SDRL: Interpretable and Data-efficient Deep Reinforcement Learning Leveraging Symbolic Planning

Bo Liu

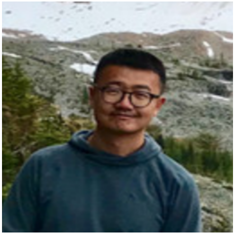
Auburn University, Auburn, AL, USA

Collaborators

SDRL: Symbolic Deep Reinforcement Learning

Liu

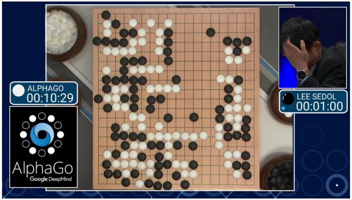
Daoming Lyu
Auburn University
Auburn, AL, USA

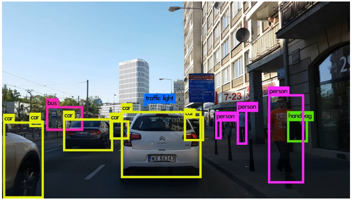


Fangkai Yang
NVIDIA Corporation
Redmond, WA, USA

Steven Gustafson
Maana Inc.
Bellevue, WA, USA

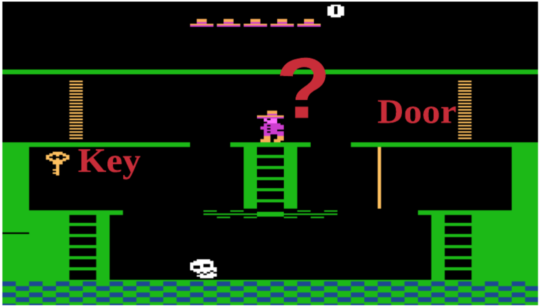
Sequential Decision-Making





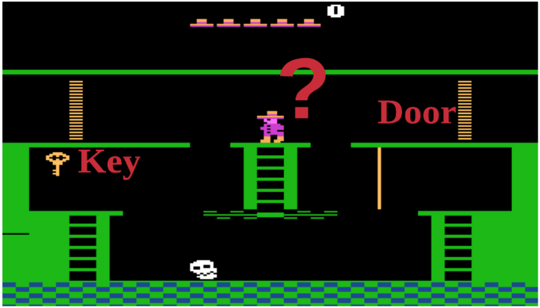
- **Sequential decision-making** (SDM) concerns an agent making a sequence of actions based on its behavior in the environment.
- **Deep reinforcement learning** (DRL) achieves tremendous success on sequential decision-making problems using deep neural networks *(Mnih et al., 2015)*.

Challenge: Montezuma's Revenge



- The avatar: climbs down the ladder, jumps over a rotating skull, **picks up the key (+100)**, goes back and uses the key to **open the right door (+300)**.
- Vanilla DQN achieves 0 score *(Mnih et al., 2015)*.

Challenge: Montezuma's Revenge



- Problem: **long horizon sequential actions**, **sparse and delayed reward**.
  - poor data efficiency.
  - lack of interpretability.

Our Solution

Solution: task decomposition

- Symbolic planning: subtasks scheduling (high-level plan).
- DRL: subtask learning (low-level control).
- Meta-learner: subtask evaluation.

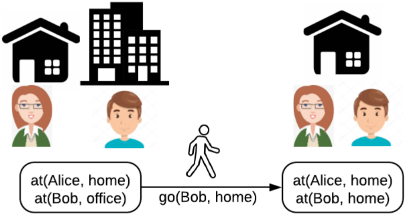
Goal

- Symbolic planning drives learning, improving **task-level interpretablility**.
- DRL learns feasible subtasks, improving **data-efficiency**.

Background: Symbolic Planning with Action Language

Action language *(Gelfond & Lifschitz, 1998)*: a formal, declarative, logic-based language that describes dynamic domains.

- Dynamic domains can be represented as a transition system.



Action Language $\mathcal{BC}$

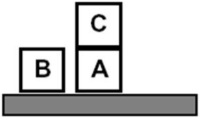
Action Language $\mathcal{BC}$ (Lee et al., 2013) is a language that describes the transition system using a set of *causal laws*.

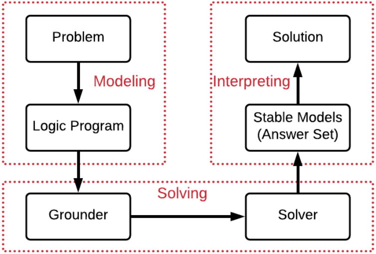
- *dynamic laws* describe transition of states

$$move(x, y_1, y_2) \text{ causes } on(x, y_2) \text{ if } on(x, y_1).$$

- *static laws* describe value of fluents inside a state

$$intower(x, y_2) \text{ if } intower(x, y_1), on(y_1, y_2).$$

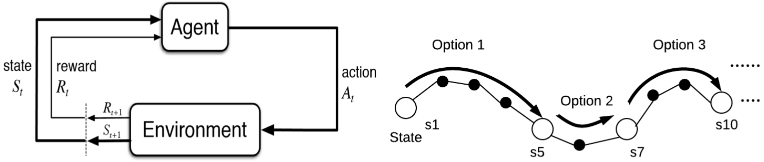




Background: Reinforcement Learning

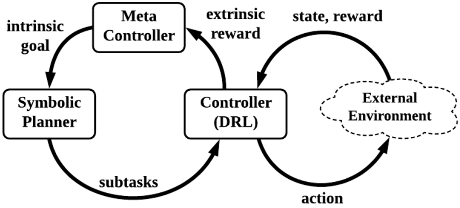
SDRL: Symbolic Deep Reinforcement Learning

