Natural Language Instruction-following with Task-related Language Development and Translation

Jing-Cheng Pang*, Xinyu Yang*, Si-Hang Yang, Xiong-Hui Chen, Yang Yu

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
Polixir Technology

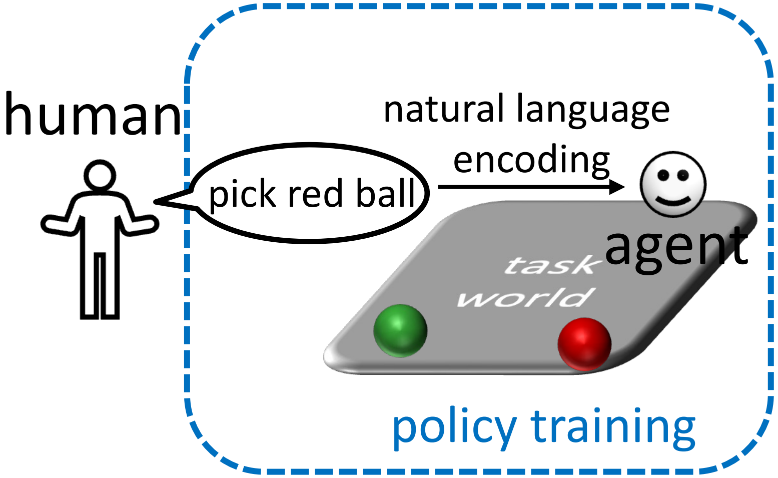
Presented by Jing-Cheng Pang
October, 2023

Background

Instruction-following agent

Background

Previous approaches:

