TOWARDS TEACHING MACHINES WITH LANGUAGE: INTERACTIVE LEARNING FROM ONLY LANGUAGE DESCRIPTIONS OF ACTIVITIES

Khanh Nguyen UMD*
Dipendra Misra, Robert Schapire, Miro Dudík Microsoft Research
Patrick Shafto Rutgers University

ABSTRACT

We present a novel interactive learning protocol that enables training request-fulfilling agents by only verbally describing their activities. Our protocol gives rise to a new family of interactive learning algorithms that offer complementary advantages against traditional algorithms like imitation learning (IL) and reinforcement learning (RL). We develop an algorithm that practically implements this protocol and employ it to train agents in two challenging request-fulfilling problems using purely language-description feedback. Empirical results demonstrate the strengths of our algorithm: compared to RL baselines, it is more sample-efficient; compared to IL baselines, it achieves competitive success rates while not requiring feedback providers to have agent-specific expertise. We also provide theoretical guarantees of the algorithm under certain assumptions on the teacher and the environment.

The goal of a *request-fulfilling* agent is to map a given language request in a situated environment to an execution that accomplishes the intent of the request (Winograd, 1972; Chen & Mooney, 2011; Tellex et al., 2012; Artzi et al., 2013; Misra et al., 2017; Anderson et al., 2018; Chen et al., 2019; Nguyen et al., 2019; Nguyen & Daumé III, 2019; Chen et al., 2020; 2021; Shridhar et al., 2020). Developing request-fulfilling agents is an important step towards creating autonomous assistants that communicate with humans naturally. Request-fulfilling agents have been typically trained using *non-verbal* interactive learning protocols such as imitation learning (IL) which assumes labeled executions as feedback (Mei et al., 2016; Anderson et al., 2018), or reinforcement learning (RL) which uses scalar rewards as feedback (Chaplot et al., 2018; Hermann et al., 2017; Zhu et al., 2017). We introduce a new interactive learning protocol for training these agents called ILIAD: **I**nteractive **L**earn**I**ng from **A**ctivity **D**escription, where feedback is *limited to language descriptions of executions*.

Algorithm 1 presents the ILIAD protocol. Learning proceeds in episodes of interaction between an agent policy and a teacher in an environment with state space $\mathcal{S}$, action space $\mathcal{A}$, and transition function $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ denotes the space of all probability distributions over $\mathcal{S}$. Let $\mathcal{R} = \{R : \mathcal{S} \times \mathcal{A} \to [0, 1]\}$ be a set of reward functions. Each episode starts with the agent being presented with a *task* q sampled from a distribution $\mathbb{P}^\star(q)$. Formally, a task q is defined by a tuple $(R, s_1, d^\star)$ of reward function $R \in \mathcal{R}$, start state $s_1 \in \mathcal{S}$, and a language request $d^\star \in \mathcal{D}$, where $\mathcal{D}$ is the set of all nonempty strings generated from a finite vocabulary. The agent starts in s_1 and is presented with $d^\star$ but *does not* observe R or any rewards generated by it. Intuitively, the request $d^\star$ verbally communicates the optimal behavior for the reward function R to the agent. For example, in robot navigation, a request $d^\star$ = "*go to the kitchen*" specifies a task given by a reward function that is maximized when the robot is in the kitchen. To make decisions, the agent maintains a *request-conditioned policy* $\pi_\theta : \mathcal{S} \times \mathcal{D} \to \Delta(\mathcal{A})$ with parameters θ, which takes in a state $s \in \mathcal{S}$ and a request $d \in \mathcal{D}$, and outputs a probability distribution over $\mathcal{A}$. Using this policy, it can generate an *execution* $\hat{e} = (s_1, \hat{a}_1, s_2, \cdots, s_H, \hat{a}_H)$, where H is the task horizon (the time limit), $\hat{a}_i \sim \pi_\theta(\cdot \mid s_i, d)$ and $s_{i+1} \sim T(\cdot \mid s_i, \hat{a}_i)$ for every i. We use the notation $e \sim \mathbb{P}_\pi(\cdot \mid s_1, d)$ to denote sampling an execution e by following policy π given a start state s_1 and a request d.

