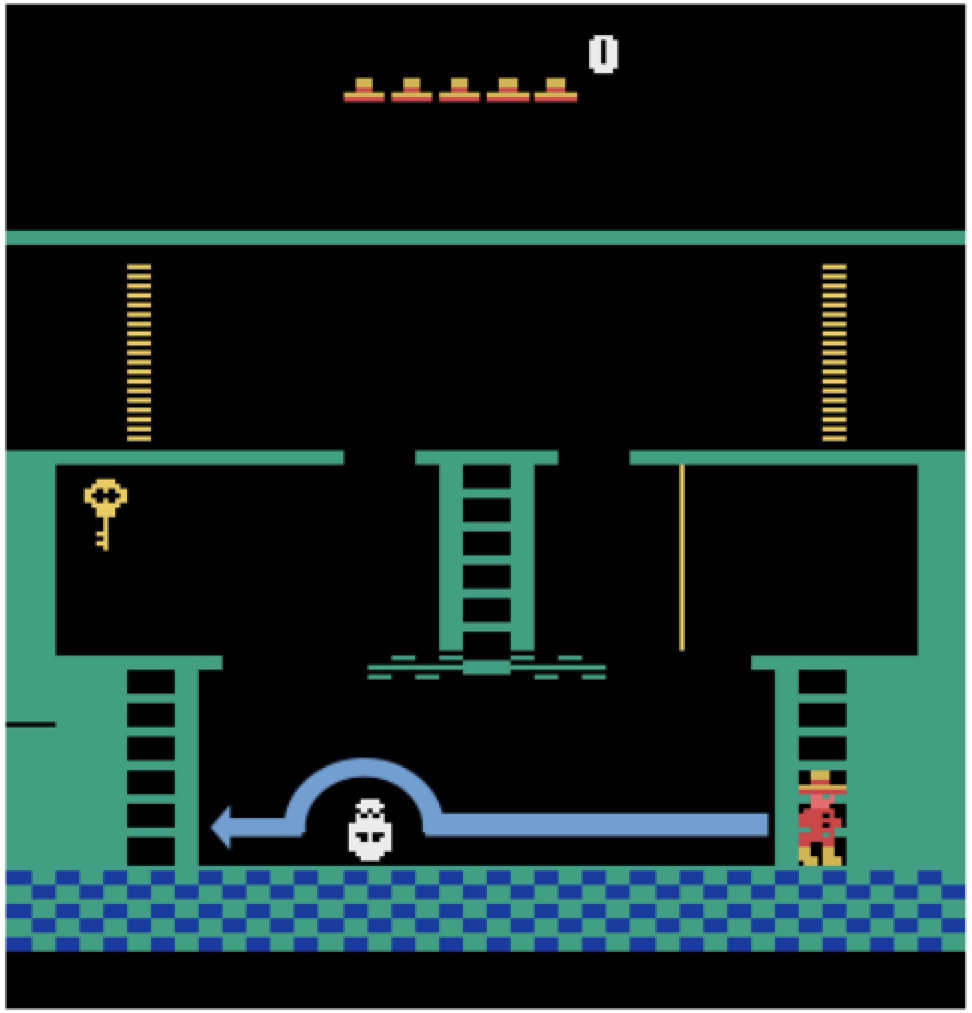
Using Natural Language for Reward Shaping in Reinforcement Learning

Prasoon Goyal , Scott Niekum and Raymond J. Mooney

Presenter: Hsiang-Shun Shih
hsiangss@uci.edu

Make learning more efficient

- Montezuma's revenge.
- Use reward shaping to reduce interaction time.
- Propose the LanguagE-Action Reward Network(LEARN) to map free-form natural language instructions to intermediate rewards.



Reward Shaping

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping.

- Rather than running reinforcement learning algorithm on MDP:

  $M = \{S, A, T, \gamma, R\}$
- Run it on transformed MDP, M':

  $M' = \{S, A, T, \gamma, R'\}$

  $R' = R + F$

Shaping reward function

$R' = R + F$

$F: S \times A \times S \mapsto \mathbb{R}$

- In original MDP, if we receive reward $R(s, a, s')$ from s to s' on a, in M', we would receive reward $R(s, a, s') + F(s, a, s')$.

