Hierarchical Imitation and Reinforcement Learning

Hoang M. Le [1] **Nan Jiang** [2] **Alekh Agarwal** [2] **Miroslav Dudík** [2] **Yisong Yue** [1] **Hal Daumé III** [3 2]

ICML-2018

Hierarchical RL

Idea: choosing subtasks, finish subtasks.

Ex. Leave a building: go to the elevator→ take the elevator down → walk out

simultaneously learn **HI-level** policy μ : S → G, called the **metacontroller**, and **LO—level** policy π_g : S → A, called the **subpolicies.**

1: **for** $h_{\text{HI}} = 1 \ldots H_{\text{HI}}$ **do**
2: observe state s and choose subgoal $g \leftarrow \mu(s)$
3: **for** $h_{\text{LO}} = 1 \ldots H_{\text{LO}}$ **do**
4: observe state s
5: **if** $\beta_g(s)$ **then break**
6: choose action $a \leftarrow \pi_g(s)$

Termination function

types of supervision

- $\texttt{HierDemo}(s)$: *hierarchical demonstration*. The expert executes its hierarchical policy starting from s and returns the resulting hierarchical trajectory $\sigma^\star = (s_1^\star, g_1^\star, \tau_1^\star, s_2^\star, g_2^\star, \tau_2^\star, \dots)$, where $s_1^\star = s$.
- $\texttt{Label}_{\text{HI}}(\tau_{\text{HI}})$: HI-*level labeling*. The expert provides a good next subgoal at each state of a given HI-level trajectory $\tau_{\text{HI}} = (s_1, g_1, s_2, g_2, \dots)$, yielding a labeled data set $\{(s_1, g_1^\star), (s_2, g_2^\star), \dots\}$.
- $\texttt{Label}_{\text{LO}}(\tau; g)$: LO-*level labeling*. The expert provides a good next primitive action towards a given subgoal g at each state of a given LO-level trajectory $\tau = (s_1, a_1, s_2, a_2, \dots)$, yielding a labeled data set $\{(s_1, a_1^\star), (s_2, a_2^\star), \dots\}$.
- $\texttt{Inspect}_{\text{LO}}(\tau; g)$: LO-*level inspection*. Instead of annotating every state of a trajectory with a good action, the expert only verifies whether a subgoal g was accomplished, returning either *Pass* or *Fail*.
- $\texttt{Label}_{\text{FULL}}(\tau_{\text{FULL}})$: *full labeling*. The expert labels the agent's full trajectory $\tau_{\text{FULL}} = (s_1, a_1, s_2, a_2, \dots)$, from start to finish, ignoring hierarchical structure, yielding a labeled data set $\{(s_1, a_1^\star), (s_2, a_2^\star), \dots\}$.
- $\texttt{Inspect}_{\text{FULL}}(\tau_{\text{FULL}})$: *full inspection*. The expert verifies whether the agent's overall goal was accomplished, returning either *Pass* or *Fail*.

Algorithm 1

Algorithm 1 Hierarchical Behavioral Cloning (**h-BC**)

Initialize data buffers $\mathcal{D}_{\text{HI}} \leftarrow \emptyset$ and $\mathcal{D}_g \leftarrow \emptyset$, $g \in \mathcal{G}$
for $t = 1, \ldots, T$ **do**
    Get a new environment instance with start state s
    $\sigma^\star \leftarrow \texttt{HierDemo}(s)$
    for all $(s_h^\star, g_h^\star, \tau_h^\star) \in \sigma^\star$ **do**
        Append $\mathcal{D}_{g_h^\star} \leftarrow \mathcal{D}_{g_h^\star} \cup \tau_h^\star$
        Append $\mathcal{D}_{\text{HI}} \leftarrow \mathcal{D}_{\text{HI}} \cup \{(s_h^\star, g_h^\star)\}$
Train subpolicies $\pi_g \leftarrow \texttt{Train}(\pi_g, \mathcal{D}_g)$ for all g
Train meta-controller $\mu \leftarrow \texttt{Train}(\mu, \mathcal{D}_{\text{HI}})$

Algorithm 2 Hierarchically Guided DAgger (hg-DAgger)