The feedback mechanism in ILIAD is provided by a *teacher* that can describe executions in language. The teacher is modeled by a fixed distribution $\mathbb{P}_T : (\mathcal{S} \times \mathcal{A})^H \to \Delta(\mathcal{D})$, where $(\mathcal{S} \times \mathcal{A})^H$ is the space over H-step executions. The teacher enables language understanding by providing the agent

*Correspondence email: kxnguyen@umd.edu

Algorithm 1 ILIAD protocol. Details of line 4 and line 6 are left to specific implementations.

1: Initialize agent policy $\pi_\theta : \mathcal{S} \times \mathcal{D} \to \Delta(\mathcal{A})$
2: **for** $n = 1, 2, \cdots, N$ **do**
3: World samples $q = (R, d^\star, s_1) \sim \mathbb{P}^\star(\cdot)$ (reward function R is *not* revealed to agent)
4: Agent generates execution $\hat{e}$ given π_θ, $d^\star$, and s_1
5: Teacher generates description $\hat{d} \sim \mathbb{P}_T(\cdot \mid \hat{e})$
6: Agent uses $\left(d^\star, \hat{e}, \hat{d}\right)$ to update π_θ
return π_θ

Algorithm 2 ADEL: our implementation of the ILIAD protocol. $\mathbb{P}_T(d \mid e)$ is the teacher model.

1: **Input**: approximate marginal $\mathbb{P}_{\pi_\omega}(e \mid s_1)$, mixing rate $\lambda \in [0, 1]$, annealing rate $\beta \in (0, 1)$
2: Initialize $\pi_\theta : \mathcal{S} \times \mathcal{D} \to \Delta(\mathcal{A})$ and $\mathcal{B} = \emptyset$
3: **for** $n = 1, 2, \cdots, N$ **do**
4: World samples task $q = (R, d^\star, s_1) \sim \mathbb{P}^\star(\cdot)$
5: Agent generates $\hat{e} \sim \lambda \mathbb{P}_{\pi_\omega}(\cdot \mid s_1) + (1 - \lambda)\mathbb{P}_{\pi_\theta}(\cdot \mid s_1, d^\star)$
6: Teacher generates description $\hat{d} \sim \mathbb{P}_T(\cdot \mid \hat{e})$. Add datapoint: $\mathcal{B} \leftarrow \mathcal{B} \cup \left(\hat{e}, \hat{d}\right)$
7: Update agent policy: $\theta \leftarrow \max_{\theta'} \sum_{(\hat{e},\hat{d}) \in \mathcal{B}} \sum_{(s,a_s) \in \hat{e}} \log \pi_{\theta'}(a_s \mid s, \hat{d})$ where a_s is the action taken by the agent in state s. Anneal the mixing rate: $\lambda \leftarrow \lambda \cdot \beta$.
return π_θ

with a *language description of the agent's execution* $\hat{d} \sim \mathbb{P}_T(\cdot \mid \hat{e})$. We assume that the descriptions are specified in the same language that is used to specify the requests. Hence, by grounding the descriptions to the corresponding executions, the agent can acquire knowledge about the description language and thus can gradually improve its request-fulfilling capability. Crucially, the agent receives *no other* feedback such as ground-truth demonstration (Mei et al., 2016), scalar reward (Hermann et al., 2017), or constraint (Miryoosefi et al., 2019). At test time, the teacher is not present and the agent must execute requests autonomously. The objective of the agent is to find a policy π with maximum value, where we define the policy value $V(\pi)$ as:

$$V(\pi) = \mathbb{E}_{q \sim \mathbb{P}^\star(\cdot), \hat{e} \sim \mathbb{P}_\pi(\cdot \mid s_1, d^\star)}\left[\sum_{i=1}^{H} R\left(s_i, \hat{a}_i\right)\right] \tag{1}$$

