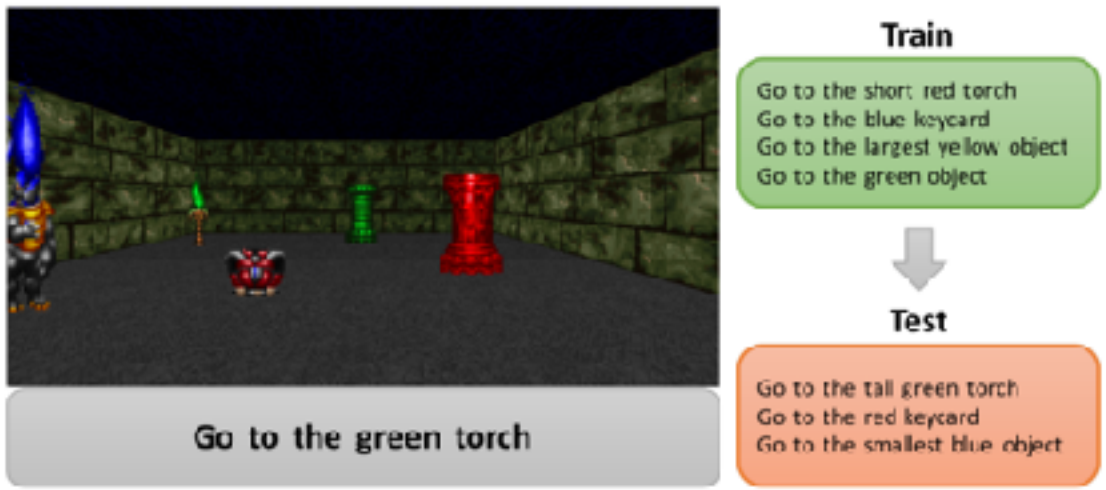
Scheduled Policy Optimization for Natural Language Communication with Intelligent Agents

Wenhan Xiong, Xiaoxiao Guo, Mo Yu, Shiyu Chang, Bowen Zhou, William Wang
UCSB   IBM Research   JD.com

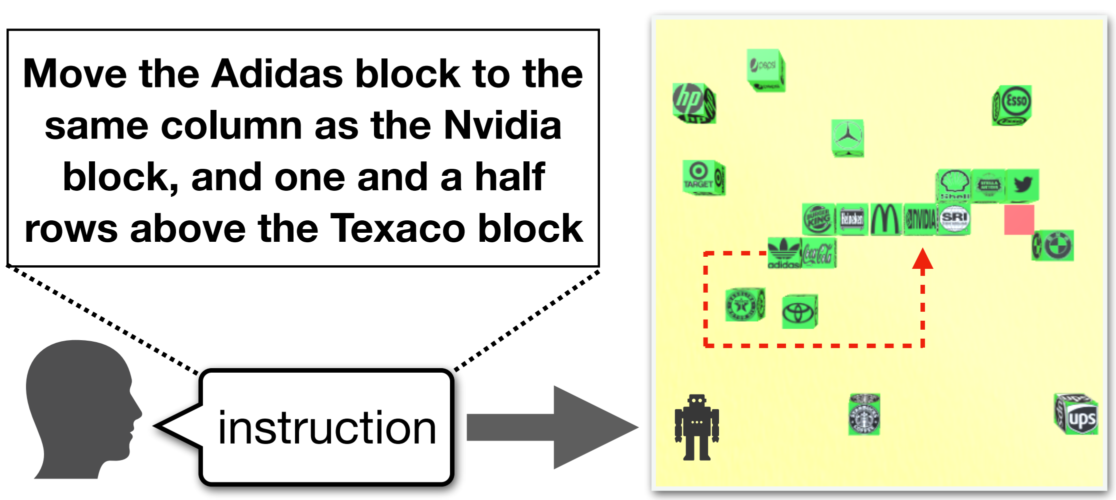
Train an agent that is able to …

understand human language instructions, *explore* the working environment and ***accomplish*** a specific task

