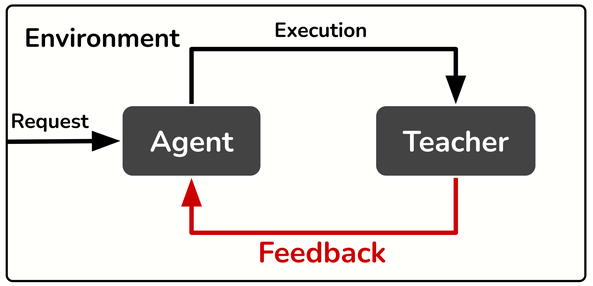
Interactive Learning from Activity Description

Khanh Nguyen[1], Dipendra Misra[2], Robert Schapire[2], Miro Dudík[2], and Patrick Shafto[3]

[1] University of Maryland, College Park [2] Microsoft Research, New York [3] Rutgers University, Newark

ILIAD: A Verbal Interactive Learning Protocol



Imitation learning

Feedback = Demonstration

Teacher demonstrates good actions that lie in agent's action space

☹ **Teacher has to learn to control agent ⇒ costly for non-experts to provide!**

Reinforcement learning

Feedback = Scalar reward

Teacher evaluates agent's performance by a floating-point number

☹ **Scalar rewards convey weak learning signals ⇒ sample-inefficiency!**

ILIAD (this work)

Feedback = Language description

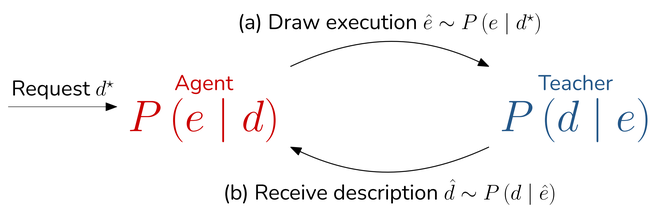
Teacher verbally describes agent's activities

☺ **Richer learning signals than reward**

☺ **Not requiring agent-specific knowledge to provide like demonstration**

ILIAD can offer ***complementary*** advantages compared to non-verbal protocols like imitation learning and reinforcement learning.

ADEL: A Practical Implementation of ILIAD



Sampling executions from a ***mixture*** of policies:

$$P(e \mid d) = \underbrace{\lambda P_{\pi_\omega}(e)}_{\text{accelerate learning}} + \underbrace{(1-\lambda) P_{\pi_\theta}(e \mid d^\star)}_{\text{ensure convergence}}$$

where P_π is execution distribution induced by policy π

π_ω is ***request-agnostic*** policy learned from unlabeled executions

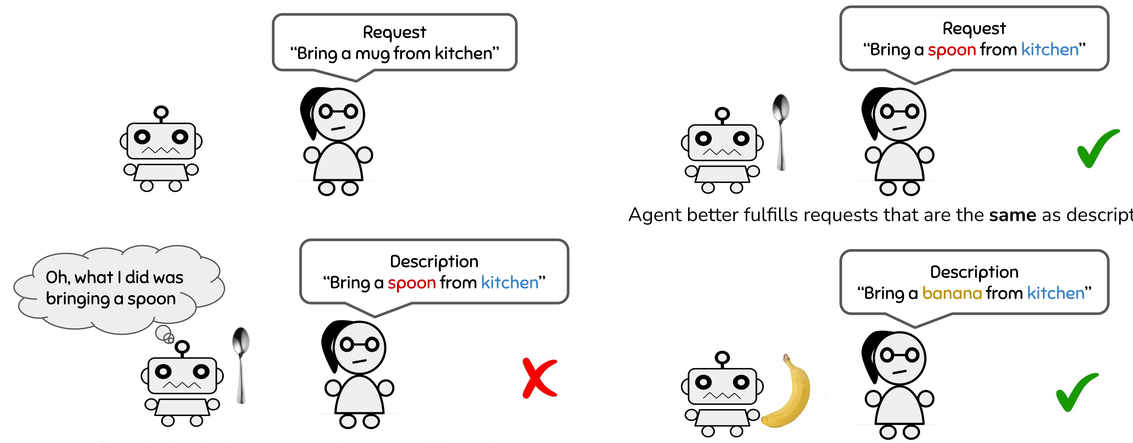
π_θ is ***request-guided*** policy of the agent (to be used at test time)

Grounding description language to executions:

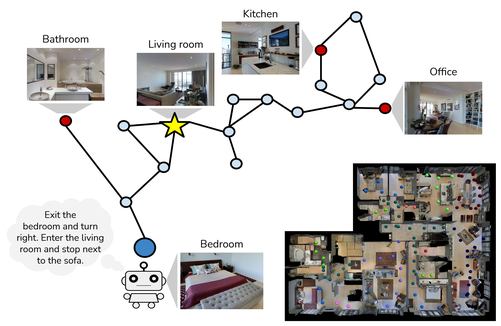
$$\theta_{\text{new}} = \arg\max_\theta \sum_{(\hat{e},\hat{d}) \in D} \sum_{(s,a_s) \in \hat{e}} \log \pi_\theta\left(a_s \mid s, \hat{d}\right)$$

Agent is trained to (re)generate executions conditioned on descriptions.

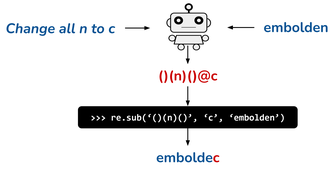
Motivational Example



Tasks



Task 1: Following Navigational Language Instructions in Photo-Realistic Environments



Task 2: Generating Regular Expressions from Word-Modifying Language Requests

Experiments

Results

Algorithm	Test success rate (%) ↑	Sample complexity ↓
Vision-language navigation (NAV)		
Imitation learning	32.0 ± 1.63	45K ± 26K
Reinforcement learning	20.5 ± 0.58	$+\infty$
ADEL (ours)	31.9 ± 0.76	406K ± 31K
Word modification (REGEX)		
Imitation learning	93.0 ± 0.37	118K ± 16K
Reinforcement learning	0.0 ± 0.00	$+\infty$
ADEL (ours)	89.0 ± 1.30	573K ± 116K

Table: Results on test set. Sample complexity is the number of training episodes (or number of teacher responses) required to reach a validation success rate of at least c (30% on NAV and 85% on REGEX).

Mixing rate	Val success rate (%) ↑	Sample complexity ↓
Vision-language navigation		
$\lambda = 1$	29.4	$+\infty$
$\lambda = 0$	0.0	$+\infty$
$\lambda = 0.5$ (final)	32.0	384K
Word modification		
$\lambda = 1$	55.7	$+\infty$
$\lambda = 0$	0.2	$+\infty$
$\lambda = 0.5$ (final)	88.0	608K

Table: Effects of mixing execution policies in ADEL.

ADEL is more **sample-efficient** than RL baselines, while achieving **competitive** success rates with IL baselines (without requiring feedback provides to have agent-specific expertise).