Liu



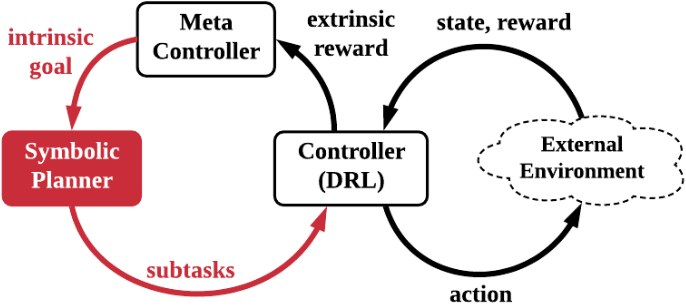
- **Reinforcement learning** is defined on a Markov Decision Process $(\mathcal{S}, \mathcal{A}, P^a_{ss'}, r, \gamma)$. To achieve optimal behavior, a policy $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is learned.
- An **option** is defined on the tuple $(I,\ \pi,\ \beta)$, which enables the decision-making to have a hierarchical structure:
  - the initiation set $I \subseteq \mathcal{S}$,
  - policy π: $\mathcal{S} \times \mathcal{A} \mapsto [0, 1]$,
  - probabilistic termination condition β: $\mathcal{S} \mapsto [0, 1]$.

SDRL: Symbolic Deep Reinforcement Learning



- **Symbolic Planner**: orchestrates sequence of *subtasks* using high-level symbolic plan.
- **Controller**: uses DRL approaches to learn the subpolicy for each subtask with *intrinsic rewards*.
- **Meta-Controller**: measures learning performance of subtasks, updates intrinsic goal to enable reward-driven plan improvement.

Symbolic Planner



Symbolic Planner: Planning with Intrinsic Goal

- **Intrinsic goal**: a linear constraint on plan quality $quality \geq quality(\Pi_t)$ where Π_t is the plan at episode t.
- Plan quality: a utility function

$$quality(\Pi_t) = \sum_{\langle s_{i-1}, g_{i-1}, s_i \rangle \in \Pi_t} \rho^{g_{i-1}}(s_{i-1})$$

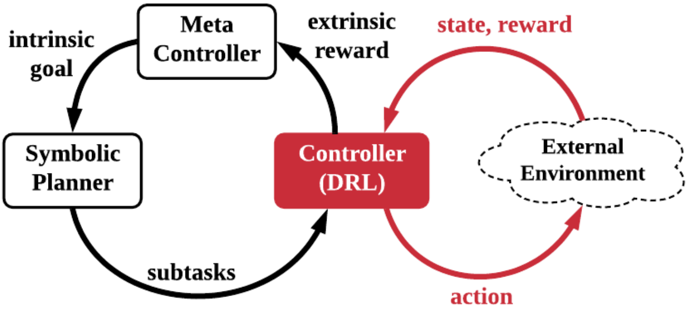
  where ρ^{g_i} is the gain reward for subtask g_i.
- Symbolic planner: generates a new plan that
  - **explores** new subtasks,
  - **exploits** more rewarding subtasks.

From Symbolic Transition to Subtask

- Assumption: given the set $\mathcal{S}$ of symbolic states and $\widetilde{\mathcal{S}}$ of sensory input, we assumed there is an *Oracle* for symbol grounding: $\mathbb{F} : \mathcal{S} \times \widetilde{\mathcal{S}} \mapsto \{\mathbf{t}, \mathbf{f}\}$.
- Given $\mathbb{F}$ and a pair of symbolic states $s, s' \in \mathcal{S}$:
  - initiation set $I = \{\tilde{s} \in \widetilde{\mathcal{S}} : \mathbb{F}(s, \tilde{s}) = \mathbf{t}\}$,
  - $\pi : \widetilde{\mathcal{S}} \mapsto \widetilde{\mathcal{A}}$ is the subpolicy for the corresponding subtask,
  - β is the termination condition such that

$$\beta(\tilde{s'}) = \begin{cases} 1 & \mathbb{F}(s', \tilde{s'}) = \mathbf{t}, \text{for } \tilde{s'} \in \widetilde{\mathcal{S}}, \\ 0 & \text{otherwise.} \end{cases}$$

Controller


Meta Controller
intrinsic goal
extrinsic reward
state, reward
Symbolic Planner
Controller (DRL)
External Environment
subtasks
action

