Discovering Hierarchical Achievements in Reinforcement Learning via Contrastive Learning

Seungyong Moon [1] Junyoung Yeom [1] Bumsoo Park [2] Hyun Oh Song [1]

[1]Seoul National University [2]KRAFTON

Introduction

We propose a new **self-supervised** method in conjunction with **model-free** RL algorithms that improves the agent's capacity for

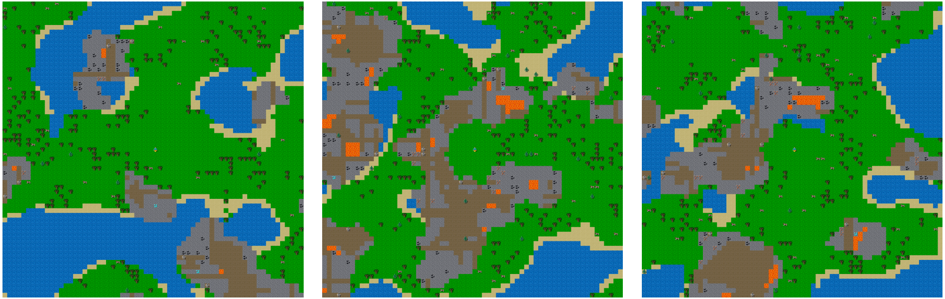
1. **generalization** across visually diverse environments
2. **discovery of various skills** with complex hierarchies.

Crafter Benchmark[1]

- Crafter is a 2D open-world game inspired by Minecraft.
- It is optimized for research purposes with
  - fast environment interactions
  - clear evaluation metrics.

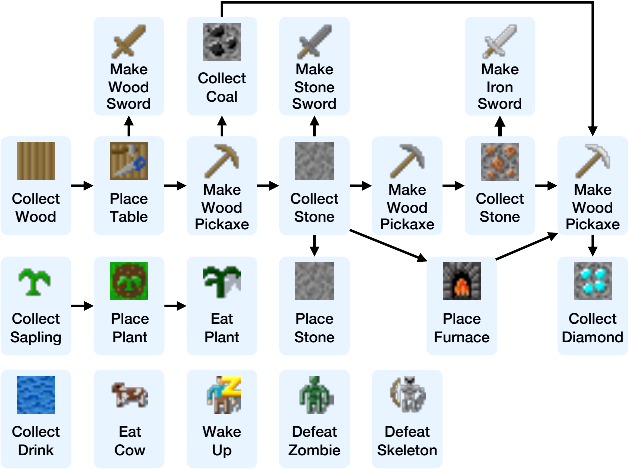
[1] D. Hafner, Benchmarking the Spectrum of Agent Capabilities, ICLR 2022.

Crafter Benchmark



An agent navigates a previously unseen, procedurally generated environment with varying maps.

Hierarchical Achievements



Make Wood Sword

Collect Coal

Make Stone Sword

Make Iron Sword

Collect Wood

Place Table

Make Wood Pickaxe

Collect Stone

Make Wood Pickaxe

Collect Stone

Make Wood Pickaxe

Collect Sapling

Place Plant

Eat Plant

Place Stone

Place Furnace

Collect Diamond

Collect Drink

Eat Cow

Wake Up

Defeat Zombie

Defeat Skeleton

An agent must discover a set of 22 hierarchical skills, called **achievements**.

Achievement Reward

- An agent receives a sparse reward of 1 upon unlocking a new achievement within an episode.

- **Note**: rewards are given only for the first accomplishment.

Objective

- An agent has no prior knowledge of the achievements and must infer it indirectly from the reward signal.

- The objective is to find the optimal policy that unlocks **as many achievements as possible**.

Prior Work

- Prior work has mainly relied on **model-based** and **hierarchical methods**, which employ **explicit modules** for long-term planning.

- However, they often suffers from sample and computational inefficiencies.

