MDP Abstraction with Successor Features

Dongge Han[1], Michael Wooldridge[1], Sebastian Tschiatschek[2]

[1] Department of Computer Science, University of Oxford, [2] Department of Computer Science, University of Vienna

Abstract

Abstraction plays an important role for generalisation of knowledge and skills, and is key to sample efficient learning. We study joint temporal and state abstraction in reinforcement learning, where temporally-extended actions in the form of options induce temporal abstractions, while aggregation of similar states with respect to abstract options induce state abstractions. We propose a novel abstraction scheme building on successor features. This includes an algorithm for transferring abstract options across different environments, and a state abstraction mechanism which allows us to perform efficient planning with the transferred options.

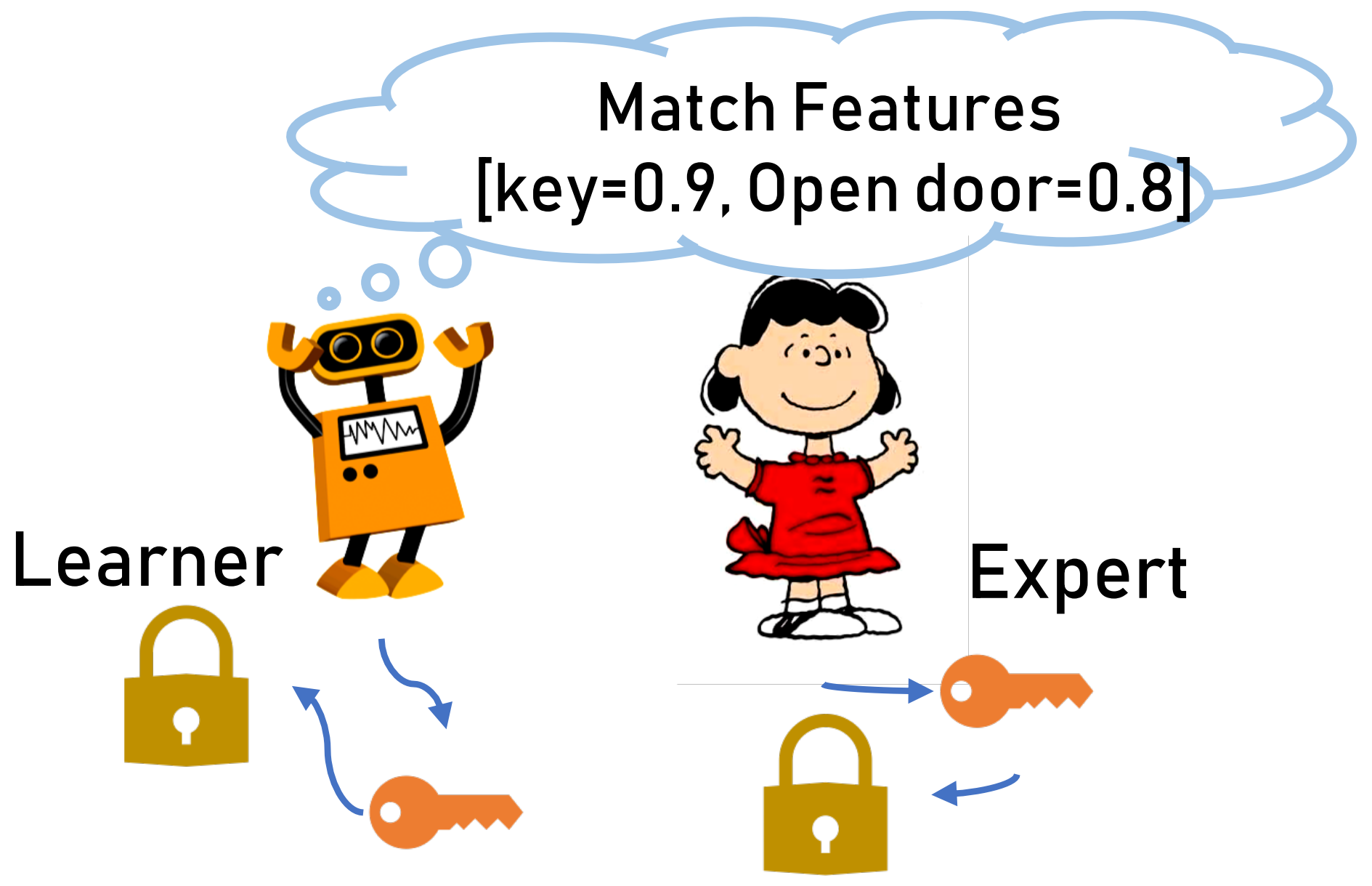
Options as Successor Features

How can we transfer learned options to new environments and reuse them for efficient planning and exploration? To enable option transfer, we provide an abstract representation $\bar{o}$ of options o by their successor features – cumulative discounted features $\theta(s,a)$ by executing the option.

$$o \mapsto \psi_s^o = \mathbb{E}[\sum_{\kappa=0}^{k} \gamma^\kappa \theta(S_{t+\kappa}, A_{t+\kappa}) | \mathscr{E}(o,s,t)]$$

$o \in g_s^{-1}(\bar{o})$ are *ground options* of $\bar{o}$ staring from state s.

