SPOTTER: Supplementary Material

December 17, 2020

Contents

1 Extended Experiments with Baselines

Below are figures showing experimental results for our baseline algorithms over 2,000,000 episodes.

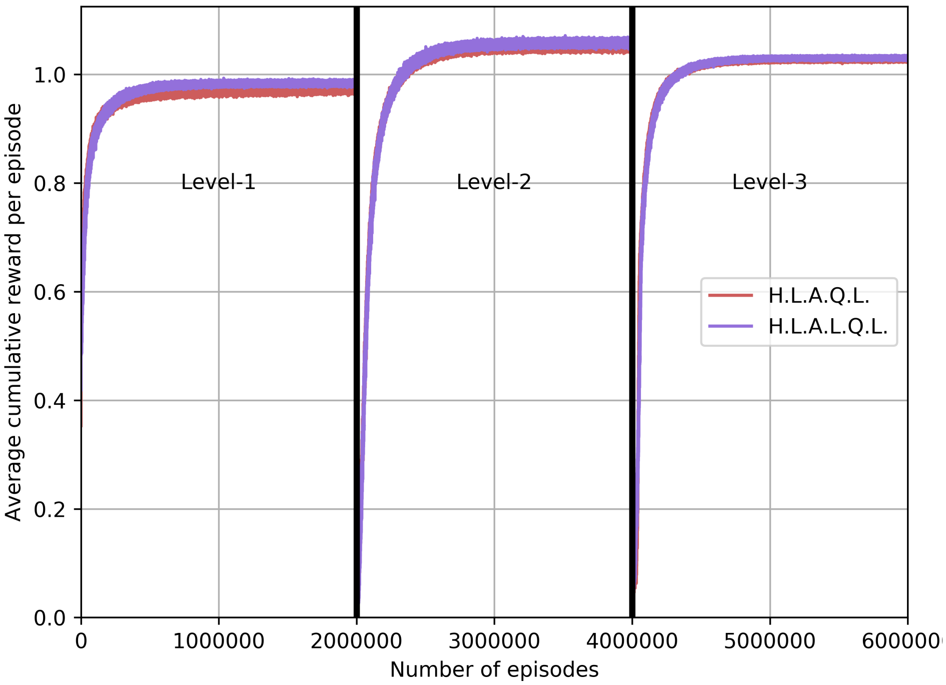


Figure 1: Baselines up to 2,000,000 episodes normalized to maximum reward obtained by SPOTTER. In the long run, the baselines perform better than SPOTTER, however, they take much longer and do not learn transferable knowledge as effectively as SPOTTER.

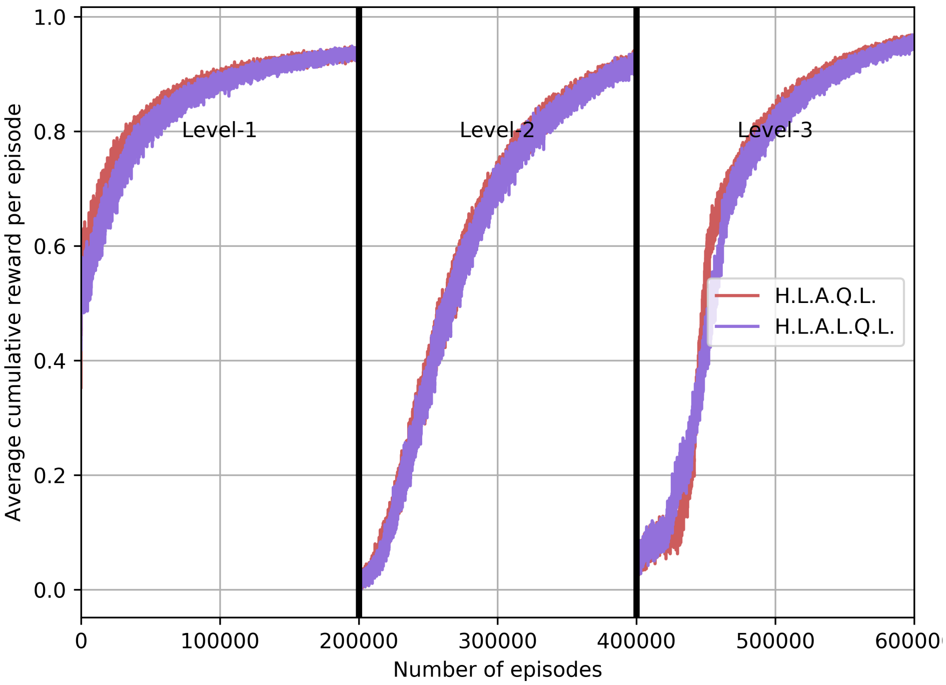


Figure 2: Baselines up to 200,000 episodes normalized to maximum reward obtained by SPOTTER.

