

Focused Skill Discovery

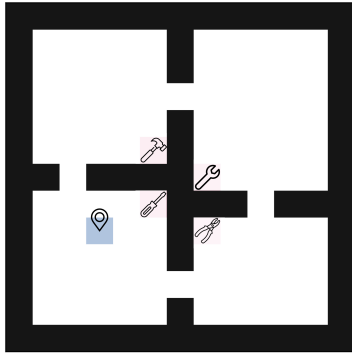
Using Per-Factor Empowerment to Control State Variables

Jonathan Colaço Carr, Qinyi Sun, Cameron Allen



Accepted to RLC 2025!

1 When states are **factored**...



$s = (\text{hammer}, \text{wrench}, \text{screwdriver}, \text{pliers}, \text{location})$

2 Skills should **focus** on specific state variables

Focused skills target and control specific state variables

E.g.:

Skill 1: Only pick up

Skill 2: Only pick up

... etc.

Benefits of focused skills:

1. Better exploration
2. Safe by design
3. Faster learning in downstream tasks

3

We can learn focused skills with a simple tweak to existing algorithms

4 Idea (two parts):

i. Apply a baseline skill discovery algorithm to each state variable

This creates skills that control individual state variables

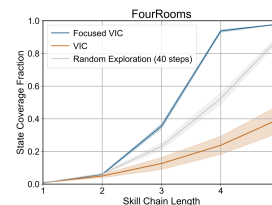
ii. Penalize changes to non-target variables

This ensures that skills only change their target variables

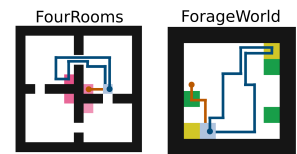
5

... leads to better skills

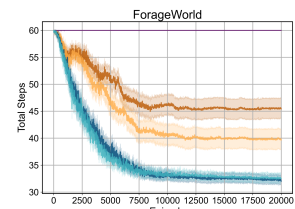
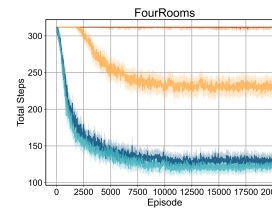
Exploration:



Side effects:



Learning in downstream tasks:



Details

Algorithm 1 Focused Variational Intrinsic Control

```

for episode = 1, ..., M do
  Sample  $s_0$  from the initial state distribution  $\mu$ 
  Sample skill  $z$  from  $\nu(\cdot|s_0)$ 
  Follow policy  $\pi_z$  until termination state  $s_T$ 
  for each target variable  $i$  of  $z$  do
    Update the skill discriminator  $d_i$  from  $(z, s_T^i, s_0^i)$ 
  end for
  Calculate reward  $r = \sum_i \log(d_i(z | s_T^i, s_0^i)) - \log(\nu(z|s_0)) - \|s_T - s_0\|_{\lambda,2}$ 
  Update  $\pi_z$  to maximize  $r$ 
  Reinforce option prior  $\nu(\cdot|s_0)$  based on  $r$ 
end for
  
```

Original

```

  Update the skill discriminator  $d$  from  $(z, s_T, s_0)$ 
  Calculate reward  $r = \log(d(z | s_T, s_0)) - \log(\nu(z|s_0))$ 
  
```