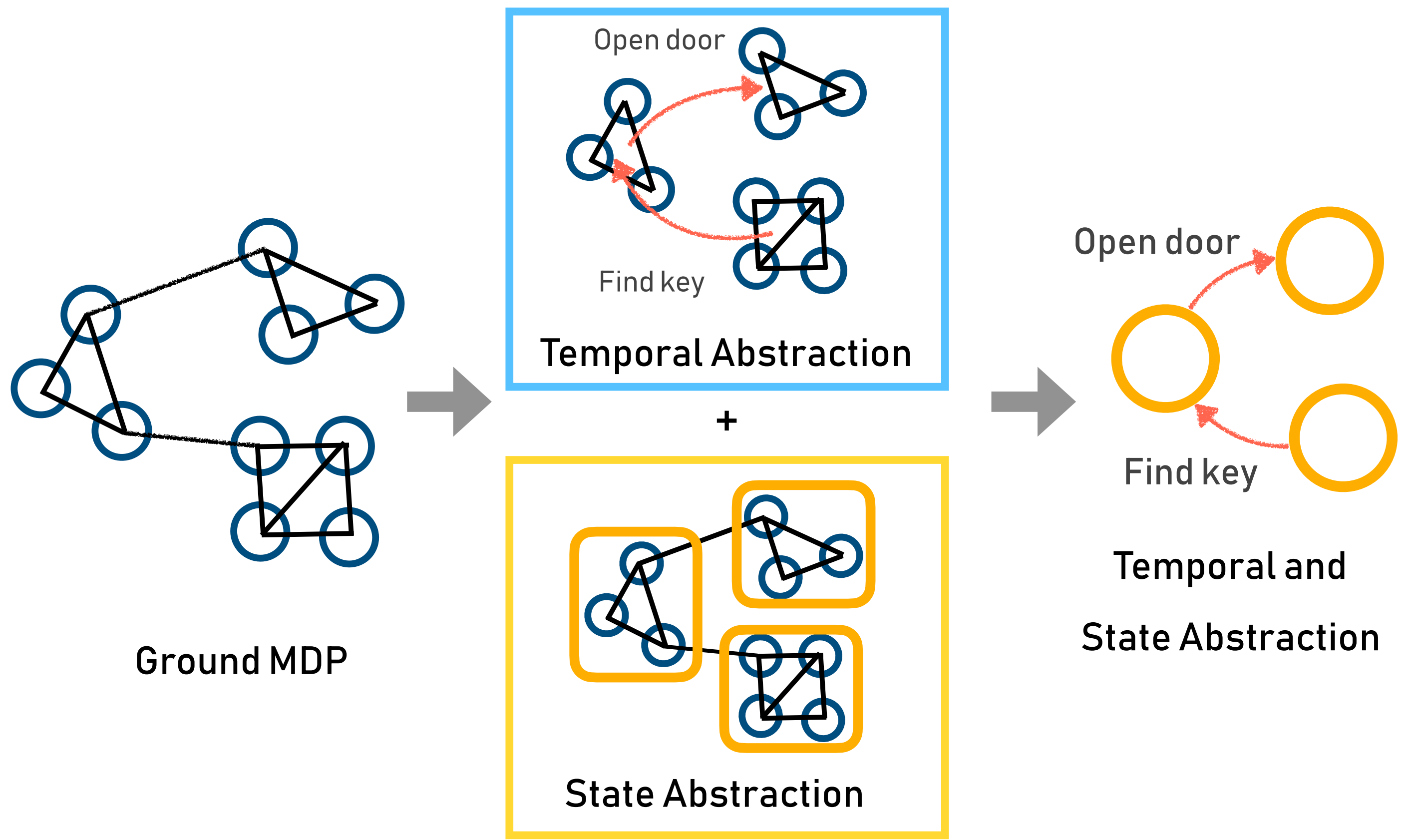
Option Grounding via Inverse RL



To find the ground option policy of an abstract option, we present two algorithms for option grounding based on Inverse Reinforcement Learning (IRL): IRL-NAIVE and IRL-BATCH. As shown in the figure, IRL is often modeled as a feature matching problem which finds a policy that realises the cumulative features of an expert.

State Abstraction with Successor Homomorphism

Integrate temporal abstraction (i.e., options) and state abstraction via Successor Homomorphism. The induced ψ-SMDP can be used for efficient planning.



ε-Approximate Successor Homomorphism The homomorphism $h=(f,g)$ produces an abstract ψ-SMDP from the ground MDP. The state abstraction function f aggregate similar ground states to abstract states $\bar{s}$ where the abstract options executed from these states have similar transition dynamics and successor features.

$$h(s_1,o_1)=h(s_2,o_2) \implies \forall \bar{s}', | \sum_{s' \in f^{-1}(\bar{s}')} P^{o_1}_{s_1,s'} - P^{o_2}_{s_2,s'}| \leq \varepsilon_P, \quad \|\psi^{o_1}_{s_1} - \psi^{o_2}_{s_2}\|_1 \leq \varepsilon_\psi.$$

With the induced abstract states, the abstract transition dynamics and the ψ-SMDP can be computed. Given any reward vector on the features w_r, planning with the abstract ψ-SMDP yields near optimal performance to the optimal ground SMDP policy, the difference bounded by $\frac{2\kappa}{(1-\gamma)^2}$, where $\kappa = |w_r|(2\varepsilon_\psi + \frac{\varepsilon_P|\bar{S}|\max_{s,a}|\theta(s,a)|}{1-\gamma})$.