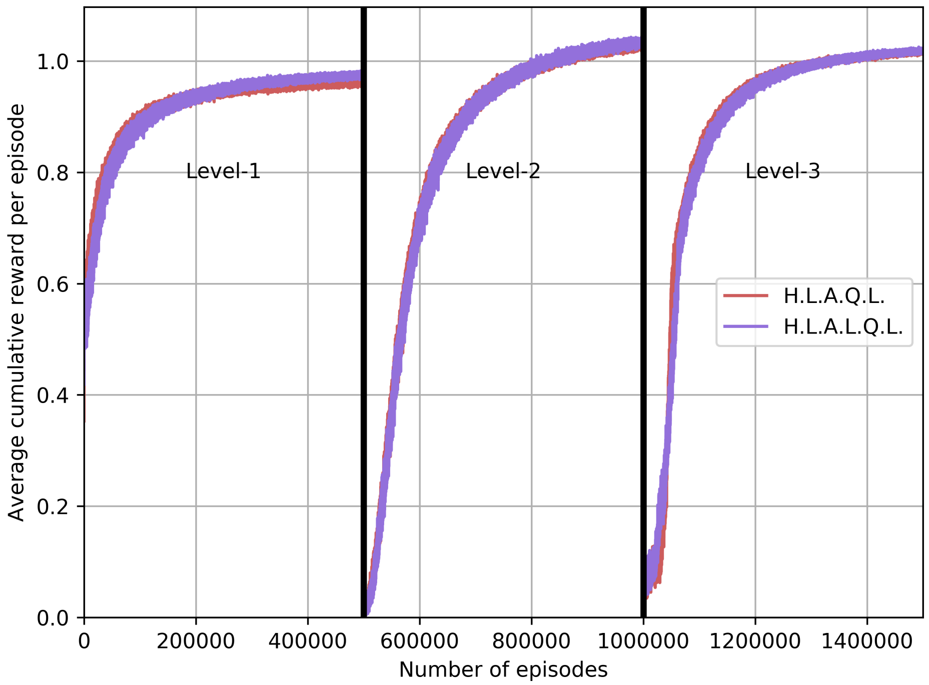
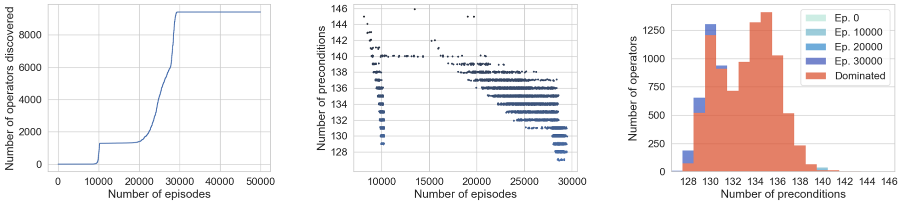


Figure 3: Baselines up to 500,000 episodes normalized to maximum reward obtained by SPOTTER.

2 Preliminary experiment: generalizing preconditions

Ordinarily, as soon as SPOTTER discovers an operator exceeding the value threshold, that operator is incorporated into the planning agent's model. We dispensed with this assumption and ran SPOTTER on puzzle 2 for 50,000 episodes, allowing the system to continue learning operator policies and generating preconditions throughout this time. Every 50 episodes, SPOTTER logged the new operators it had created. Figure 4 shows the results for one particular operator learner (i.e., each of the output operators has the same postconditions and the same underlying policy, but has a unique set of preconditions). As Figure 4a indicates, by episode 29,500 this learner discovered 9,049 unique operators; no further operators were discovered after this episode.

Figures 4b and 4c demonstrate that with additional exploration, the agent can construct more general operators. Recall that an operator with a set of preconditions is accepted whenever the average value of all MDP states satisfying those preconditions is greater than the value threshold (in this experiment, 0.9). As the values of additional MDP states increase past the threshold, more gen-



(a) The number of unique sets of preconditions discovered for which the average value is above threshold increases as SPOTTER is allowed to continue exploring.

(b) Generation of operators with decreasing numbers of preconditions proceeds in "waves" as the value estimates of additional MDP states converge.

(c) By episode 30,000, almost all operators have been dominated (replaced by superior operators), and almost all non-dominated operators were created late in the run.

