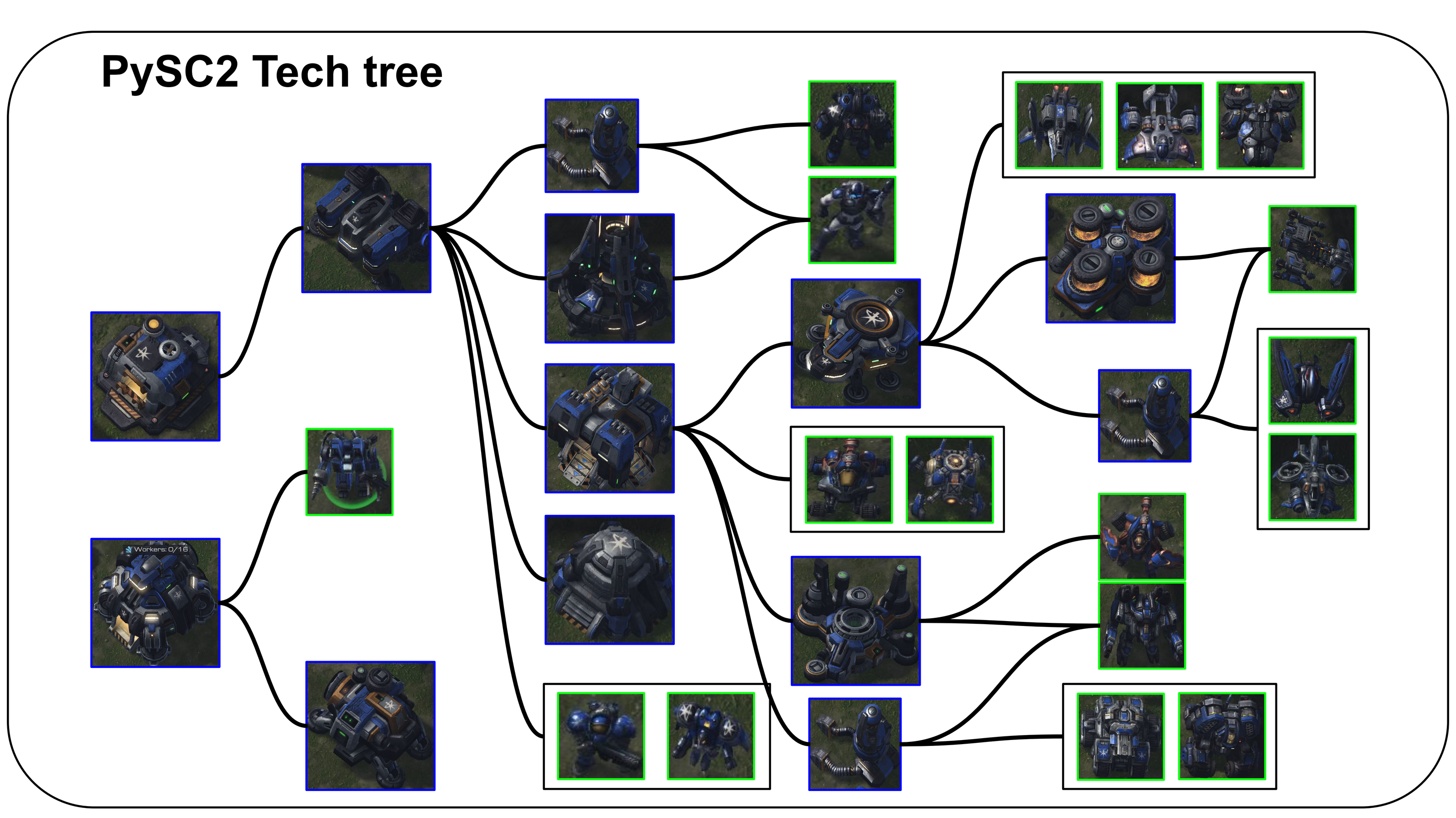
$$\widehat{G}^{\text{MLE}} = (\widehat{G}_{\mathbf{c}}^{\text{MLE}}, \widehat{G}_{\mathbf{r}}^{\text{MLE}}) = \arg\max_{G_{\mathbf{c}}, G_{\mathbf{r}}} p(\tau | G_{\mathbf{c}}, G_{\mathbf{r}})$$