Motivation

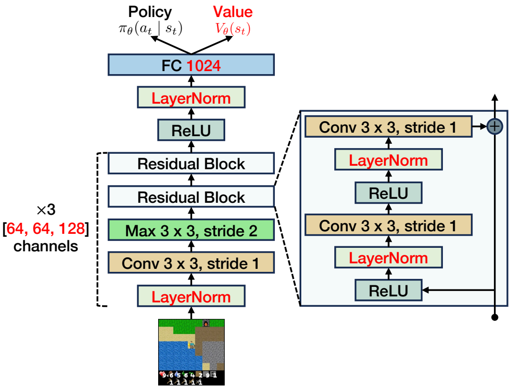
- The model-free approach has been less explored, despite its simplicity and versatility.

- In this work, we aim to explore **the capability of PPO** in discovering hierarchical achievements.

Model-Free PPO is a Strong Baselines

- We adopt recent implementation practices of PPO.
  - Network size
  - Layer normalization
  - Value normalization

- These modifications significantly improve the performance of PPO.



Achievement Distillation

- We propose a new self-supervised method, **achievement distillation**, to provide guidance to the encoder for predicting the next achievement **in the latent space**.

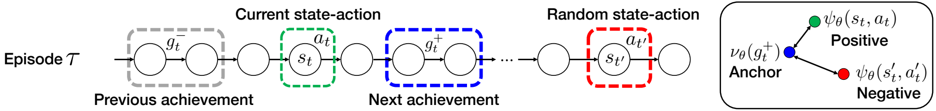
- It leverages **the temporal information** on when new achievements are unlocked, which is directly identifiable from the reward signal.

Notation

- For each episode τ, we define
  - $(t_i)_{i=1}^m$: the sequence of timesteps at which achievements are unlocked
  - $(g_i)_{i=1}^m$: the sequence of unlocked achievements, each defined by a transition tuple $g_i = (s_{t_i}, a_{t_i}, s_{t_i+1})$.

- For each timestep t, we define
  - g_t^+: the next immediate achievement
  - g_t^-: the previous immediate achievement.

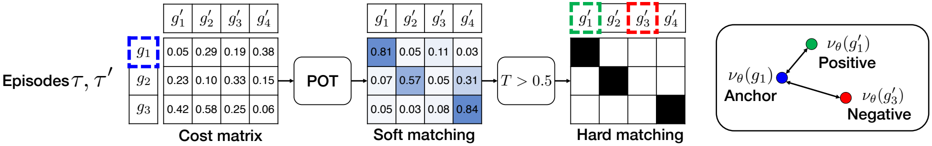
Intra-Trajectory Achievement Prediction



- Given an episode τ, it trains the encoder θ to maximize the similarity in the latent space between
  - a given state-action pair (s_t, a_t) (**positive**)
  - its next achievement g_t^+ (**anchor**).

- We sample a random state-action pair $(s_{t'}, a_{t'})$ (**negative**) and employ contrastive learning.

Cross-Trajectory Achievement Matching



	g'_1	g'_2	g'_3	g'_4
g_1	0.05	0.29	0.19	0.38
g_2	0.23	0.10	0.33	0.15
g_3	0.42	0.58	0.25	0.06

Cost matrix

g'_1	g'_2	g'_3	g'_4
0.81	0.05	0.11	0.03
0.07	0.57	0.05	0.31
0.05	0.03	0.08	0.84

Soft matching

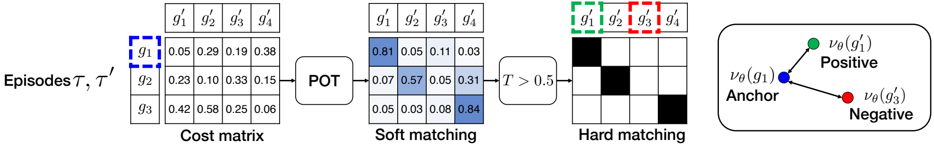
g'_1	g'_2	g'_3	g'_4
■			
	■		
			■

Hard matching

- Given episodes τ, τ', it trains the encoder θ to maximize the similarity in the latent space between the identical achievements
  - $g_i \in \tau$ (anchor)
  - $g'_k \in \tau'$ (positive).