Figure 4: Results of precondition generalization experiment for a single subgoal learner on puzzle 2.

eral operators (with fewer preconditions) cross this threshold. Figure 4b plots, for each operator discovered, the episode in which it was logged and its total number of preconditions. Note that there are several "waves" of precondition generalization. Beginning with the discovery of the first viable set of preconditions, as additional states cross the threshold there is a rapid discovery of new operators with additional preconditions. Eventually (a little after 10,000 episodes), the existing operators are sufficiently general that many rarely-seen MDP states would have to be thoroughly explored before more general preconditions can be found. For a while, any new operators discovered (while they are not merely specifications of existing operators) have a larger number of preconditions. This culminates in a "second wave" in which enough MDP states have been explored that more general operators can be produced, ultimately surpassing the first wave.

Figure 4c shows that operators created later not only have fewer preconditions than earlier operators, but *dominate* earlier operators (in that the preconditions of the dominated operator are a strict superset of the preconditions of the dominating operator). Orange bars represent dominated operators, blue bars those for which a superior operator has not yet been found. Relatively few non-dominated operators persist by step 30,000, and nearly all that do were created in the last few thousand episodes, suggesting that running SPOTTER for longer before incorporating operators allows the construction of strictly better (more general) operators.[1]

[1] An animated version of this chart, showing how operators are constructed and then dominated as the agent continues exploring, appears in the supplementary material.

3 Videos

Attached in the supplementary materials package are several videos, showing specific runs of SPOTTER through the environment. Beginning with level 1, which SPOTTER can solve entirely by planning (i.e., no need for learning), then proceeding to learning in Level 2. In Level 2, SPOTTER at first needs to explore (episode 0), then when it hits a plannable state, can transition to planning the rest of the way to the goal, and then when it finds the operator, it can complete Level 2 entirely without any RL and only with planning. One promising aspect of SPOTTER is that in Level 3, we show that it can simply use the learned operator in a plan, without instantiating a new learning session, even though the goals have changed substantially.

- SPOTTER Level 1 (planning)
- SPOTTER Level 2 (episode 0)
- SPOTTER Level 2 (hit plannable state)
- SPOTTER Level 2 (found operator)
- SPOTTER Level 3 (plan with learned operator

We also provide a video for precondition generalization in which we show, over time, how the distribution of preconditions evolves. The orange bars show that more general preconditions have superceded less general ones (blue).