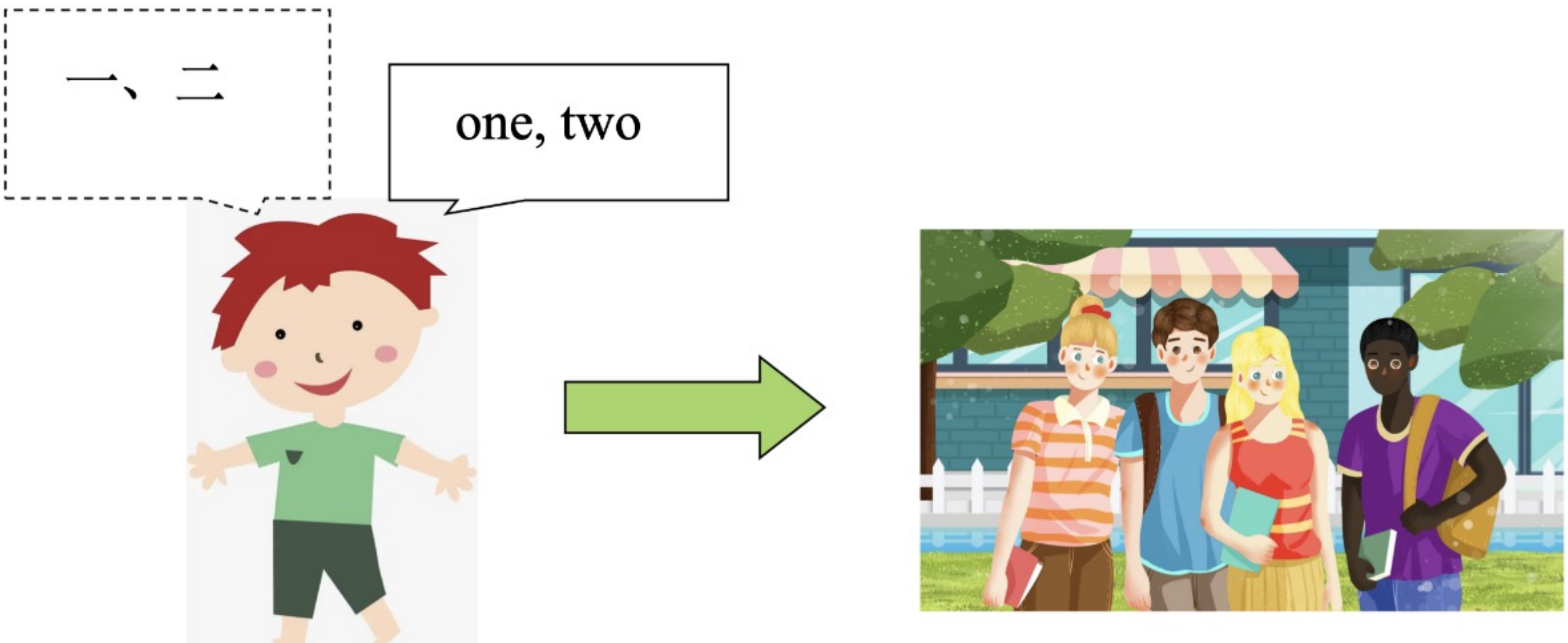


Outside-in learning

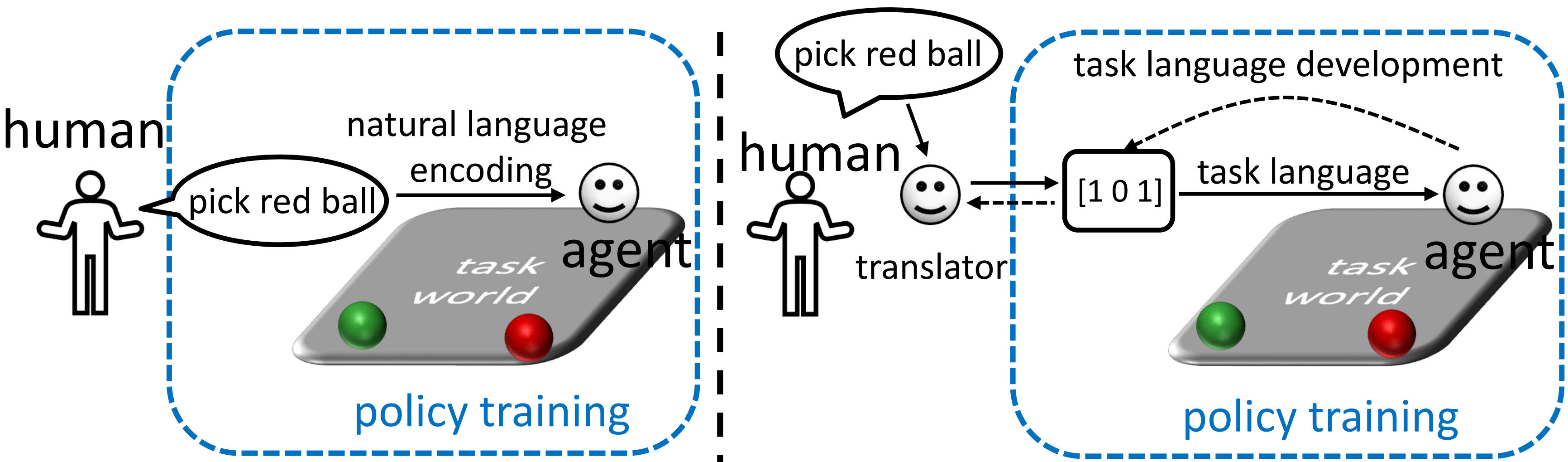
Unbounded representation of natural language instruction brings extra burden for policy learning.

Motivation

But how do we communicate with foreigners?