- Since the achievement labels are unavailable, we compute the matching between the achievement sequences using **partial optimal transport**, followed by thresholding.

Cross-Trajectory Achievement Matching



	g_1'	g_2'	g_3'	g_4'
g_1	0.05	0.29	0.19	0.38
g_2	0.23	0.10	0.33	0.15
g_3	0.42	0.58	0.25	0.06

Cost matrix

g_1'	g_2'	g_3'	g_4'
0.81	0.05	0.11	0.03
0.07	0.57	0.05	0.31
0.05	0.03	0.08	0.84

Soft matching

g_1'	g_2'	g_3'	g_4'
■	□	□	□
□	■	□	□
□	□	□	■

Hard matching

- Then, it trains the encoder to maximize the similarity in the latent space for matched achievements.

- We sample a random achievement $g_j' \in \tau$ (**negative**) and employ contrastive learning.

Integration with PPO

- We integrate achievement distillation with PPO by introducing **an auxiliary phase**.
- During the auxiliary phase, we perform achievement distillation with off-policy data collected during multiple PPO updates.
- We **jointly regularize** the policy and value networks to maintain their outputs.

Experiments

- We train the agent on Crafter for 1M environment steps and measure the score and reward.
  - Score: the geometric mean of success rates for individual achievements during training
  - Reward: the number of unlocked achievements

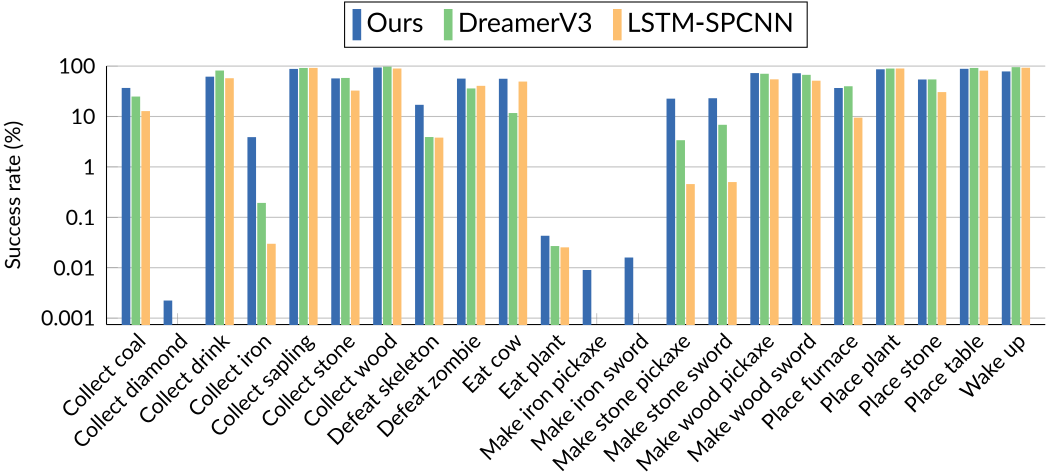
- We compare our method with four baseline methods.
  - Model-based: MuZero, DreamerV3
  - Hierarchical: SEA
  - Model-free: LSTM-SPCNN

Results

Method	Parameters	Score (%)	Reward
Human Expert	-	50.5 ± 6.8	14.3 ± 2.3
Ours	9M	**21.79** ± **1.37**	**12.60** ± **0.31**
PPO	4M	15.60 ± 1.66	10.32 ± 0.53
DreamerV3	201M	14.77 ± 1.42	10.92 ± 0.53
LSTM-SPCNN	135M	11.67 ± 0.80	9.34 ± 0.23
MuZero	54M	4.4 ± 0.4	8.5 ± 0.1
SEA	1.5M	1.22 ± 0.13	0.63 ± 0.08

Our method outperforms all the baselines in both metrics.

Results



Our method is the only one capable of discovering all 22 achievements.

Conclusion

Model-free algorithms can possess the strong capability to discover hierarchical achievements via **self-supervised learning**.

- Paper: https://arxiv.org/abs/2307.03486
- GitHub: https://github.com/snu-mllab/Achievement-Distillation