6-11: execute agent's policy

13-19: label

- High-level
- Low-level

3: **for** $t = T_{\text{warm-start}} + 1, \ldots, T$ **do**
4: Get a new environment instance with start state s
5: Initialize $\sigma \leftarrow \emptyset$
6: **repeat**
7: $g \leftarrow \mu(s)$
8: Execute π_g, obtain LO-level trajectory τ
9: Append (s, g, τ) to σ
10: $s \leftarrow$ the last state in τ
11: **until** end of episode
12: Extract τ_{FULL} and τ_{HI} from σ
13: **if** $\texttt{Inspect}_{\text{FULL}}(\tau_{\text{FULL}}) = \textit{Fail}$ **then**
14: $\mathcal{D}^\star \leftarrow \texttt{Label}_{\text{HI}}(\tau_{\text{HI}})$
15: Process $(s_h, g_h, \tau_h) \in \sigma$ in sequence as long as g_h agrees with the expert's choice $g_h^\star$ in $\mathcal{D}^\star$:
16: **if** $\texttt{Inspect}(\tau_h; g_h) = \textit{Fail}$ **then**
17: Append $\mathcal{D}_{g_h} \leftarrow \mathcal{D}_{g_h} \cup \texttt{Label}_{\text{LO}}(\tau_h; g_h)$
18: **break**
19: Append $\mathcal{D}_{\text{HI}} \leftarrow \mathcal{D}_{\text{HI}} \cup \mathcal{D}^\star$
20: Update subpolicies $\pi_g \leftarrow \texttt{Train}(\pi_g, \mathcal{D}_g)$ for all g
21: Update meta-controller $\mu \leftarrow \texttt{Train}(\mu, \mathcal{D}_{\text{HI}})$

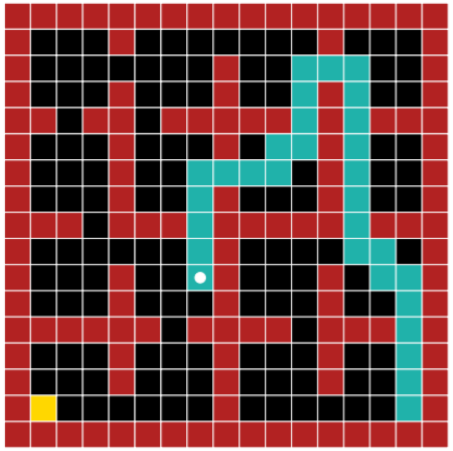
Algorithm 3 Hierarchically Guided DAgger / Q-learning (hg-DAgger/Q)

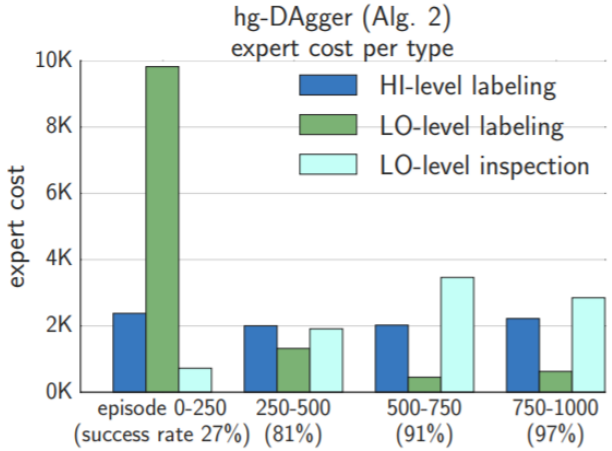
Train subpolicy with RL based on pseudo reward

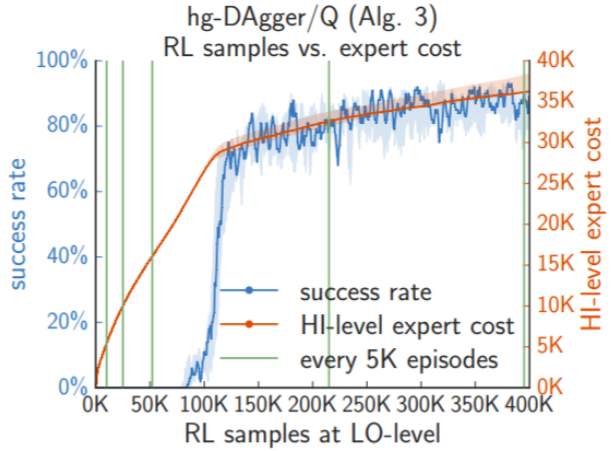
$$\begin{cases} 1 & \text{if } \texttt{success}(s;g) \\ -1 & \text{if } \neg\texttt{success}(s;g) \text{ and } \texttt{terminal}(s;g) \\ -\kappa & \text{otherwise,} \end{cases}$$