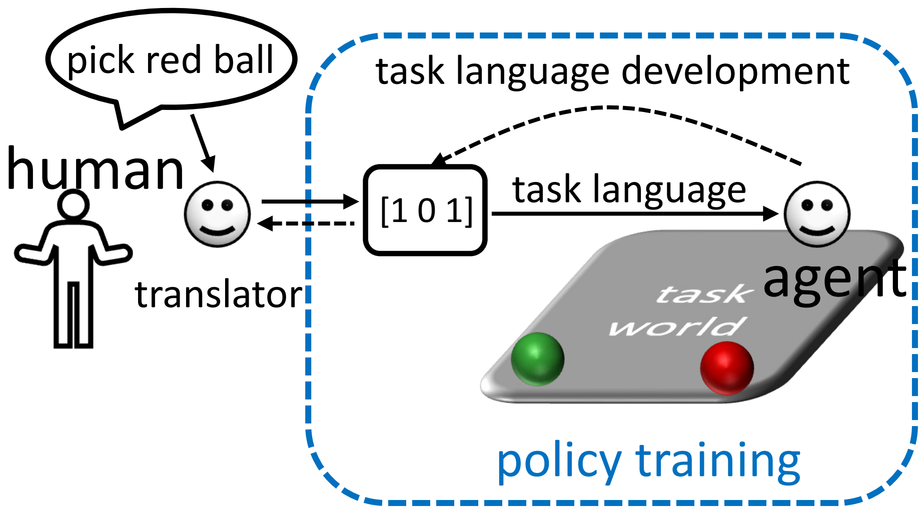


Background



Outside-in learning **Inside-out learning (ours)**

Inside-out learning



1. Task language generator
2. Natural language translator
3. Policy learning

Inside-out learning

1. Task language generator
   Develop TL to effectively and succinctly convey the task's objective to the policy.

2. Natural language translator
   Connect policy with the task language.

3. Policy learning
   Improve policy through task language following.

Method

Learning And Mining from DatA
http://www.lamda.nju.edu.cn

Task Language Development in Predicate Representation

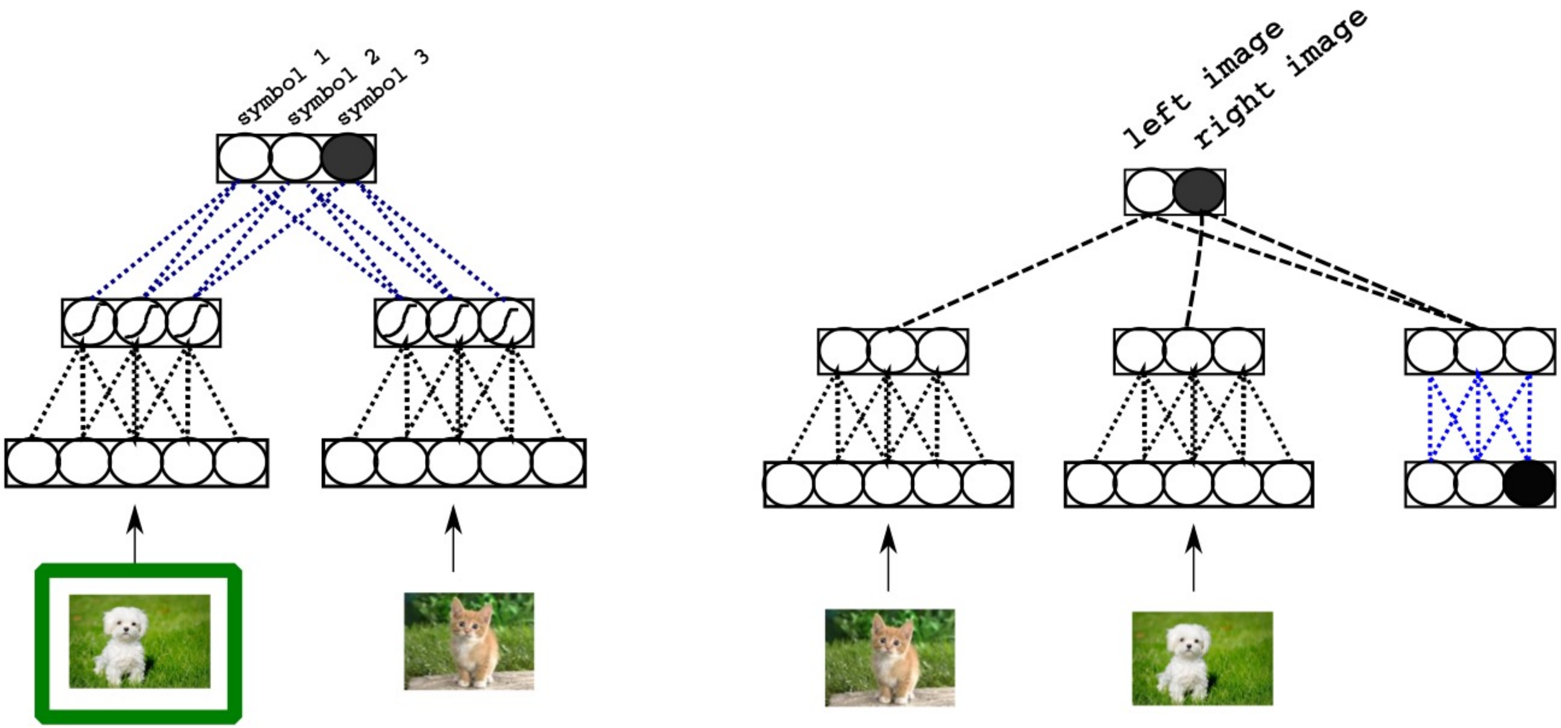
Predicate representation utilizes binary codes to represent diverse relationships in the world.

Examples:
[0, 1, 1]
[1, 0, 0]
[0, 1, 0, 1, 1, 0, 0, 0]

Figure 8: An illustration of the interpretability of predicate representation. For an anonymous predicate expression: `Pred(cat,grassland)=True`, we can guess that `Pred` represents [above].

Method

Referential game develops meaningful language through multi-agent communication.



Sender

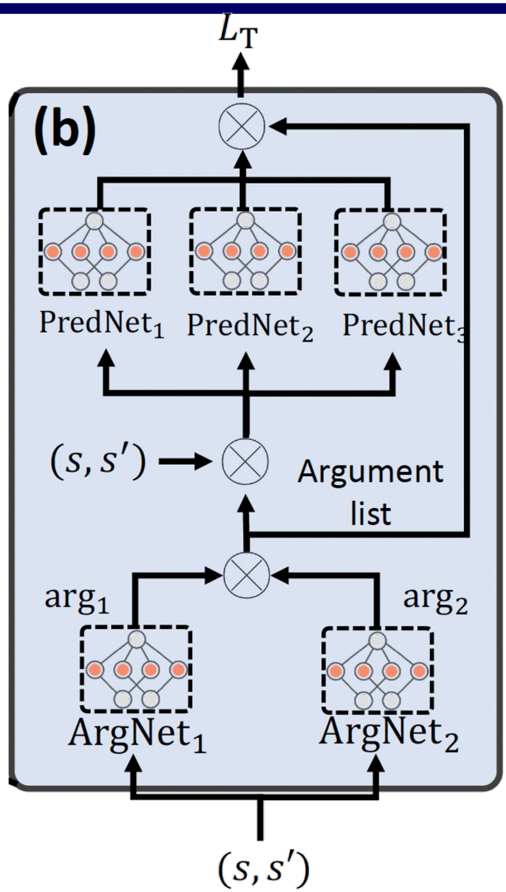
Receiver

Method

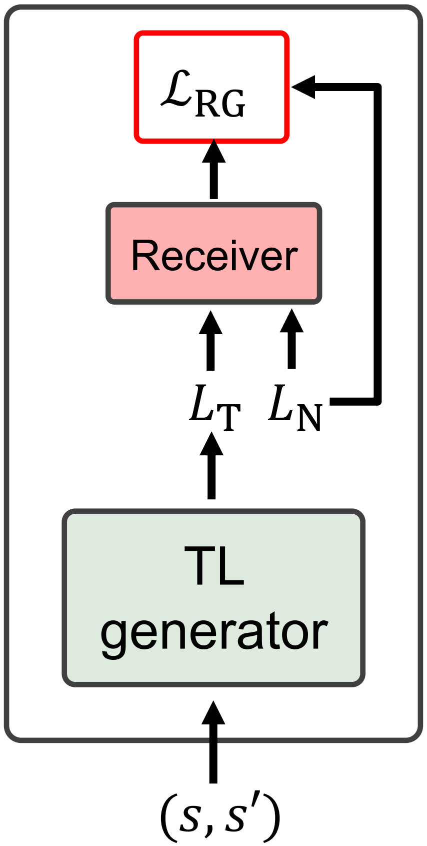
Task dataset:

$A\ task\ dataset\ \mathcal{D} = \{(s, s', L_N)_i\}\ consists\ of\ multiple\ triplets.\ Each\ triplet\ contains\ a\ natural\ language\ instruction\ L_N\ and\ a\ task\ state\ \ pair\ (s, s'), where\ L_N\ describes\ state\ change\ from\ s\ to\ s'\ in\ natural\ language, e.g.,$ $\color{red}{move\ the\ red\ ball\ to\ the\ blue\ ball}$.