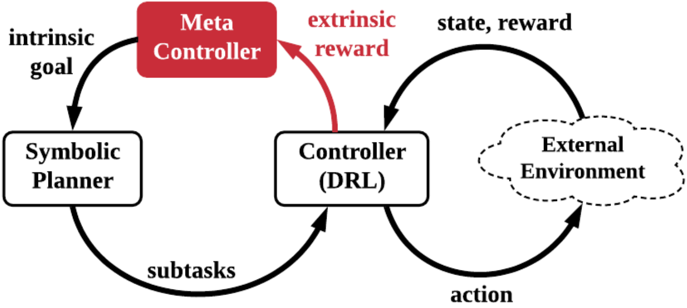
Controllers: DRL with Intrinsic Reward

- **Intrinsic reward**: pseudo-reward crafted by the human.
- Given a subtask defined on (I, π, β), intrinsic reward

$$r_i(\tilde{s'}) = \begin{cases} \phi & \beta(\tilde{s'}) = 1 \\ r & \text{otherwise} \end{cases}$$

where ϕ is a positive constant encouraging achieving subtasks and r is the reward from the environment at state $\tilde{s'}$.

Meta-Controller



Meta-Controller: Evaluation with Extrinsic Reward

- **Extrinsic rewards**: $r_e(s,g) = f(\epsilon)$ where ϵ can measure the competence of the learned subpolicy for each subtask.
- For example, let ϵ be the **success ratio**, f can be defined as

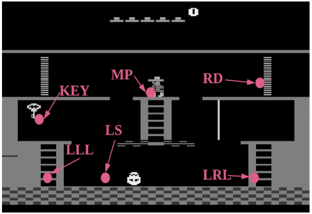
$$f(\epsilon) = \begin{cases} -\psi & \epsilon < \mathit{threshold} \\ r(s,g) & \epsilon \geq \mathit{threshold} \end{cases}$$

  - ψ is a positive constant to punish selecting unlearnable subtasks,
  - $r(s,g)$ is the cumulative environmental reward by following the subtask g.

Experimental Results I.


```
% object declaration
location(mp;rd;ls;lll;lrl;key).
% dynamic causal law declaration
move(L) causes loc=L if location(L).
move(L) causes cost=L+Z if rho((at(L1)),move(L))=Z,
          loc=L1,picked(key)=false.
move(L) causes cost=L+Z if rho((at(L1),picked(key)),
          move(L))=Z,loc=L1,picked(key)=true.
inertial loc. inertial quality.
% static causal law declaration
picked(key)=true if loc=key.
nonexecutable move(key) if picked(key).
default rho((at(L1)),move(L))=10.
default rho((at(L1),picked(key)),move(L))=10.
```

No.	subtask	policy learned	in optimal plan
1	MP to LRL, no key	✓	✓
2	LRL to LLL, no key	✓	✓
3	LLL to key, no key	✓	✓
4	key to LLL, with key	✓	✓
5	LLL to LRL, with or without key	✓	✓
6	LRL to MP, with or without key	✓	✓
7	MP to RD, with key	✓	✓
8	LRL to LS, with or without key	✓	
9	LS to key, with or without key	✓	
10	MP to RD, no key	✓	
11	LRL to key, with or without		
12	key to LRL, with key		
13	LRL to RD, with key		



MP: middle platform

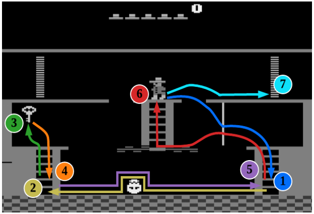
LRL: lower right ladder

LLL: lower left ladder

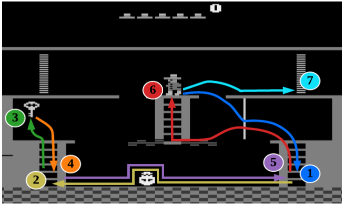
KEY: key

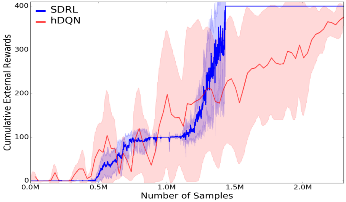
LS: left of rotating skull

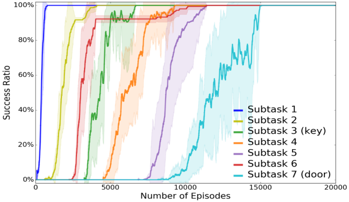
RD: right door

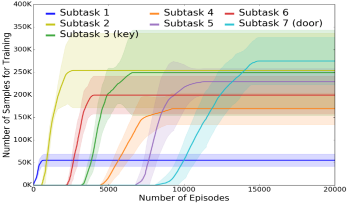


Experimental Results II.









Baseline: Kulkarni et. al, *Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation*, NIPS'2016.

Conclusion

- We present a **SDRL** framework features:
  - **High-level symbolic planning** based on intrinsic goal
  - **Low-level policy control** with DRL.
  - **Subtask learning evaluation** by a meta-learner.
- This is the first work on integrating symbolic planning with DRL that achieves both **task-level interpretability** and **data-efficiency** for decision-making.
- Future work.