Task-Oriented Language Grounding



(a) Chaplot et al. AAAI'18

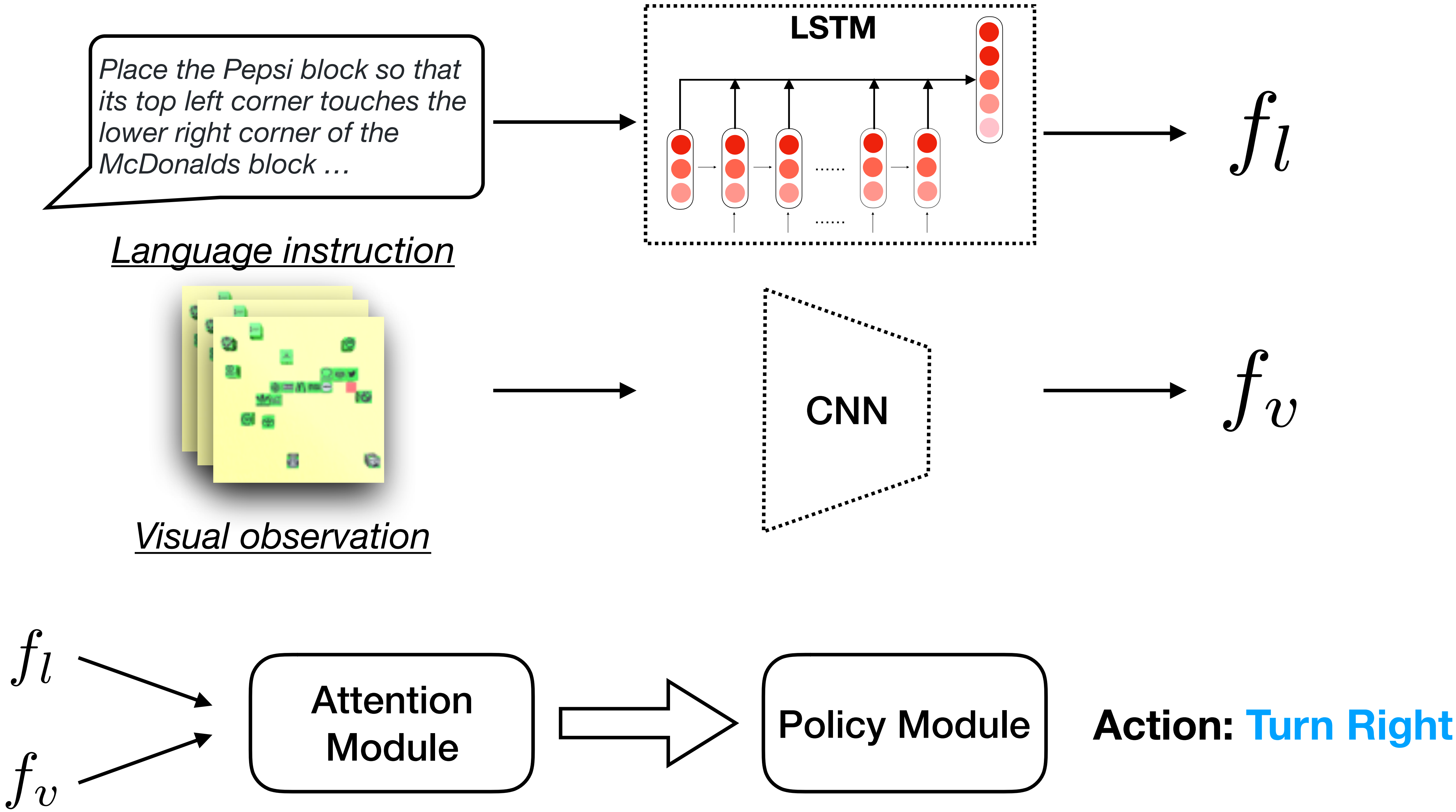


(b) Misra et al. EMNLP'17

Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at pictures and table. Wait by the moose antlers hanging on the wall.

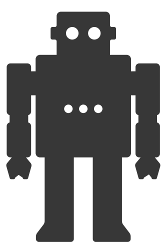
(c) Anderson et al. CVPR'18

Learning to *understand*: Model Design



Learning to *explore*: RL

Reinforcement Learning:



- Use a ***parametrized stochastic*** policy to explore
- Improve the policy by learning from rewards

Problem:

- Rewards can be sparse, large action space —
  slow training

Learning to *explore*: Demonstration + RL

Use human demonstration to guide the RL agent

If we have sufficient demonstration data,
then we can do supervised learning:
learn a model that maps states to demonstration actions