Method



Predicate representation

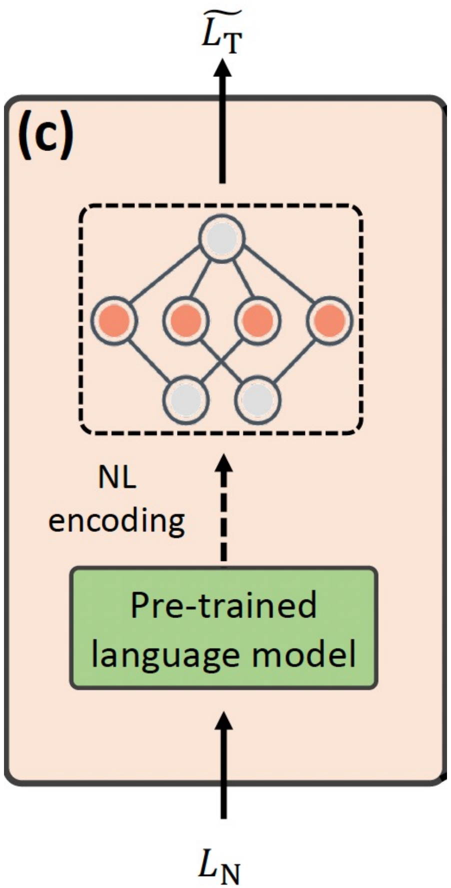


Referential Game

$$\mathcal{L}_{\text{RG}}(\theta, \delta) = \mathop{\mathbb{E}}_{(s,s',L_N)\sim\mathcal{D}, T_i\sim T_{L_N}} \left[-\log f_\delta(T_i | g_\theta(s, s'), b(T_{1,\ldots,i-1})) \right]$$

Method

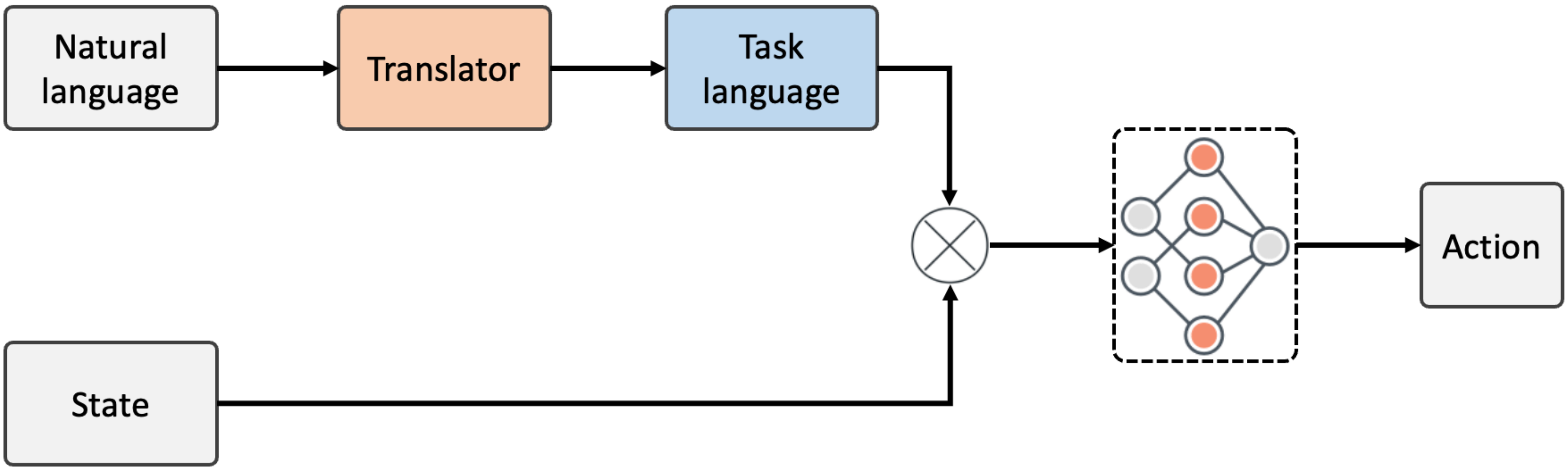
Natural language translation:



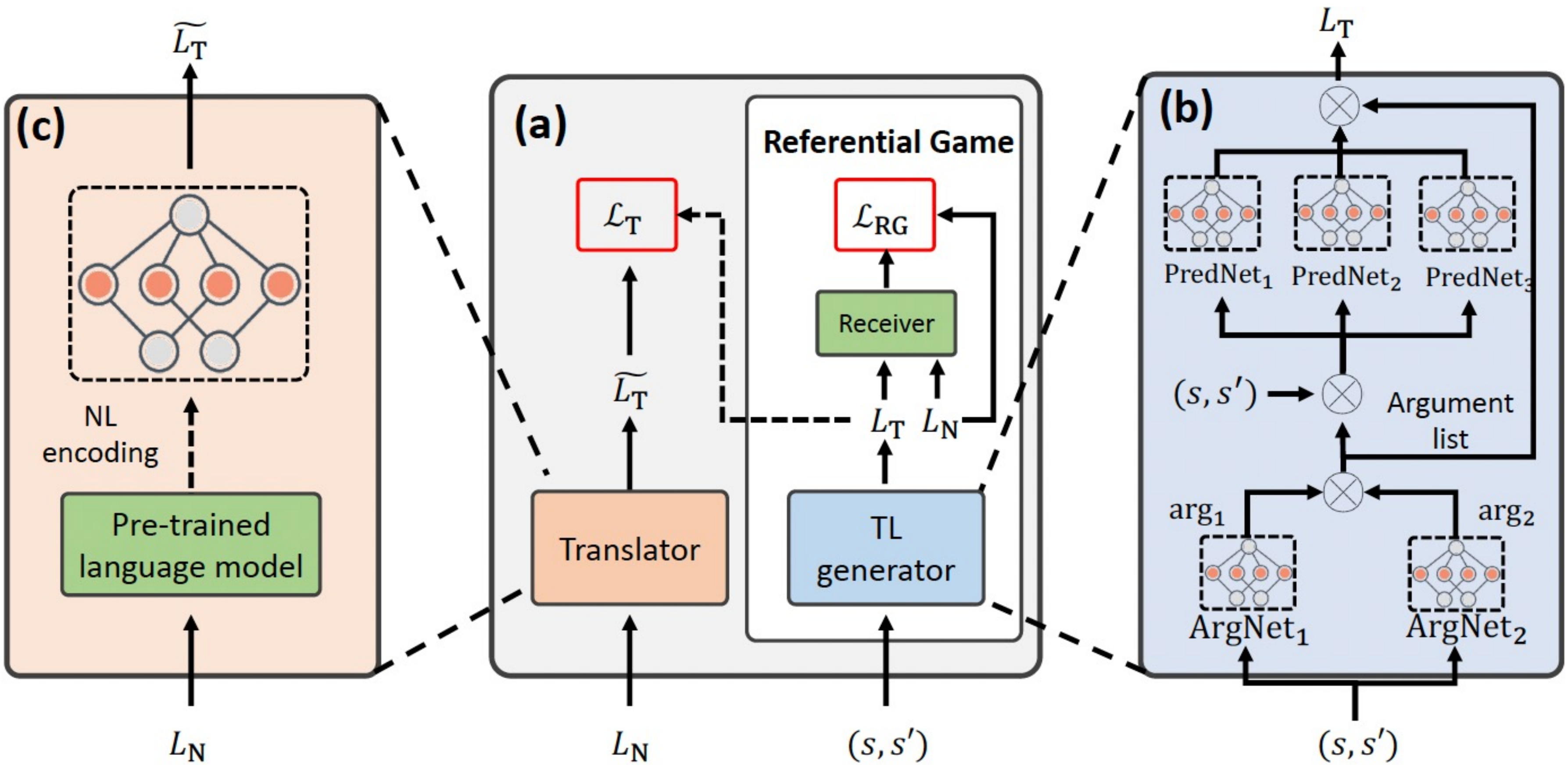
$$\mathcal{L}_{\mathrm{T}}(\phi) = \mathop{\mathbb{E}}_{(s,s',L_{\mathrm{N}})\sim\mathcal{D}} \left[-\log p_{\phi}(g_{\theta}(s,s')|b(L_{\mathrm{N}})) \right]$$

Method

Policy learning

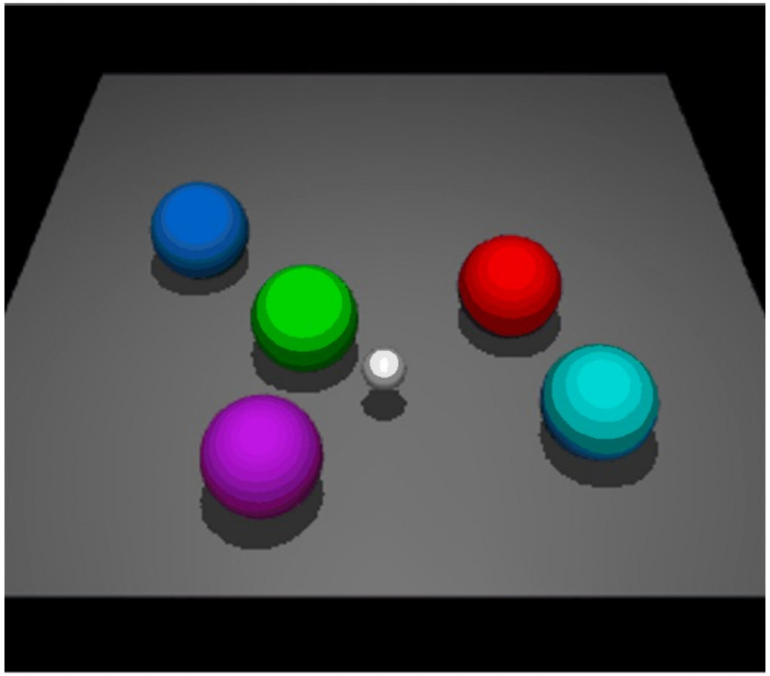


Method


(c)
$\widetilde{L_T}$
NL
encoding
Pre-trained
language model
L_N
(a)
Referential Game
$\mathcal{L}_T$
$\mathcal{L}_{RG}$
Receiver
$\widetilde{L_T}$
L_T
L_N
Translator
TL
generator
L_N
(s, s')
(b)
L_T
PredNet_1
PredNet_2
PredNet_3
(s, s')
Argument
list
arg_1
arg_2
ArgNet_1
ArgNet_2
(s, s')