To formulate the learning problem in ILIAD, we define a joint distribution over tasks and executions:

$$\mathbb{P}^\star\left(e, R, s_1, d\right) = \mathbb{P}_{\pi^\star}\left(e \mid s_1, d\right)\mathbb{P}^\star\left(R, d, s_1\right) \tag{2}$$

where $\pi^\star$ be the optimal policy that maximizes $V(\pi)$. We then implement the ILIAD protocol by reducing it to a *density-estimation* problem: given that we can effectively draw samples from the marginal $\mathbb{P}^\star(s_1, d)$ and an approximately *consistent* teacher $\mathbb{P}_T(d \mid e) \approx \mathbb{P}^\star(d \mid e)$, how do we learn a policy π_θ such that $\mathbb{P}_{\pi_\theta}(e \mid s_1, d)$ is close to $\mathbb{P}^\star(e \mid s_1, d)$? Here, the marginal $\mathbb{P}^\star(s_1, d)$, the consistent teacher $\mathbb{P}^\star(d \mid e)$, and the (ground-truth) execution distribution $\mathbb{P}^\star(e \mid s_1, d) = \mathbb{P}_{\pi^\star}(e \mid s_1, d)$ are obtained from the joint distribution defined in Equation 2.

We develop an algorithm named ADEL: **A**ctivity-**D**escription **E**xplorative **L**earner (Algorithm 2) that offers practical solutions to this problem. ADEL implements a semi-supervised sampling scheme that efficiently explores the execution space. Specifically, in the algorithm, the agent generates executions from a mixture distribution that combines a request-agnostic execution distribution $\mathbb{P}_{\pi_\omega}(e \mid s_1)$, which can be learned from unlabeled executions, and a request-guided execution distribution $\mathbb{P}_{\pi_\theta}(\cdot \mid s_1, d^\star)$ (line 5). The agent then employs behavior cloning (Pomerleau, 1991) to ground descriptions to executions (line 7). We theoretically prove convergence for a variant of ADEL in the contextual bandit setting (Langford & Zhang, 2008).

Our paper does *not* argue for the primacy of ILIAD over other protocols like RL or IL. In fact, an important point we raise is that there are multiple, possibly competing metrics for comparing learning protocols. We focus on the trade-off between the learning effort of the agent and the teacher in each protocol (Table 1). In all protocols, the agent and the teacher establish a communication channel that

Table 1: Trade-offs between the learning effort of the agent and the teacher in learning protocols. Each protocol employs a different medium for the teacher to convey feedback. If a medium is not natural to the teacher (e.g. IL-style demonstration), it must learn to encode feedback intent using that medium (*teacher communication-learning effort*). Similarly, if a medium is not natural to the agent (e.g. human language), it needs to learn to interpret feedback (*agent communication-learning effort*). The agent also learns tasks from information decoded from feedback (*agent task-learning effort*). The qualitative claims on the "agent learning effort" column summarize our empirical findings on the learning efficiency (measured by sample complexity) of these protocols.

		Learning effort	
Protocol	Feedback medium	Teacher (communication learning)	Agent (communication & task learning)
IL	Demonstration	Highest	Lowest
RL	Scalar reward	None	Highest
ILIAD	Language description	None	Medium

Table 2: Main results. We report means and standard deviations of success rates (%) over five runs with different random seeds. RL-Binary and RL-Cont refer to the RL settings with binary and continuous rewards, respectively. Sample complexity is the number of training episodes (or number of teacher responses) required to reach a validation success rate of at least c.

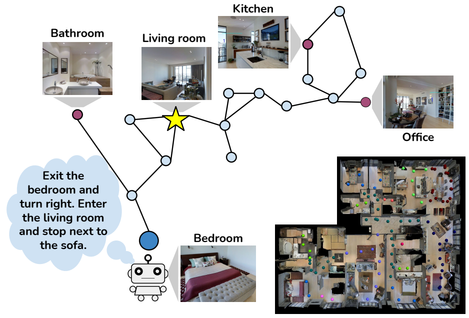
				Sample complexity ↓		
Learning setting	Algorithm	Val success rate (%) ↑	Test success rate (%) ↑	# Demonstrations	# Rewards	# Descriptions
Vision-language navigation					($c = 30.0\%$)	
IL	DAgger	35.6 ± 1.35	32.0 ± 1.63	45K ± 26K	-	-
RL-Binary	REINFORCE	22.4 ± 1.15	20.5 ± 0.58	-	$+\infty$	-
RL-Cont	REINFORCE	11.1 ± 2.19	11.3 ± 1.25	-	$+\infty$	-
ILIAD	ADEL	32.2 ± 0.97	31.9 ± 0.76	-	-	406K ± 31K
Word modification					($c = 85.0\%$)	
IL	DAgger	92.5 ± 0.53	93.0 ± 0.37	118K ± 16K	-	-
RL-Binary	REINFORCE	0.0 ± 0.00	0.0 ± 0.00	-	$+\infty$	-
RL-Cont	REINFORCE	0.0 ± 0.00	0.0 ± 0.00	-	$+\infty$	-
ILIAD	ADEL	88.1 ± 1.60	89.0 ± 1.30	-	-	573K ± 116K