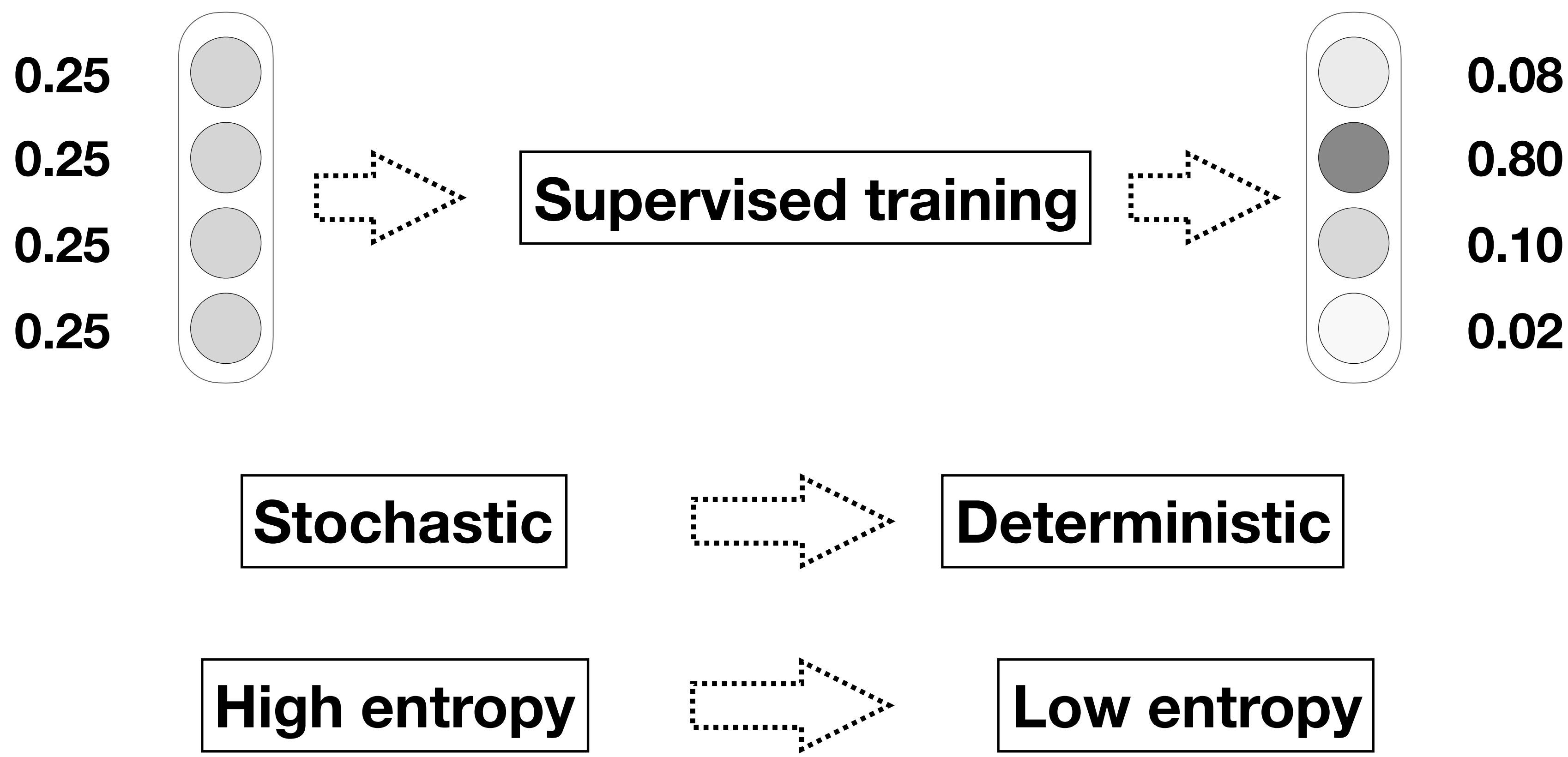
In practice, it is hard to obtain enough human demonstration

Insufficient demonstation ⇨ ***cannot perform well on unseen environment***

SL for initialization (accelerate training)
+
RL for exploration (better generalization) **?**

Learning to *explore*: Demonstration + RL

Effect of Supervised Learning using Demonstration:



RL agent explores the state-action space by sampling actions from this policy

Policy Entropy Evolution

Training steps

Scheduled Policy Optimization

Idea:

- Let the agent starts with RL instead of SL
- The agent calls for a demonstration when needed
- Keep track of the performance during training

$$b = \text{average}(\mathcal{H}) + \lambda\sigma_c$$

If the agent performs worse than baseline, fetch one demonstration

Challenge: **REINFORCE (**William'1992**) is highly unstable, hard to get a useful baseline**