Some examples

- To encourage moving towards a goal, a possible shaping-reward function is:

$$\begin{cases} F(s,a,s') = r, if\ s'\ is\ closer\ to\ the\ goal\ than\ s \\ F(s,a,s') = 0, otherwise \end{cases}$$

- To encourage taking action a_1, set F as:

$$\begin{cases} F(s,a,s') = r, if\ a = a_1 \\ F(s,a,s') = 0, otherwise \end{cases}$$

Feasibility

- In many case, $M = \{S, A, T, \gamma, R\}$ is not explicitly given.
- We have to learn through taking actions and by observing the resulting state transition rewards.
- Make the observed reward $R(s, a, s') + F(s, a, s')$.

Optimal policy guarantee

- M' is an aid to helping the agent to learn faster.
- We have to guarantee that $\pi^*_{M'}$, the optimal policy in M', will also be optimal in M?
- How can we find such forms of shaping-reward functions F?

Theorem

- If $F(s, a, s')$ is a potential-based shaping function, then every optimal policy in M' will also be an optimal policy in M.
- If there is a real-valued function, ϕ: $S \mapsto \mathbb{R}$,

  let $F(s, a, s') = \gamma\phi(s') - \phi(s)$

  We can say F is potential-based.

Proof

- For MDP M, the optimal Q function satisfies Bellman equation:

$$Q_M^*(s,a) = E_{s\prime}[R(s,a,s') + \gamma \max_{a\prime \epsilon A} Q_M^*(s',a')]$$

- For M',

$$Q_M^*(s,a) - \phi(s)$$
$$= E_{s'}\left[R(s,a,s') + \gamma\phi(s') - \phi(s) + \gamma \max_{a' \epsilon A} Q_M^*(s',a') - \phi(s')\right]$$

- Define $\hat{Q}_{M\prime}(s,a) = Q_{M\prime}^*(s,a) - \phi(s)$, $F(s,a,s') = \gamma\phi(s') - \phi(s)$,

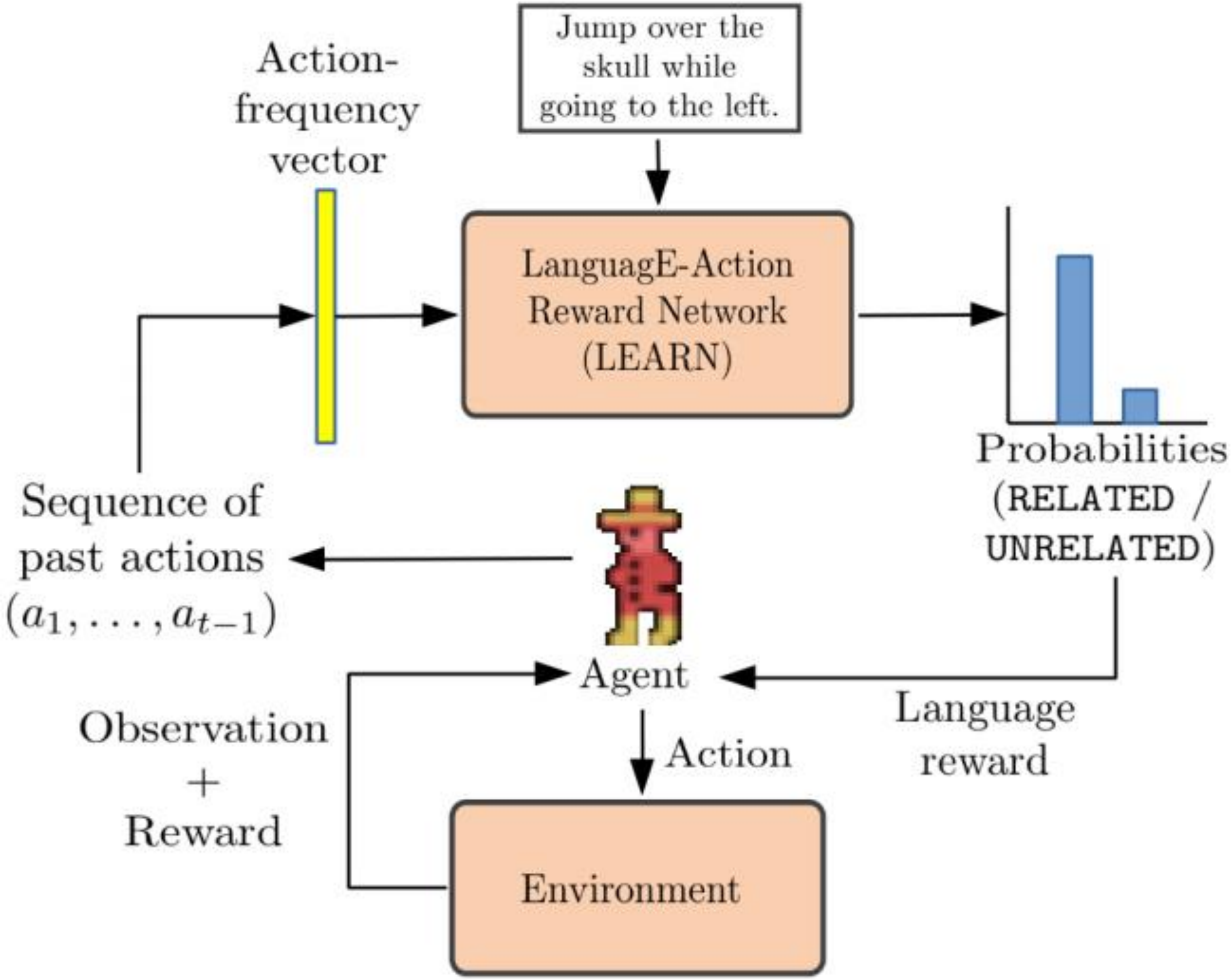
$$\hat{Q}_{M\prime}(s,a) = E_{s\prime}[R'(s,a,s') + \gamma \max_{a\prime \epsilon A} \hat{Q}_{M\prime}(s',a')]$$