allows the teacher to encode feedback and send it to the agent, who learns tasks based on information decoded from feedback. At one extreme, IL places the burden of establishing the communication channel entirely on the teacher. To provide a demonstration, the teacher in IL must learn to control the agent to accomplish tasks by specifying actions that lie in the agent's action space.[1] To compensate for this effort, the agent usually learns very efficiently with IL because it does not have to learn to interpret feedback, and the feedback directly specifies desired behavior. At another extreme, we have RL and ILIAD, where the teacher provides feedback via agent-agnostic media (reward and language, respectively). RL eliminates the agent communication-learning effort by hard-coding the semantics of scalar rewards into the learning algorithm.[2] But the trade-off of using such limited feedback is that the effort required by the agent to learn the task increases. State-of-the-art RL algorithms are notorious for their high sample complexity, making them expensive to use outside simulators (Hermann et al., 2017; Chaplot et al., 2018; Chevalier-Boisvert et al., 2019). ILIAD offers a compromise between RL and IL: it can be more sample-efficient than RL while not requiring the teacher to master the agent's control interface. Overall, no protocol is superior in all metrics and the choice of protocol depends on users' preferences.

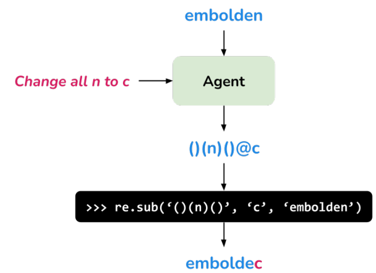
We empirically evaluate ADEL against IL and RL baselines on two tasks: vision-language navigation (Anderson et al., 2018), and word-modification via regular expressions (Andreas et al., 2018).

[1]Third-person or observational IL (Stadie et al., 2017; Sun et al., 2019) allows the teacher to demonstrate tasks with their action space. However, this framework is *non-interactive* because the agent imitates pre-collected demonstrations and does not interact with a teacher. We consider interactive IL (Ross et al., 2011), which is shown to be more effective than non-interactive counterparts.

[2]By design, RL algorithms understand that higher reward value implies better performance.



(a) *Vision-language navigation* (NAV): a (robot) agent fulfills a navigational natural-language request in a photo-realistic simulated house. Locations in the house are connected as a graph. In each time step, the agent receives a photo of the panoramic view at its current location (due to space limit, here we only show part of a view). Given the view and the language request, the agent chooses an adjacent location to go to.



(b) *Word modification via regular expressions* (REGEX): an agent is given an input word and a natural-language request that asks it to modify the word. The agent outputs a regular expression that follows our specific syntax. The regular expression is executed by the Python's `re.sub()` method to generate an output word.

Figure 1: Illustrations of the two request-fulfilling problems that we conduct experiments on.