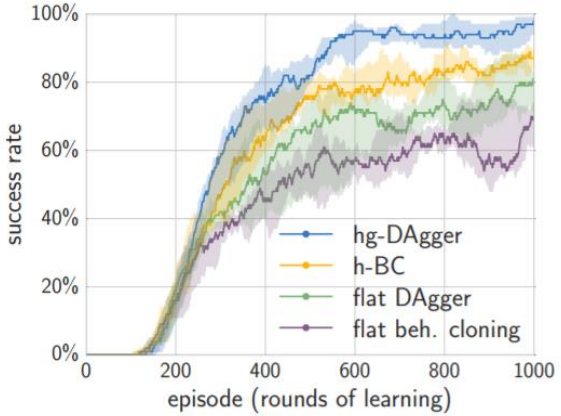
for $t = 1, \ldots, T$ **do**
  Get a new environment instance with start state s
  Initialize $\sigma \leftarrow \emptyset$
  repeat
    $s_{\text{HI}} \leftarrow s$, $g \leftarrow \mu(s)$ and initialize $\tau \leftarrow \emptyset$
    repeat
      $a \leftarrow \epsilon_g$-greedy(Q_g, s)
      Execute a, next state $\tilde{s}$, $\tilde{r} \leftarrow \texttt{pseudo}(\tilde{s}; g)$
      Update Q_g: a (stochastic) gradient descent step on a minibatch from $\mathcal{D}_g$
      Append $(s, a, \tilde{r}, \tilde{s})$ to τ and update $s \leftarrow \tilde{s}$
    until $\texttt{terminal}(s; g)$
    Append (s_{HI}, g, τ) to σ
  until end of episode
  Extract τ_{FULL} and τ_{HI} from σ
  if $\texttt{Inspect}_{\text{FULL}}(\tau_{\text{FULL}}) =$ *Fail* **then**
    $\mathcal{D}^\star \leftarrow \texttt{Label}_{\text{HI}}(\tau_{\text{HI}})$
    Process $(s_h, g_h, \tau_h) \in \sigma$ in sequence as long as g_h agrees with the expert's choice $g_h^\star$ in $\mathcal{D}^\star$:
      Append $\mathcal{D}_{g_h} \leftarrow \mathcal{D}_{g_h} \cup \tau_h$
    Append $\mathcal{D}_{\text{HI}} \leftarrow \mathcal{D}_{\text{HI}} \cup \mathcal{D}^\star$
  else
    Append $\mathcal{D}_{g_h} \leftarrow \mathcal{D}_{g_h} \cup \tau_h$ for all $(s_h, g_h, \tau_h) \in \sigma$
  Update meta-controller $\mu \leftarrow \texttt{Train}(\mu, \mathcal{D}_{\text{HI}})$

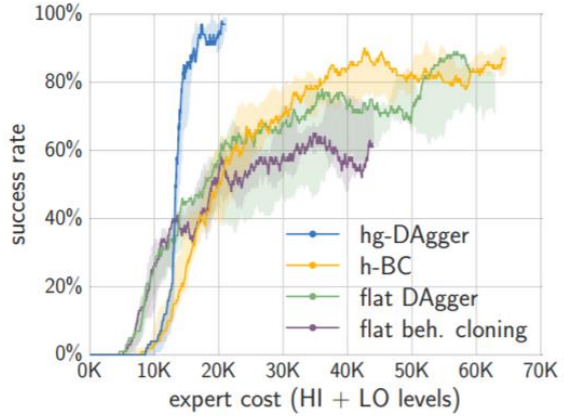
Maze Navigation Domain

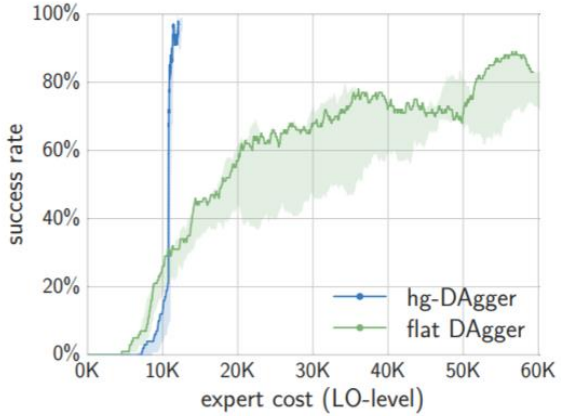












Montezuma's Revenge