MDP+L

- An extension of the MDP framework.
- $< S, A, R, T, \gamma, l >, l \in L$
- L defines all possible language commands.
- Learn an optimal policy under reward function $(R_{ext} + R_{lang})$.

LanguagE-Action Reward Network (LEARN)

- Takes a pair of (trajectory, language).
- Predicts if the language describes the actions within the trajectory.



LanguagE-Action Reward Network (LEARN)

- Transform trajectory τ into action-frequency vector f_t
$$(\tau, l) \rightarrow (f, l)$$
- The dimensionality of f is equal to the number of actions in the MDP+L. The kth component of f is the fraction of timesteps action k appears in τ
- Then, generate positive/negative examples of (f, l)

LanguagE-Action Reward Network (LEARN)

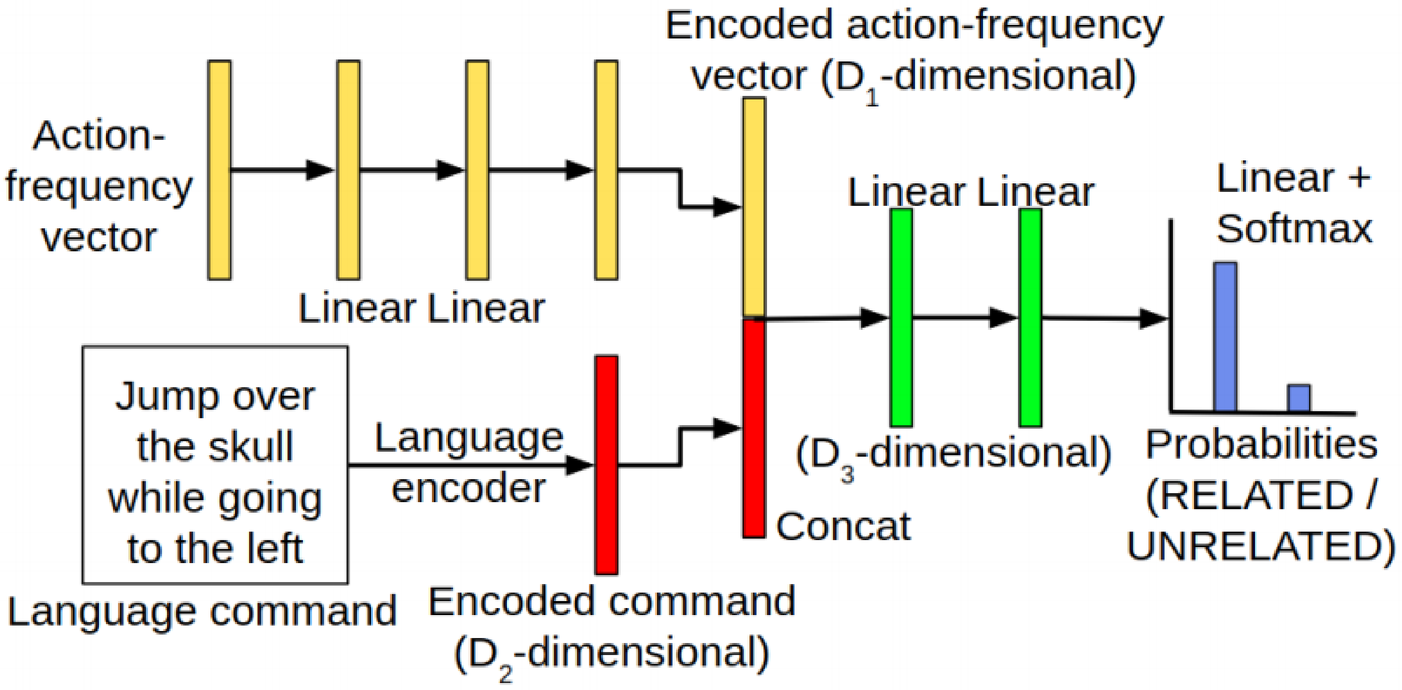


Figure 3: Neural network architecture for LEARN (Section 3.1)

LanguagE-Action Reward Network (LEARN)

To embed the natural language instruction into a vector, experiment with three models:

1. Infersent
2. GloVe+RNN
3. RNNOnly

The final output is a probability distribution over two classes:

1. Related
2. Unrelated

Language-aided RL

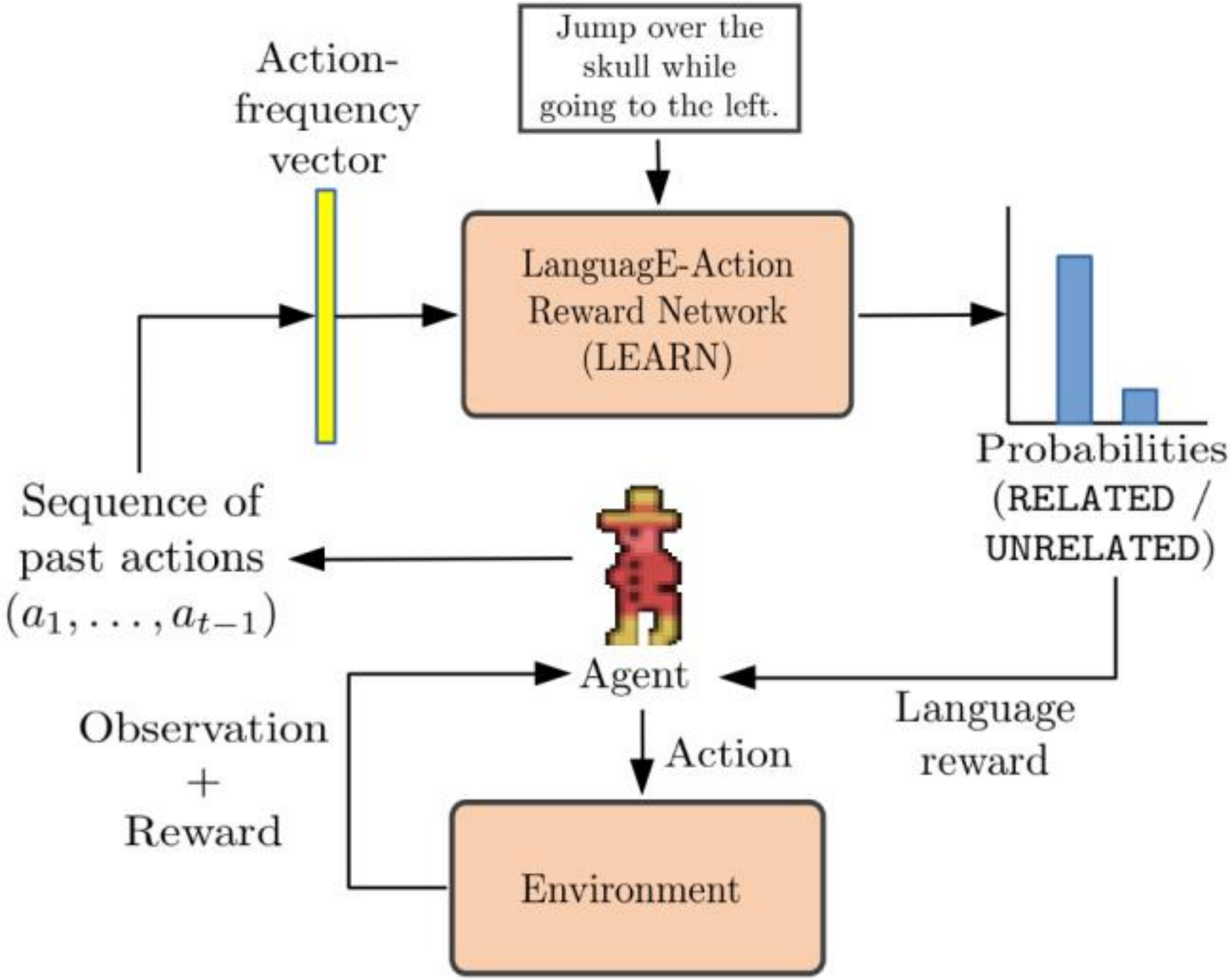
- Action-frequency vector f_t
- Softmax result:

  $P_R(f_t) \Rightarrow$ Related

  $P_V(f_t) \Rightarrow$ Unrelated
- Potential function $\phi(f_t) = P_R(f_t) - P_V(f_t)$
- $R_{lang}(f_t) = \gamma\phi(f_t) - \phi(f_{t-1})$
- $R_{ext} + R_{lang}$

Language-aided RL

- Given the trajectory executed by the agent so far and the language instruction, use LEARN to predict whether the agent is making progress.

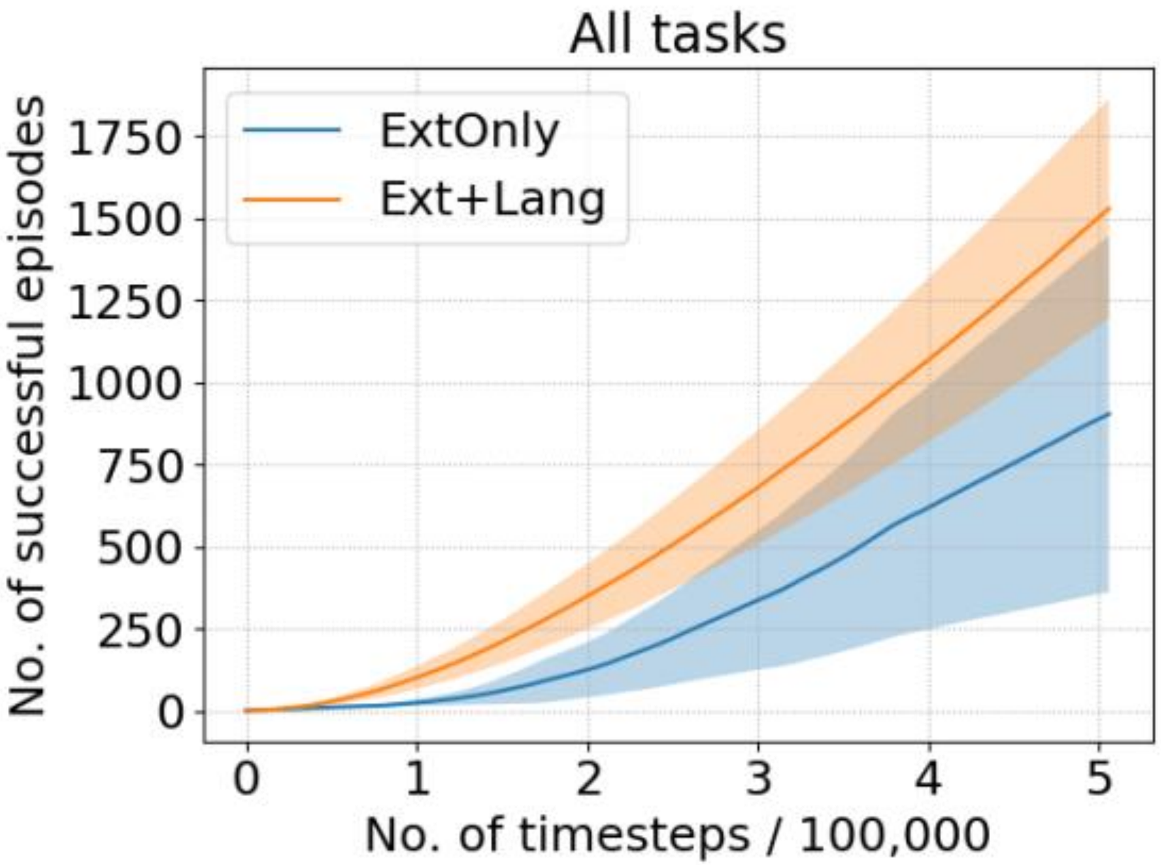


Experimental Evaluation

- Define a set of 15 diverse tasks in multiple rooms, each of which requires the agent to go from a fixed start position to a fixed goal position while interacting with some of the objects present in the path.
- For each task, the agent gets an extrinsic reward of +1 from the environment for reaching the goal, and an extrinsic reward of zero in all other cases.
- $R_{total} = R_{ext} + \lambda R_{lang}$
- Proximal Policy Optimization is used.