4 Open World PDDL

Below is the original PDDL domain given to SPOTTER for the 2-room-blocked minigrid environment.

```
(define (domain gridworld_abstract)

  (:requirements :strips :typing )

  (:types thing room - object
          agent physobject - thing
          door nondoor - physobject
          notgraspable graspable - nondoor
          wall floor goal lava - notgraspable
          key ball box - graspable)

  (:predicates
           (nexttofacing ?a - agent ?thing - physobject)
           (open ?t - door)
           (closed ?t - door)
           (locked ?t - door)
           (holding ?a - agent ?t - graspable)
           (handsfree ?a - agent)
           (obstructed ?a - agent)
           (blocked ?t - door)
```

```
         (atgoal ?a - agent ?g - goal)
         (inroom ?a - agent ?g - physobject)
         )

(:action pickup
 :parameters (?a - agent ?thing - graspable)
 :precondition (and (handsfree ?a) (nexttofacing ?a ?thing) (not (holding ?a ?thing)))
 :unknown (and)
 :effect (and (not (handsfree ?a))
              (not (nexttofacing ?a ?thing))
              (not (obstructed ?a))
              (holding ?a ?thing))
)

(:action putdown
 :parameters (?a - agent ?thing - graspable)
 :precondition (and (not (obstructed ?a)) (holding ?a ?thing))
 :unknown (and (blocked *))
 :effect (and (not (holding ?a ?thing))
              (handsfree ?a)
              (nexttofacing ?a ?thing)
              (obstructed ?a)
              ))

(:action gotoobj1
 :parameters (?a - agent ?thing - graspable)
 :precondition (and (not (holding ?a ?thing)) (not (obstructed ?a)) (inroom ?a ?thing))
 :unknown (and)
 :effect (and (obstructed ?a)
              (nexttofacing ?a ?thing)))

(:action gotoobj2
 :parameters (?a - agent ?thing - notgraspable)
 :precondition (and (not (obstructed ?a)) (inroom ?a ?thing))
 :unknown (and)
 :effect (and (obstructed ?a)
              (nexttofacing ?a ?thing)))

(:action gotoobj3
 :parameters (?a - agent ?thing - graspable ?obstruction - thing)
 :precondition (and (not (holding ?a ?thing)) (nexttofacing ?a ?obstruction)
 (inroom ?a ?thing))
 :unknown (and)
 :effect (and (nexttofacing ?a ?thing)
              (not (nexttofacing ?a ?obstruction))))

(:action gotoobj4
 :parameters (?a - agent ?thing - notgraspable ?obstruction - thing)
 :precondition (and (nexttofacing ?a ?obstruction) (inroom ?a ?thing))
 :unknown (and)
 :effect (and (nexttofacing ?a ?thing)
              (not (nexttofacing ?a ?obstruction))))

(:action usekey
 :parameters (?a - agent ?key - key ?door - door)
 :precondition (and (nexttofacing ?a ?door) (holding ?a ?key) (locked ?door))
 :unknown (and)
```

```
   :effect (and (open ?door) (not (closed ?door)) (not (locked ?door))))

  (:action opendoor
   :parameters (?a - agent ?door - door)
   :precondition (and (nexttofacing ?a ?door) (closed ?door))
   :unknown (and)
   :effect (and (open ?door) (not (closed ?door)) (not (locked ?door))))

  (:action stepinto
   :parameters (?a - agent ?g - goal)
   :precondition (and (nexttofacing ?a ?g))
   :unknown (and (nexttofacing ?a *))
   :effect (and (not (nexttofacing ?a ?g)) (atgoal ?a ?g)))

  (:action gotodoor1
   :parameters (?a - agent ?thing - door)
   :precondition (and (not (obstructed ?a)) (not (blocked ?thing)) (inroom ?a ?thing))
   :unknown (and)
   :effect (and (obstructed ?a)
                (nexttofacing ?a ?thing)))

  (:action gotodoor2
   :parameters (?a - agent ?thing - door ?obstruction - thing)
   :precondition (and (nexttofacing ?a ?obstruction) (not (blocked ?thing))
   (inroom ?a ?thing))
   :unknown (and)
   :effect (and (nexttofacing ?a ?thing)
                (not (nexttofacing ?a ?obstruction))))

  (:action enterroomof
   :parameters (?a - agent ?d - door ?g - physobject)
   :precondition (and (not (blocked ?d)) (nexttofacing ?a ?d) (open ?d))
   :unknown (and)
   :effect (and (inroom ?a ?g) (not (nexttofacing ?a ?d)) (not (obstructed ?a))))

)
```