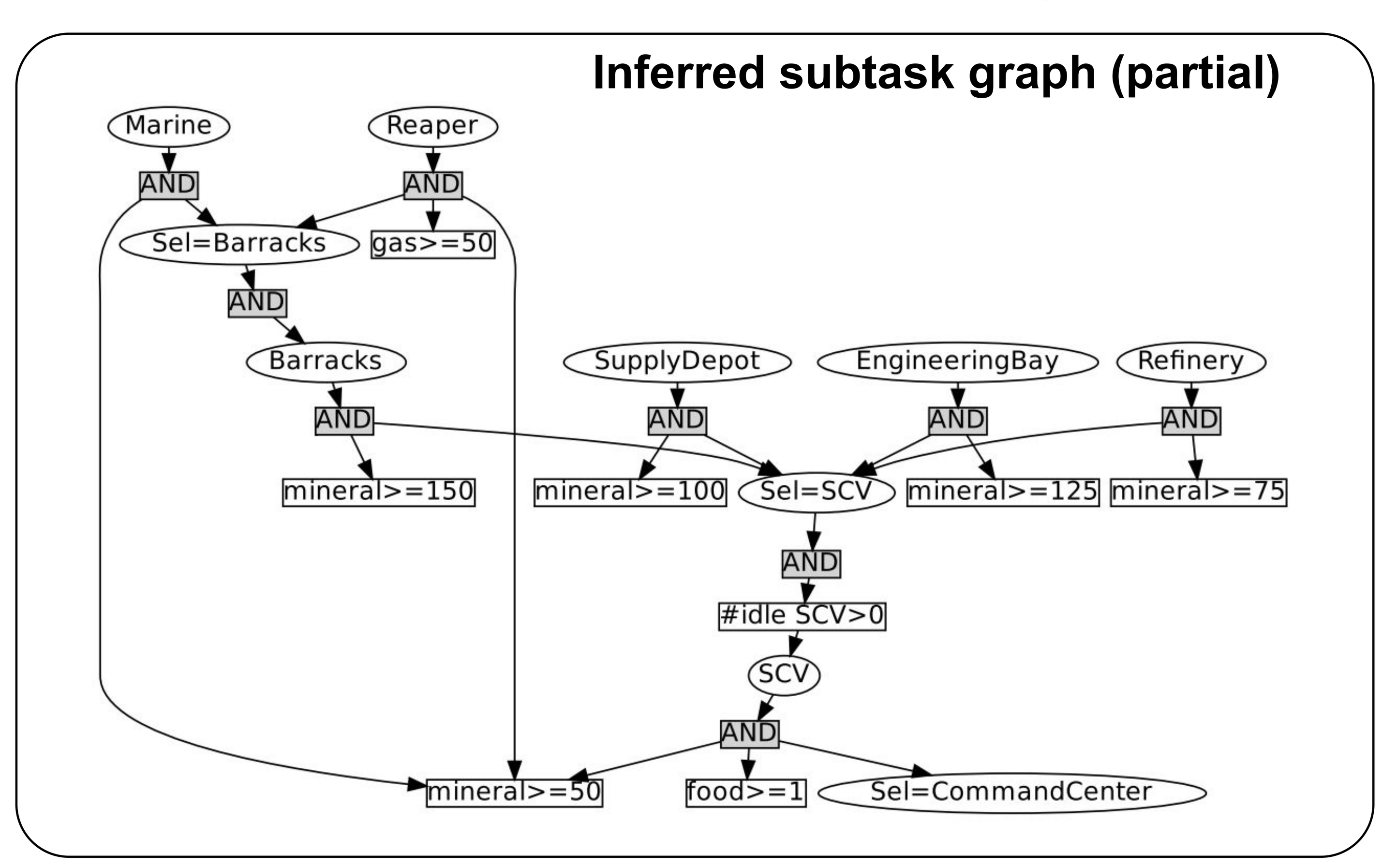
  - **Precondition** ($G_{\mathbf{c}}$): CART-based logic induction algorithm to infer the precondition function ($\mathbf{e} = \widetilde{f}_{G_{\mathbf{c}}}(\mathbf{x})$)
  - **Subtask reward** ($G_{\mathbf{r}}$): $G_{\mathbf{r}}^i \sim \mathcal{N}(\widehat{\mu}^i, \widehat{\sigma}^i)$, $\widehat{G}_{\mathbf{r}}^{\text{MLE},i} = \widehat{\mu}_{\text{MLE}}^i = \mathbb{E}\left[r_t \mid \mathbf{o}_t = \mathcal{O}^i\right]$

Experiment

- Setting
  - Domain: Playground [2], Mining [2], SC2LE [3]
  - Agents: Random, HRL [4], RL^2 [5], MSGI-random (ours), MSGI-meta (ours)
- Result on Playground and Mining domain
  - Adaptation: our learned adaptation policy enables faster adaptation (MSGI-meta V.S. MSGI-random)
  - Generalization: MSGI (ours) consistently outperforms baselines on unseen tasks with **longer** (**unseen**) adaptation horizon
- Result on StarCraft II (SC2LE)
  - Subtask graph inference: given 20 episodes budget with 2.5K environment steps, our method correctly infers **93%** of the preconditions







[1] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation.arXiv preprint arXiv,2015.
[2] Sungryull Sohn, Junhyuk Oh, and Honglak Lee. Hierarchical reinforcement learning for zero-shot generalization with subtask dependencies. InNeurIPS, pp. 7156–7166, 2018
[3] Vinyals, Oriol, et al. Starcraft ii: A new challenge for reinforcement learning. arXiv, 2017.
[4] Jacob Andreas et al. Modular multitask reinforcement learning with policy sketches, ICML, 2017
[5] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rlˆ2: Fast reinforcement learning via slow reinforcement learning. arXiv, 2016.