Experimental Evaluation RL

- ExtOnly: use the original environment reward.
- Ext+Lang: in addition to the rewards after completing the task successfully, R_{lang} is added.



Result

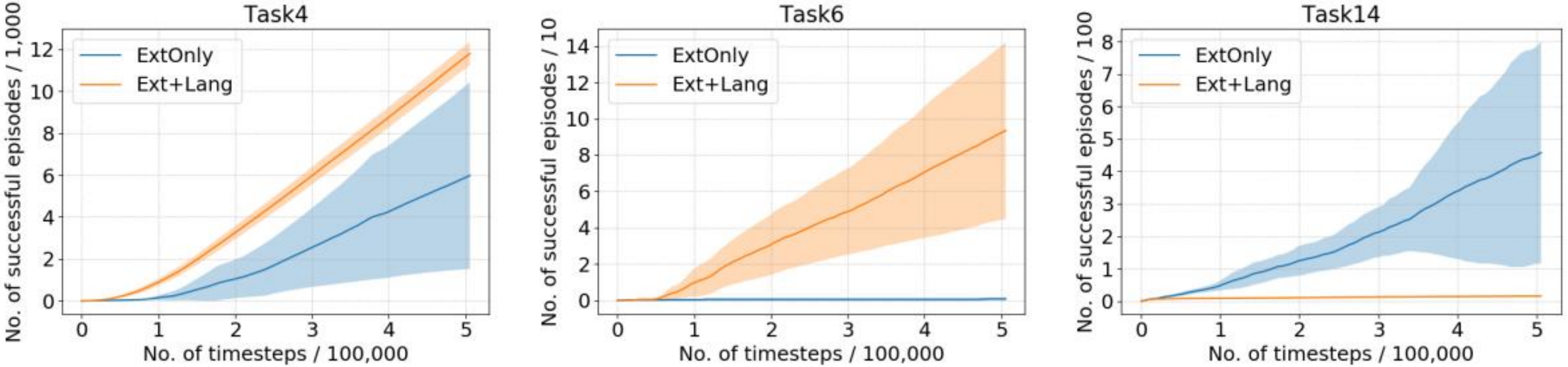
- On average: 30% speed-up, 60% success rate improvement.
- For each case:

	Improvement	Deterioration	No difference
AUC	11	1	3
Final Policy	8	0	7

Analysis of language-based reward

Task Id	Description	Correlation coefficients of different actions							
		NO-OP	JUMP	UP	RIGHT	LEFT	DOWN	JUMP-RIGHT	JUMP-LEFT
4	climb down the ladder	-0.60	-0.58	-0.59	-0.61	-0.55	0.07	-0.57	-0.56
	go down the ladder to the bottom	-0.58	-0.58	-0.58	-0.60	-0.53	0.09	-0.59	-0.60
	move on spider and down on the lader	-0.58	-0.54	-0.59	-0.60	-0.49	0.10	-0.58	-0.56
6	go to the left and go under skulls and then down the ladder	-0.37	-0.40	-0.49	-0.43	0.33	0.16	-0.46	-0.01
	go to the left and then go down the ladder	-0.24	-0.26	-0.35	-0.31	0.28	0.36	-0.34	-0.04
	move to the left and go under the skulls	-0.16	-0.25	-0.60	-0.48	0.27	-0.63	-0.52	-0.40
14	Jump once then down	0.00	0.07	-0.15	-0.13	0.51	0.50	0.09	0.52
	go down the rope and to the bottom	-0.03	0.10	-0.16	0.56	0.54	0.33	0.28	0.01
	jump once and climb down the stick	0.11	0.11	0.06	0.04	0.14	0.40	0.25	0.11

Analysis of language-based reward



Question

- Describe the Language-aided method used in this paper. What steps should be taken to generate the immediate rewards for reinforcement learning?

Thanks for listening