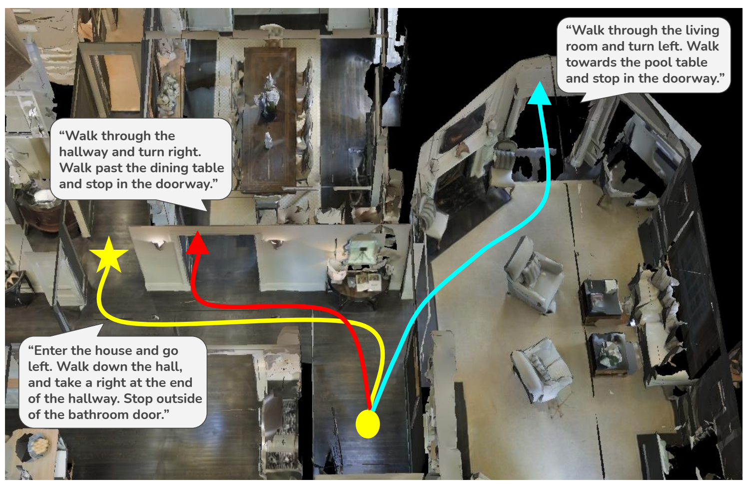


Figure 2: A qualitative example of training an agent to fulfill a navigation request in 3D environments (Anderson et al., 2018) using ADEL. The agent receives a request *"Enter the house..."* which implies the path. Initially, it does not understand language and thus wanders far from the goal. Its execution (the path) is described as *"Walk through the living room..."*. To ground the description language, the agent learns to generate the path conditioned on the description. After a number of interactions, its execution () is closer to the optimal path. As this process iterates, the agent gradually improves its understanding of the description language and thus also executes requests more precisely.

Figure 2 illustrates an example of training an agent to fulfill a navigation request using ADEL. Our results (Table 2) show that ADEL significantly outperforms RL baselines in terms of both learning efficiency and effectiveness. On the other hand, ADEL's success rate is competitive with those of the IL baselines on the navigation task and is lower by 4% on the word modification task. It takes approximately 5-8 times more training episodes than the IL baselines to reach comparable success rates, which is quite respectable considering that the algorithm has to search in an exponentially large space for the ground-truth executions whereas the IL baselines are *given* these executions. Therefore, ADEL can be a preferred algorithm whenever annotating ground-truth executions is not feasible or is substantially more expensive than describing executions. For example, in the word-modification task, ADEL teaches the agent without requiring a teacher with knowledge about regular expressions, who can be costly to recruit in practice. We believe the capability of non-experts to provide feedback will make ADEL and more generally the ILIAD protocol a strong contender in many scenarios.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Jacob Andreas, Dan Klein, and Sergey Levine. Learning with latent language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2166–2179, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1197. URL `https://www.aclweb.org/anthology/N18-1197`.

Yoav Artzi, Nicholas FitzGerald, and Luke Zettlemoyer. Semantic parsing with combinatory categorial grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Tutorials)*, pp. 2, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P13-5002`.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *Association for the Advancement of Artificial Intelligence*, 2018.

Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020.

Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *Proceedings of the International Conference on Learning Representations*, 2021.

David L Chen and Raymond J Mooney. Learning to interpret natural language navigation instructions from observations. In *Association for the Advancement of Artificial Intelligence*, 2011.

Howard Chen, Alane Suhr, Dipendra Misra, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *Proceedings of the International Conference on Learning Representations*, 2019.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Czarnecki, Max Jaderberg, Denis Teplyashin, Marcus Wainwright, Chris Apps, Demis Hassabis, and Phil Blunsom. Grounded language learning in a simulated 3D world. *CoRR*, abs/1706.06551, 2017.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20, pp. 817–824. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper/2007/file/4b04a686b0ad13dce35fa99fa4161c65-Paper.pdf`.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016.

Sobhan Miryoosefi, Kianté Brantley, Hal Daume III, Miro Dudik, and Robert E Schapire. Reinforcement learning with convex constraints. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 14093–14102. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/873be0705c80679f2c71fbf4d872df59-Paper.pdf`.

Dipendra Misra, John Langford, and Yoav Artzi. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017.

Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, November 2019. URL `https://arxiv.org/abs/1909.01871`.

Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. URL `https://arxiv.org/abs/1812.04155`.

D. A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2011.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020.

Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. In *Proceedings of the International Conference on Learning Representations*, 2017.

Wen Sun, Anirudh Vemula, Byron Boots, and J. Andrew Bagnell. Provably efficient imitation learning from observation alone. In *Proceedings of the International Conference of Machine Learning*, June 2019.

Stefanie Tellex, Pratiksha Thaker, Josh Joseph, and Nicholas Roy. Toward learning perceptually grounded word meanings from unaligned parallel data. In *Proceedings of the Second Workshop on Semantic Interpretation in an Actionable Context*, pp. 7–14, Montréal, Canada, June 2012. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W12-2802`.

Terry Winograd. Understanding natural language. *Cognitive Psychology*, 3(1):1–191, 1972.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3357–3364. IEEE, 2017.