Experiments

(a) FrankaKitchen



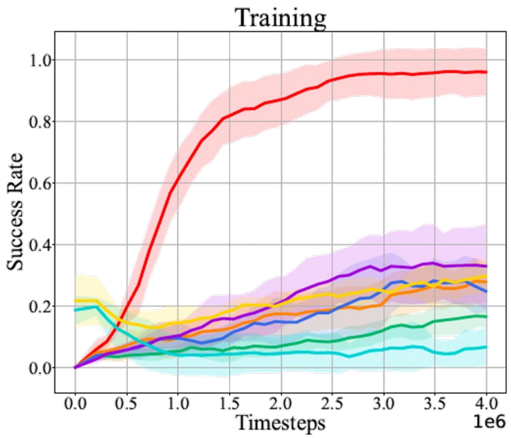
(b) CLEVR-Robot

Experiments

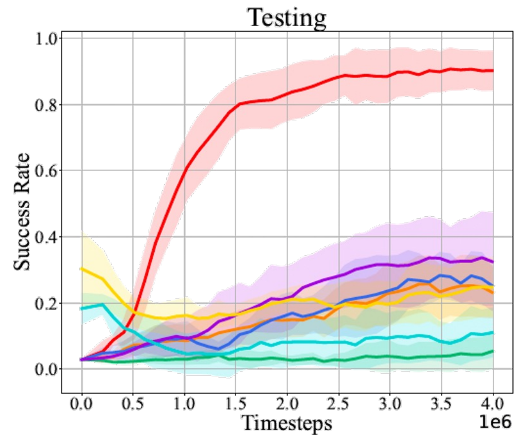
Examples of instructions:

--- Training instructions ---
Please open the door of the microwave.
Can you open the microwave door for me?
I would appreciate it if you could open the microwave door.
Can you open the microwave door at your earliest convenience?
Would you be able to open the door of the microwave?
I need the door of the microwave to be opened, please.
--- Testing instructions ---
Twist the oven knob to turn on the top burner.
Manipulate the oven knob to activate the top burner.
Turn the oven knob to the right to activate the top burner

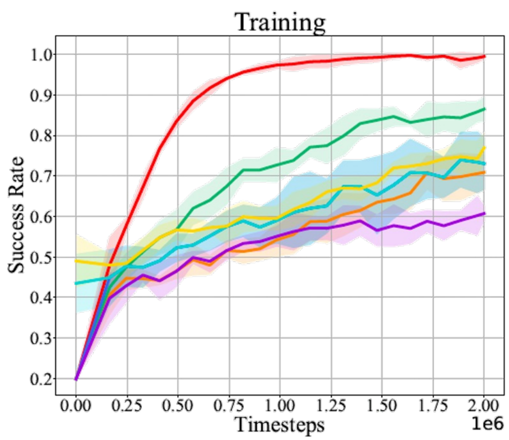
Main results



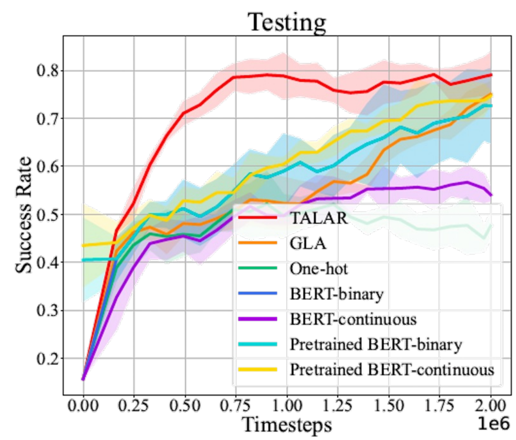
FrankaKitchen-Train



FrankaKitchen-Test



CLEVR-Robot-Train



CLEVR-Robot-Test

Experiments

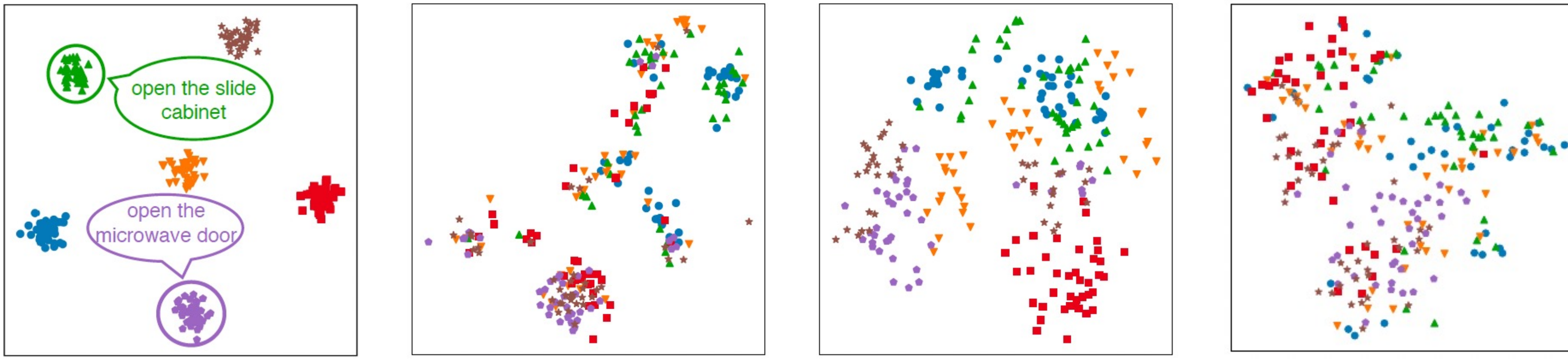
Can you open the microwave door for me?