Proximal Policy Optimization

Schulman et al. 2017 ArXiv

Use constrained policy gradient for more stable update

Proximal policy optimization:

$$\mathcal{J}^{PPO}(\theta) = \mathbb{E}\Big[\min\big(\rho_t(\theta)A_t, [\rho_t(\theta)]_{1-\epsilon}^{1+\epsilon}A_t\big)\Big]$$

$$\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

constraining the difference of the updated policy and old policy

Policy Entropy Evolution

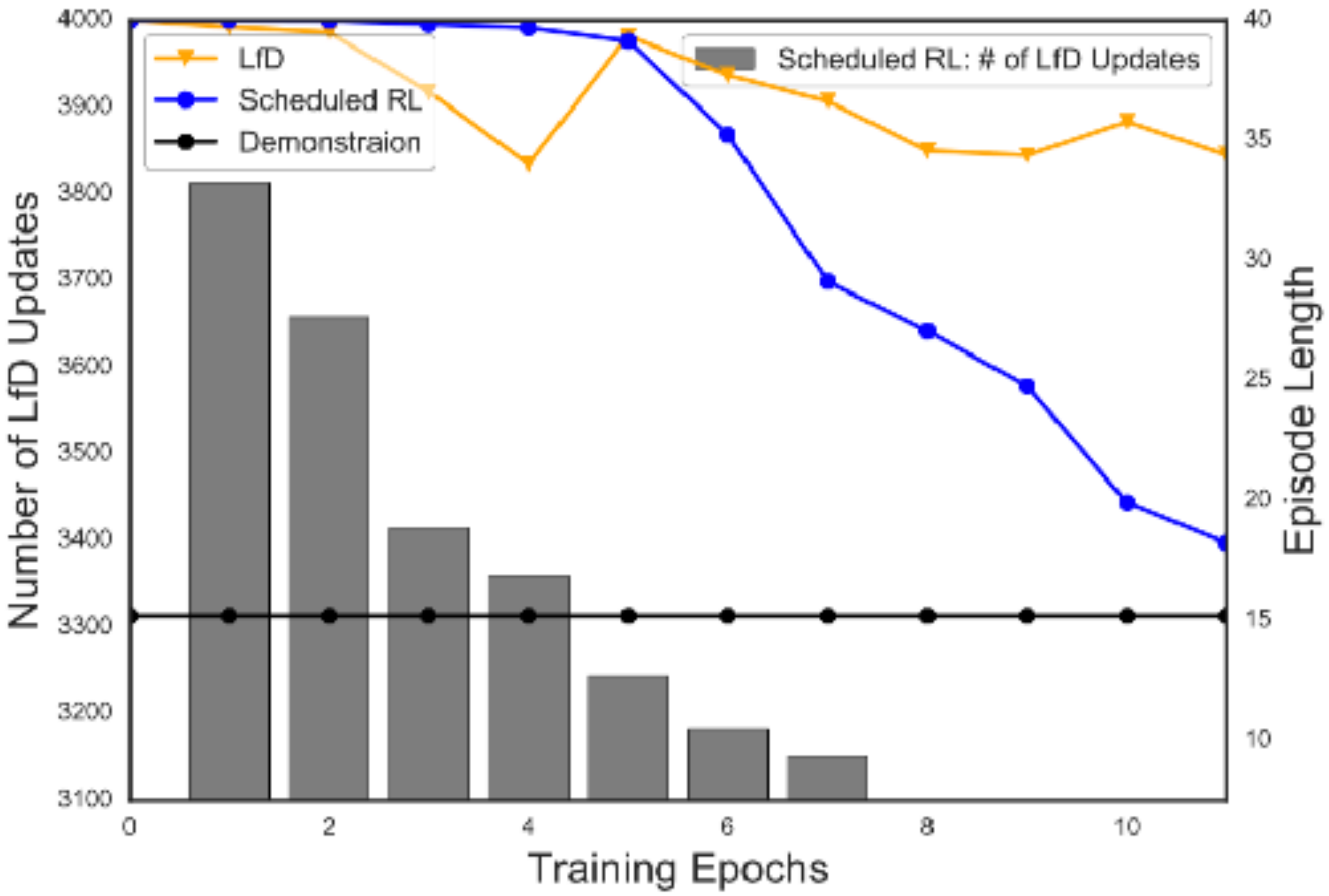
Upper: Scheduled RL; **Down:** RL

Results on Block-world

Misra et al. EMNLP'17

Methods	Dev Error Mean	Dev Error Med.	Test Error Mean	Test Error Med.
HUMAN	0.35	0.30	0.37	0.31
INITIAL	5.95	5.71	6.23	6.12
RANDOM	15.3	15.70	15.11	15.35
Misra el al.				