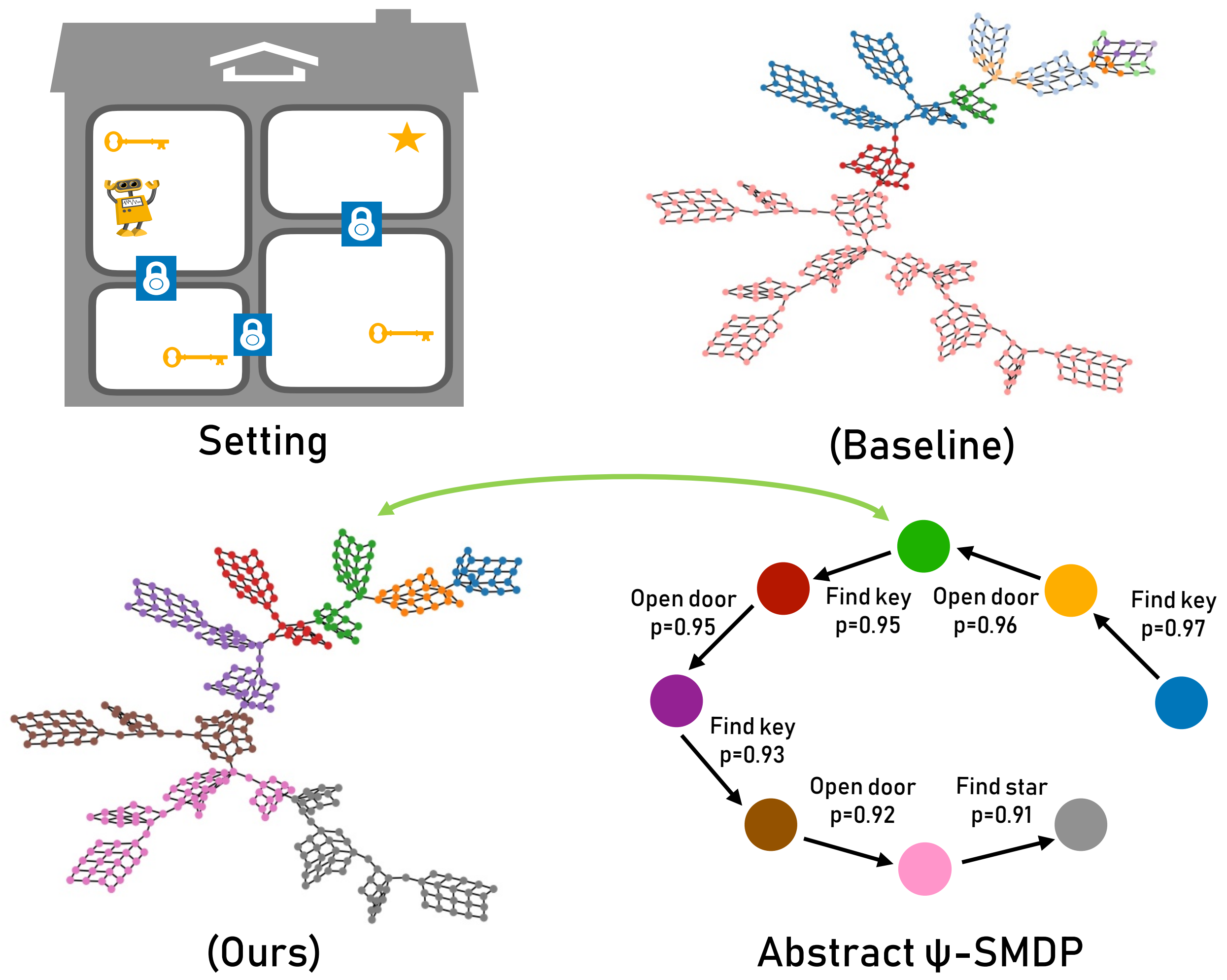
