Generalisation in Lifelong RL through Logical Composition

Geraud Nangue Tasse*, Steven James and Benjamin Rosman

University of the Witwatersrand, Johannesburg, South Africa

WITS

R A I L
L A B

We leverage logical composition in lifelong RL to achieve both zero-shot and few-shot transfer leading to fast generalisation over unknown task distributions.

Introduction

- Given a new task, can we determine if it is **expressible** in terms of learned ones? If yes, can we solve it **zero-shot**? If no, can we solve it **few-shot**? How about **generalisation** over any unknown non-stationary task distribution?
- Prior works [1,2] achieve a subset of these by **assuming base skills are learned**. Most lifelong RL works [3] focus on learning new tasks faster but do not consider the generalisation problem (they have to **learn all or most new tasks**).

Logical composition

To achieve **zero-shot composition**, the agent learns an **extended value function (EVF)** for each task M with reward function $r_M(s,a)$:

$$Q(s,g,a) = \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty} \gamma^t \bar{r}(s_t, g, a_t)\right]$$

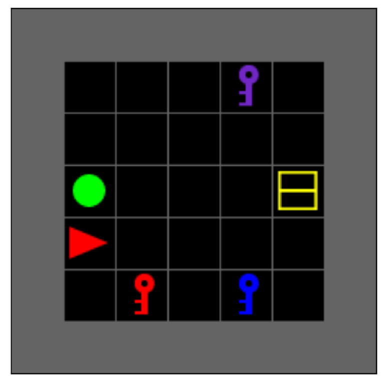
where $\bar{r}(s,g,a) = \begin{cases} r_{MIN} & \text{if } g \neq s \in G, \\ r_M(s,a) & \text{otherwise.} \end{cases}$

The agent can recover the task policy from an EVF as follows: $\boldsymbol{\pi(s) \sim argmax_a \max_g Q(s,g,a)}$.

The EVFs can then be composed as follows:
$\boldsymbol{Q_1 \vee Q_2 = \max\{Q_1, Q_2\}}$, $\boldsymbol{Q_1 \wedge Q_2 = \min\{Q_1, Q_2\}}$, and $\boldsymbol{\neg Q = Q_{MAX} \text{ if } (|Q - Q_{MIN}| \leq |Q - Q_{MAX}|) \text{ else } Q_{MIN}}$.

[1] **G. Nangue Tasse**, S. James, B. Rosman. **A Boolean task algebra for reinforcement learning**. NeurIPS 2020.
[2] **A. Barreto**, Shaobo Hou, Diana Borsa, David Silver, Doina Precup. **Fast reinforcement learning with generalized policy updates**. NAS 2020.
[3] **D. Abel**, Y. Jinnai, S. Y. Guo, G. Konidaris, M. Littman. **Policy and value transfer in lifelong reinforcement learning**. ICML 2018.

SOPGOL



Goals															
	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Learned

For each episode:

- Sum of products: $T_{SOP} := \neg \wedge \neg \wedge \wedge \neg$ and $Q_{SOP} := \neg Q^*() \wedge \neg Q^*() \wedge Q^*() \wedge \neg Q^*()$
- $T = T_{SOP}$? **(No!)**
- If yes, use $\boldsymbol{\pi \sim Q_{SOP}}$ and don't add anything to library.
- If no, learn a new $\boldsymbol{Q}$ with goal-oriented learning, using $\boldsymbol{\pi \sim Q \vee Q_{SOP}}$ to speed up training.

After n episodes (or when $\boldsymbol{Q}$ is sufficiently good), add $(\mathbf{T}, \boldsymbol{Q})$ to the library if $\boldsymbol{T \neq T'}$.

Fast transfer and generalization in lifelong RL

Theorem 1: Let $\widetilde{\boldsymbol{T}}$ be the learned binary represention for a given deterministic task. Given the learned binary representions $\widetilde{\boldsymbol{T}}_{\boldsymbol{n}}$ and Q-functions $\widetilde{\boldsymbol{Q}}_{\boldsymbol{n}}$ for n tasks, we have

$$\|Q^* - Q_{SOP}\|_\infty \leq (\mathbf{1}_{T \neq T_{SOP}}) r_\Delta + \epsilon$$

where $\boldsymbol{Q_{SOP}}$ and $\boldsymbol{T_{SOP}}$ are obtained via logical composition using the Boolean expression obtained by the sum of products method, $\boldsymbol{SOP(\widetilde{T}_n, \widetilde{T})}$.