Ensem-LfD	4.64	4.27	4.95	4.53
Ensem-DQN	5.85	5.59	6.15	5.97
Ensem-REIN	5.28	5.23	5.69	5.57
Ensem-BEST	3.59	3.03	3.78	3.14
Our Models				
S-REIN	2.94	2.23	2.95	2.21
S-A2C	2.79	2.21	2.75	2.18
S-PPO	**1.69**	**0.99**	**1.71**	**1.04**

Distance Error

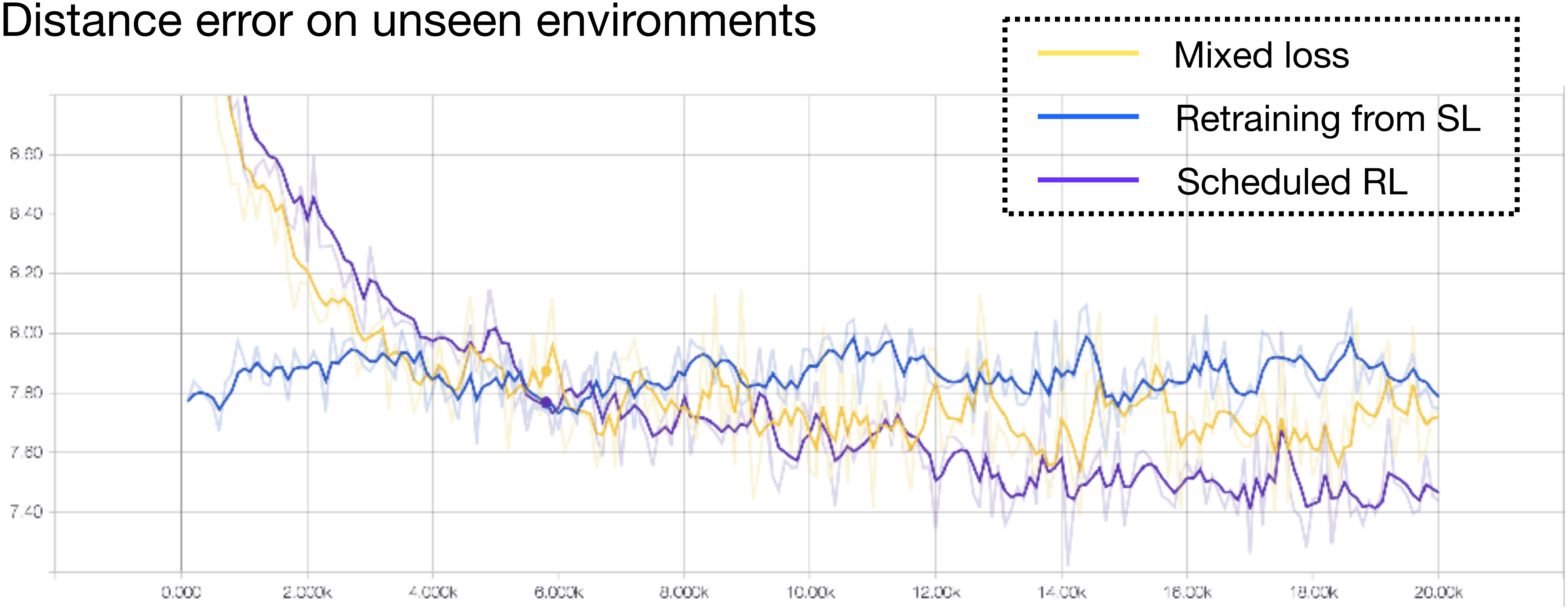


# of Demonstration Calls

Results on Vision-Language-Navigation

Peterson et al. CVPR'18

Distance error on unseen environments



Training steps

Summary

- Empirical analysis on the policy entropy evolution
- A novel scheduled mechanism that makes better use of limited demonstration data
- Achieve the best performance on Block-World

Thank you!

Code will be released at: https://github.com/xwhan

xwhan@cs.ucsb.edu