5 Discovered Operator

Below is an example of an operator discovered by SPOTTER.

```
+PR: (inroom agent obj0013)
  +PR: (inroom agent key)
  +PR: (inroom agent obj0026)
  +PR: (inroom agent obj0005)
  +PR: (inroom agent obj0011)
  +PR: (inroom agent obj0028)
  +PR: (locked door)
  +PR: (inroom agent obj0009)
  +PR: (inroom agent obj0023)
  +PR: (inroom agent obj0001)
  +PR: (inroom agent obj0007)
  +PR: (inroom agent obj0002)
```

```
+PR: (inroom agent obj0020)
+PR: (holding agent key)
+PR: (inroom agent obj0025)
+PR: (inroom agent obj0016)
+PR: (inroom agent obj0006)
+PR: (nexttofacing agent obj0015)
+PR: (inroom agent obj0012)
+PR: (inroom agent obj0017)
+PR: (inroom agent obj0027)
+PR: (inroom agent obj0003)
+PR: (inroom agent obj0000)
+PR: (inroom agent obj0022)
+PR: (inroom agent obj0015)
+PR: (inroom agent obj0031)
+PR: (inroom agent obj0008)
+PR: (inroom agent obj0019)
+PR: (inroom agent obj0004)
+PR: (inroom agent obj0029)
+PR: (inroom agent obj0021)
+PR: (inroom agent ball)
+PR: (inroom agent obj0030)
+PR: (inroom agent obj0014)
+PR: (inroom agent obj0024)
+PR: (inroom agent door)
+PR: (inroom agent obj0018)
+PR: (inroom agent obj0010)
+PR: (obstructed agent)
–PR: (inroom agent obj0038)
–PR: (inroom agent obj0058)
–PR: (inroom agent obj0062)
–PR: (nexttofacing agent obj0004)
–PR: (nexttofacing agent obj0023)
–PR: (nexttofacing agent obj0032)
–PR: (nexttofacing agent obj0014)
–PR: (inroom agent obj0047)
–PR: (nexttofacing agent obj0029)
–PR: (nexttofacing agent obj0046)
–PR: (nexttofacing agent obj0016)
–PR: (inroom agent obj0059)
–PR: (inroom agent obj0042)
–PR: (nexttofacing agent obj0052)
–PR: (nexttofacing agent obj0049)
–PR: (nexttofacing agent obj0006)
–PR: (nexttofacing agent obj0033)
–PR: (nexttofacing agent obj0003)
–PR: (inroom agent obj0054)
```

```
–PR: (inroom agent obj0048)
–PR: (inroom agent obj0055)
–PR: (nexttofacing agent obj0027)
–PR: (nexttofacing agent obj0063)
–PR: (inroom agent obj0057)
–PR: (nexttofacing agent obj0013)
–PR: (nexttofacing agent obj0048)
–PR: (inroom agent obj0053)
–PR: (nexttofacing agent goal)
–PR: (holding agent ball)
–PR: (nexttofacing agent obj0012)
–PR: (inroom agent obj0033)
–PR: (inroom agent obj0040)
–PR: (nexttofacing agent obj0051)
–PR: (inroom agent obj0045)
–PR: (inroom agent obj0039)
–PR: (nexttofacing agent obj0041)
–PR: (nexttofacing agent obj0022)
–PR: (inroom agent obj0036)
–PR: (inroom agent obj0051)
–PR: (nexttofacing agent obj0058)
–PR: (handsfree agent)
–PR: (nexttofacing agent obj0040)
–PR: (inroom agent obj0035)
–PR: (nexttofacing agent obj0002)
–PR: (inroom agent obj0050)
–PR: (nexttofacing agent obj0026)
–PR: (inroom agent obj0041)
–PR: (nexttofacing agent obj0028)
–PR: (atgoal agent goal)
–PR: (nexttofacing agent obj0054)
–PR: (nexttofacing agent obj0045)
–PR: (nexttofacing agent obj0059)
–PR: (inroom agent obj0056)
–PR: (inroom agent obj0046)
–PR: (inroom agent obj0034)
–PR: (nexttofacing agent ball)
–PR: (nexttofacing agent obj0009)
–PR: (nexttofacing agent obj0010)
–PR: (inroom agent obj0044)
–PR: (nexttofacing agent obj0008)
–PR: (nexttofacing agent obj0025)
–PR: (nexttofacing agent obj0030)
–PR: (nexttofacing agent obj0024)
–PR: (nexttofacing agent obj0011)
–PR: (nexttofacing agent obj0061)
```

```
–PR: (nexttofacing agent key)
–PR: (nexttofacing agent obj0005)
–PR: (inroom agent goal)
–PR: (closed door)
–PR: (nexttofacing agent obj0053)
–PR: (nexttofacing agent agent)
–PR: (inroom agent obj0052)
–PR: (nexttofacing agent obj0044)
–PR: (open door)
–PR: (nexttofacing agent obj0043)
–PR: (inroom agent obj0037)
–PR: (nexttofacing agent obj0019)
–PR: (nexttofacing agent obj0021)
–PR: (nexttofacing agent door)
–PR: (nexttofacing agent obj0050)
–PR: (nexttofacing agent obj0056)
–PR: (inroom agent obj0063)
–PR: (nexttofacing agent obj0047)
–PR: (nexttofacing agent obj0020)
–PR: (nexttofacing agent obj0039)
–PR: (nexttofacing agent obj0007)
–PR: (nexttofacing agent obj0031)
–PR: (inroom agent obj0032)
–PR: (nexttofacing agent obj0000)
–PR: (nexttofacing agent obj0034)
–PR: (nexttofacing agent obj0062)
–PR: (inroom agent obj0060)
–PR: (inroom agent obj0043)
–PR: (nexttofacing agent obj0060)
–PR: (nexttofacing agent obj0038)
–PR: (inroom agent obj0049)
–PR: (nexttofacing agent obj0055)
–PR: (nexttofacing agent obj0001)
–PR: (nexttofacing agent obj0017)
–PR: (nexttofacing agent obj0042)
–PR: (inroom agent obj0061)
–PR: (nexttofacing agent obj0037)
–PR: (nexttofacing agent obj0035)
–PR: (nexttofacing agent obj0036)
–PR: (nexttofacing agent obj0018)
–PR: (nexttofacing agent obj0057)
ADD: (nexttofacing agent ball)
ADD: (handsfree agent)
ADD: (inroom agent key)
ADD: (inroom agent door)
ADD: (locked door)
```

```
DEL: (holding agent key)
DEL: (blocked door)
```