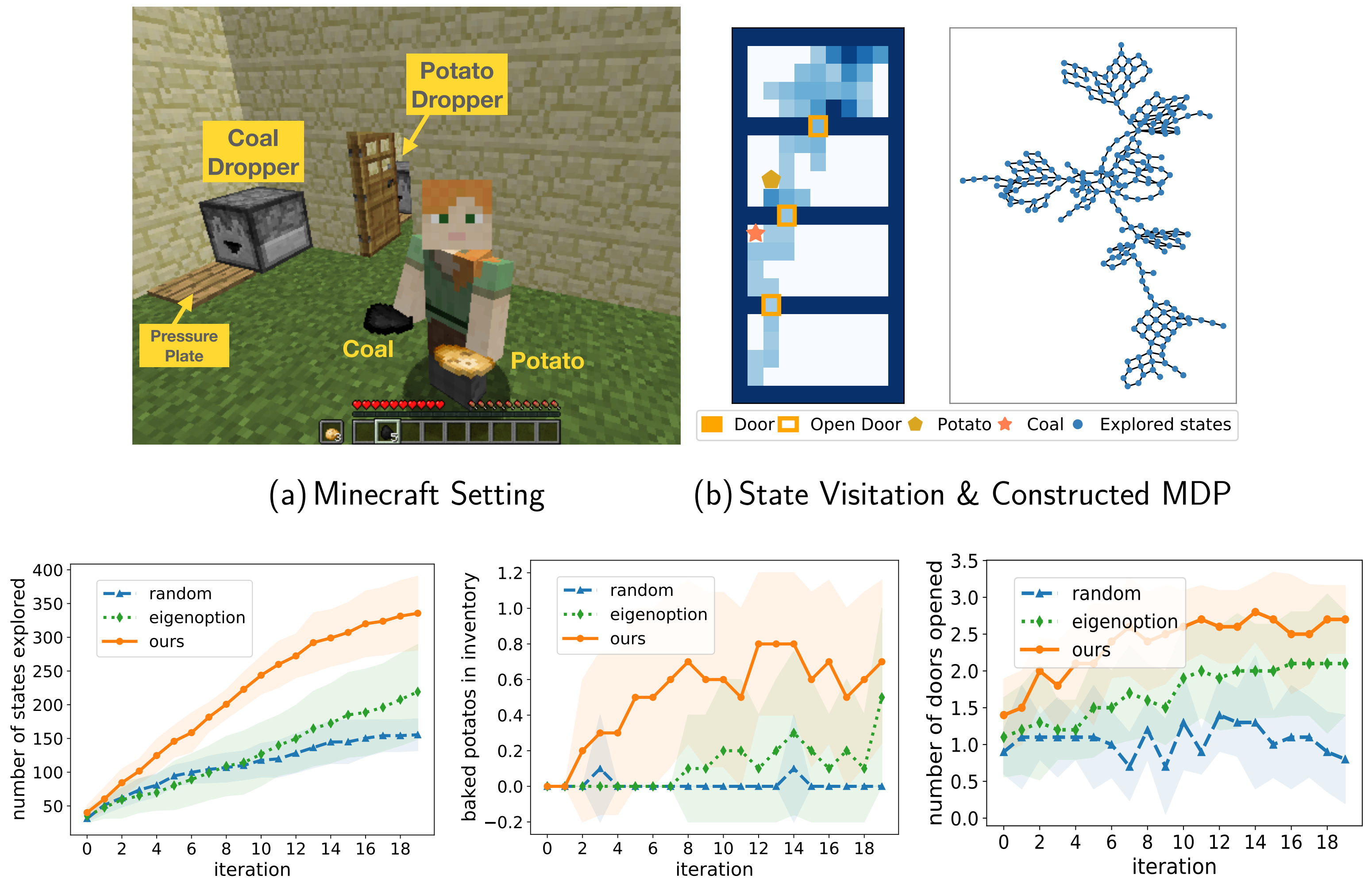