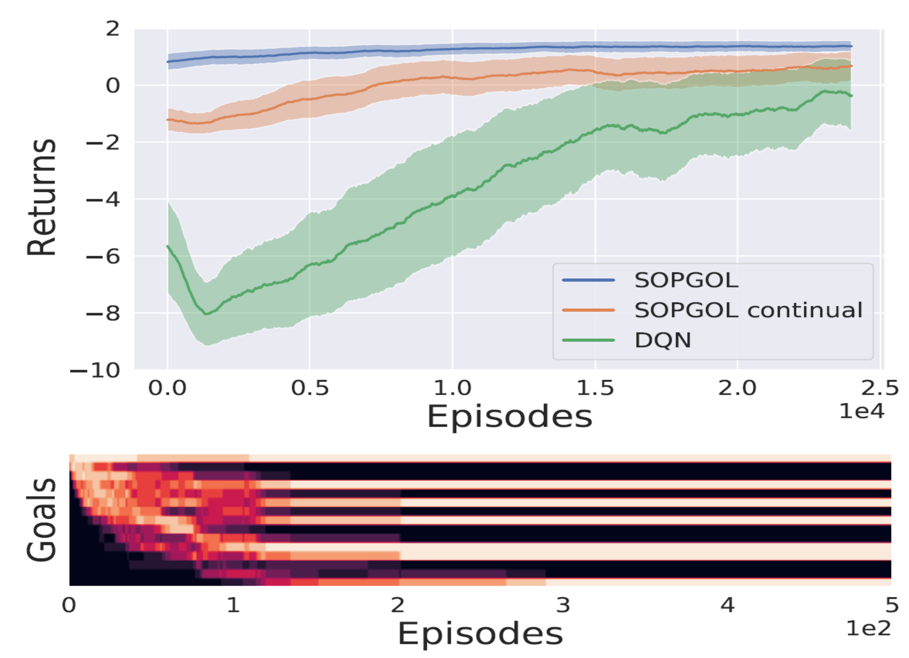
Theorem 2: Let the t^{th} task be sampled from an unknown (possibly non-stationary) task distribution. Let $Skills_{t+1}$ be the library of skills stored by SOPGOL after learning the t^{th} task. Then,

$$\lceil \log(|goals|) \rceil \leq \lim_{t \to \infty} (|\boldsymbol{skills_t}|) \leq |goals|$$

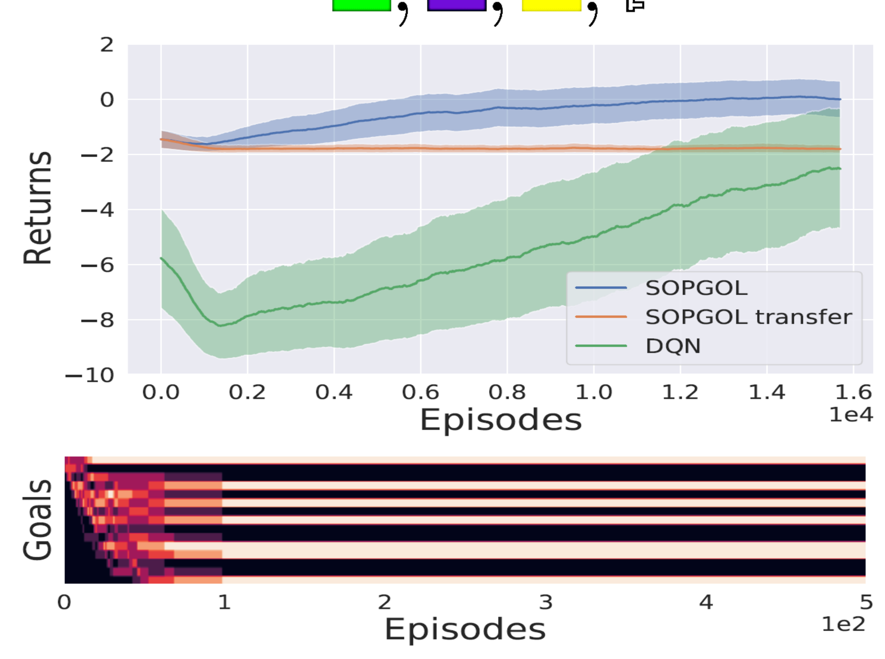
Experiment: Transfer after pretraining

Goals															
T_2	0	0	1	0	1	1	1	0	1	1	0	0	1	0	0

Pretrained on base tasks: M_a, M_b, M_c, M_d



Pretrained on non-base tasks:



Inferred: $(M_a \wedge \neg M_b \wedge \neg M_d) \vee (M_a \wedge M_c \wedge M_d) \vee (\neg M_a \wedge M_b \wedge \neg M_c \wedge \neg M_d) \vee (\neg M_a \wedge \neg M_b \wedge \neg M_c \wedge M_d)$

Inferred: $(\wedge \neg \wedge \neg) \vee (\neg \wedge \wedge \neg \wedge) \vee (\neg \wedge \neg \wedge \wedge) \vee (\neg \wedge \neg \wedge \neg)$

Experiment: Lifelong transfer (~ 1 Trillion tasks)