Baseline

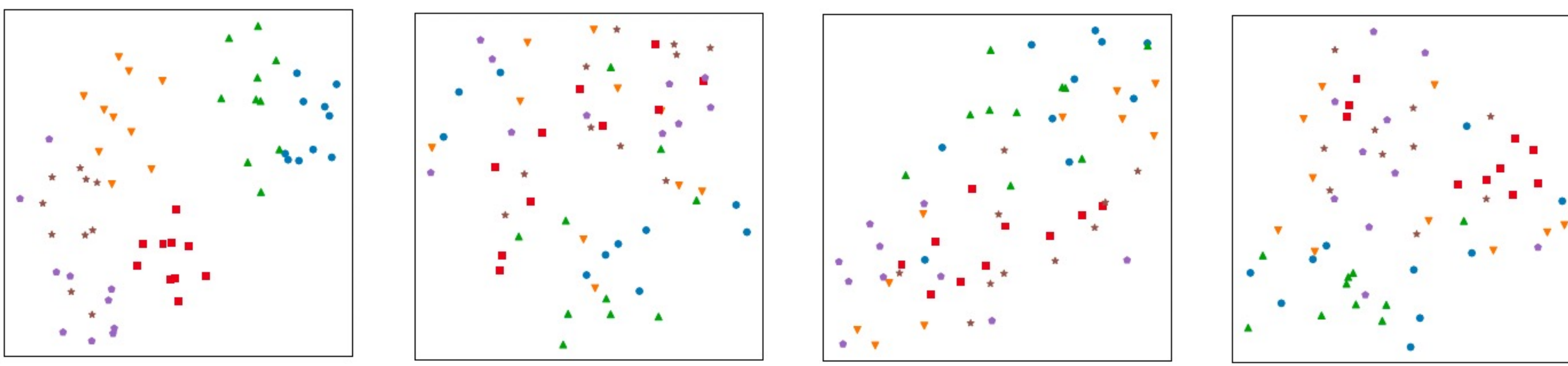
TALAR

Experiments

Learned natural language representation:



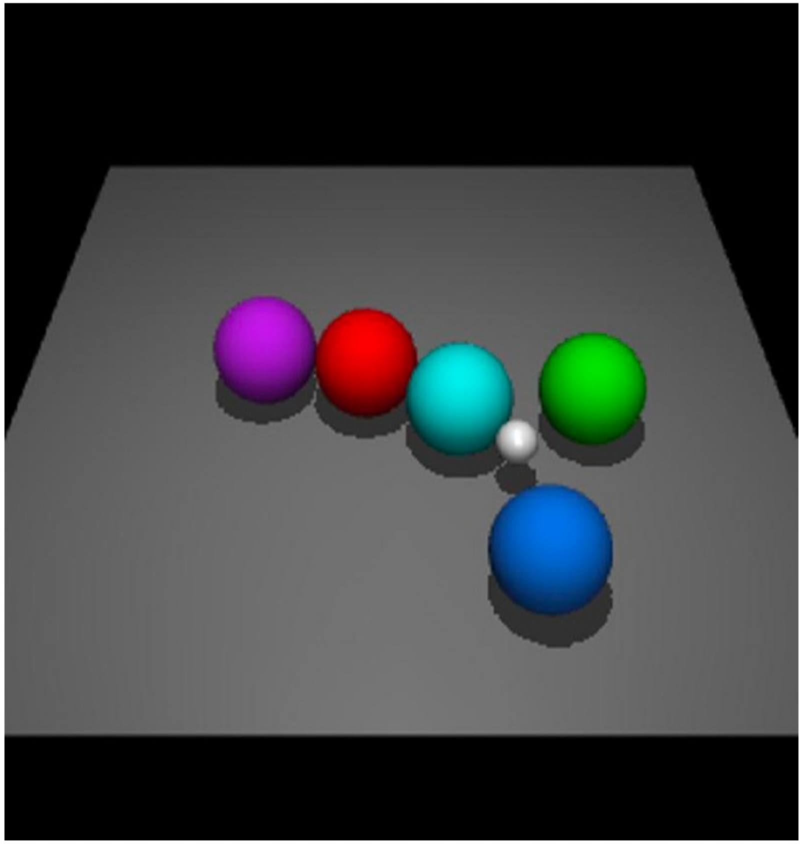
(a) TALAR (b) GLA (c) BERT (d) OIL baseline