Figure 1:Abstract ψ-SMDP generated by our successor homomorphism and the mapping of the ground states to abstract states (Colours show the ground states that map to the same abstract state).

Experiments

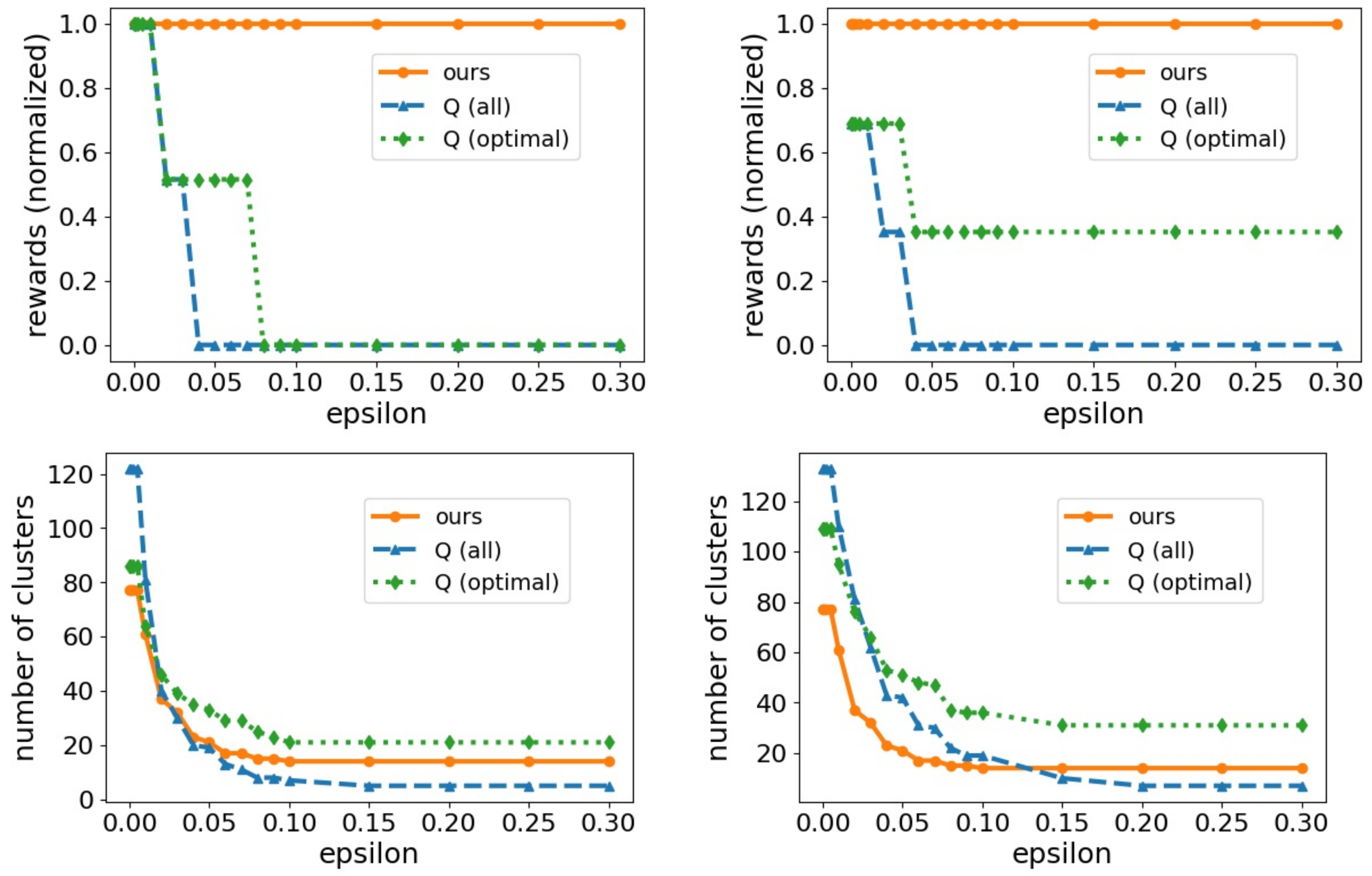
Option Grounding with Exploration Given demonstrations of the options, the Minecraft agent learns to open doors and navigate the rooms to obtain potato, coal and bake the potatos, through iterative option grounding and exploration (with the grounded options) in an unknown environment, and construct the MDP in only a few iterations.



(a) Minecraft Setting (b) State Visitation & Constructed MDP

(c) Number of States Explored (d) Baked Potatos Collected (e) Doors Opened

Performance of Planning with the Abstract ψ-SMDP's The upper row shows the total rewards and the lower row shows the corresponding number of abstract states, w.r.t., thresholds ε. Left/Right columns show (w.o./with) task transfer i.e., different reward function.



Contact Information

- Email Address: dongge.han@cs.ox.ac.uk