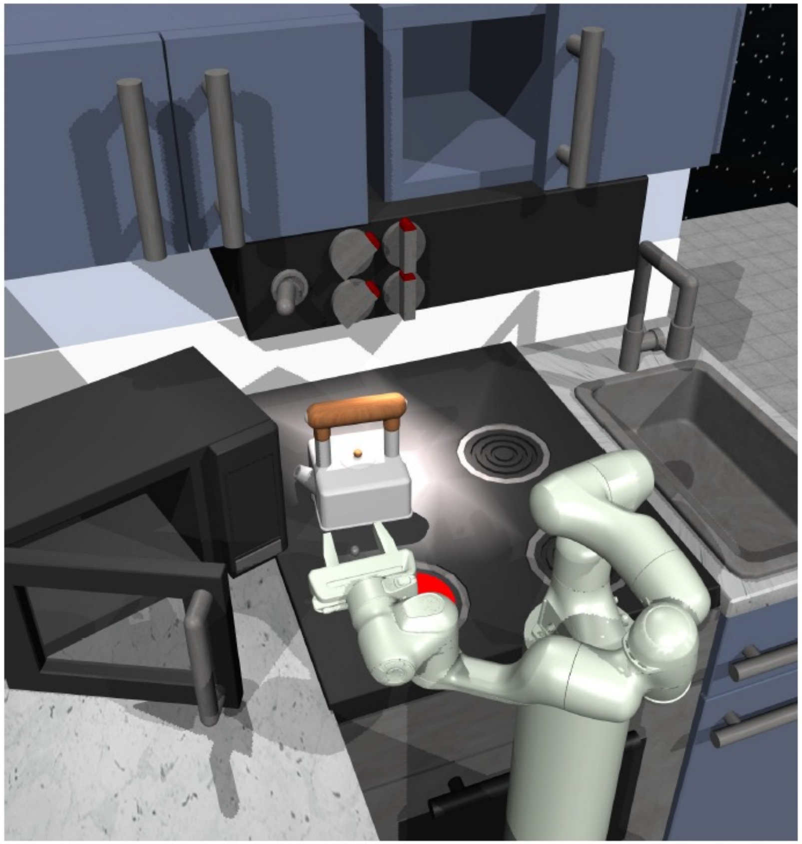
Experiments

Learning And Mining from DatA
http://www.lamda.nju.edu.cn

Learning task language as a high-level abstraction



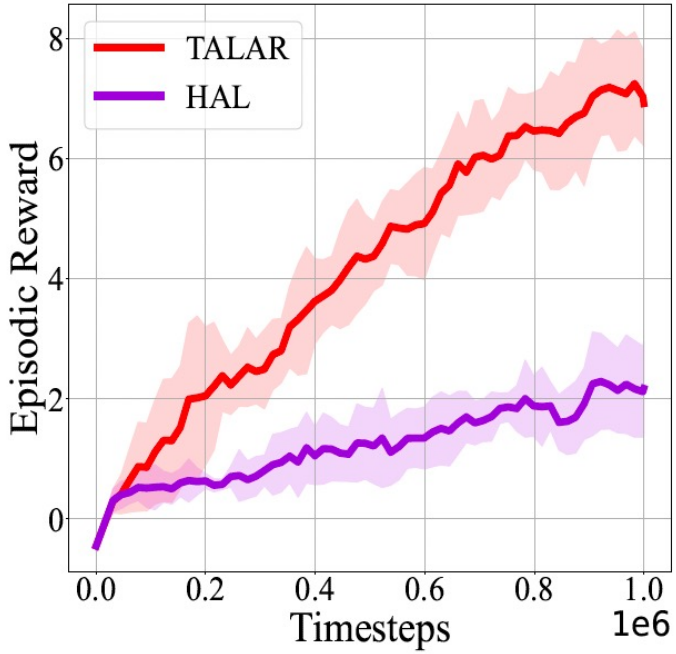
(a) CLEVR-Robot



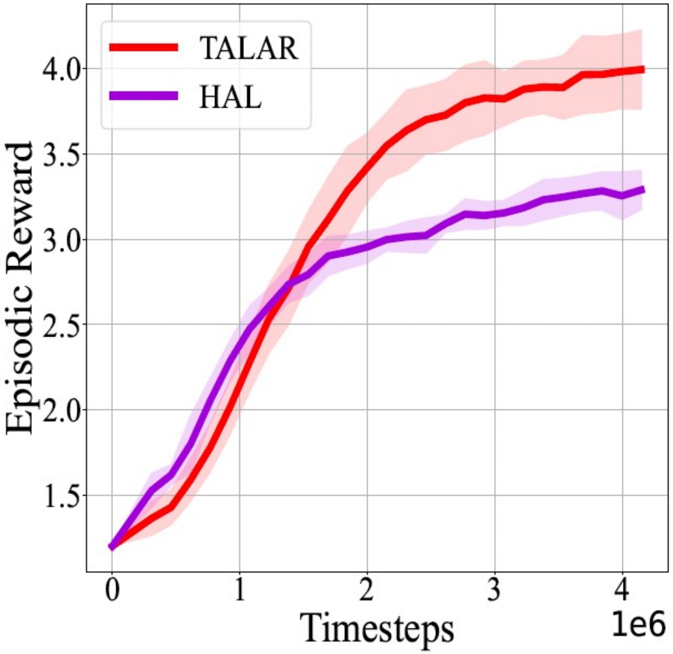
(b) FrankaKitchen

Experiments

Learning task language as a high-level abstraction



(a) CLEVR-Robot



(b) FrankaKitchen

Takeaways

1. The importance of natural language representation in the development of instruction-following agents, as it significantly impacts their performance and effectiveness.

2. The necessity of generating unique learned representations to enhance the agent's ability to understand and execute tasks.

3. The need to investigate the instruction-following in open environments, broadening the scope of the agent's capabilities and promoting its adaptability to diverse situations.