Yes, this Way! Learning to Ground Referring Expressions into Actions with Intra-episodic Feedback from Supportive Teachers

Philipp Sadler[1], **Sherzod Hakimov**[1] and **David Schlangen**[1,2]
[1]CoLabPotsdam / Computational Linguistics
Department of Linguistics, University of Potsdam, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
firstname.lastname@uni-potsdam.de

Abstract

The ability to pick up on language signals in an ongoing interaction is crucial for future machine learning models to collaborate and interact with humans naturally. In this paper, we present an initial study that evaluates intra-episodic feedback given in a collaborative setting. We use a referential language game as a controllable example of a task-oriented collaborative joint activity. A teacher utters a referring expression generated by a well-known symbolic algorithm (the "Incremental Algorithm") as an initial instruction and then monitors the follower's actions to possibly intervene with intra-episodic feedback (which does not explicitly have to be requested). We frame this task as a reinforcement learning problem with sparse rewards and learn a follower policy for a heuristic teacher. Our results show that intra-episodic feedback allows the follower to generalize on aspects of scene complexity and performs better than providing only the initial statement.

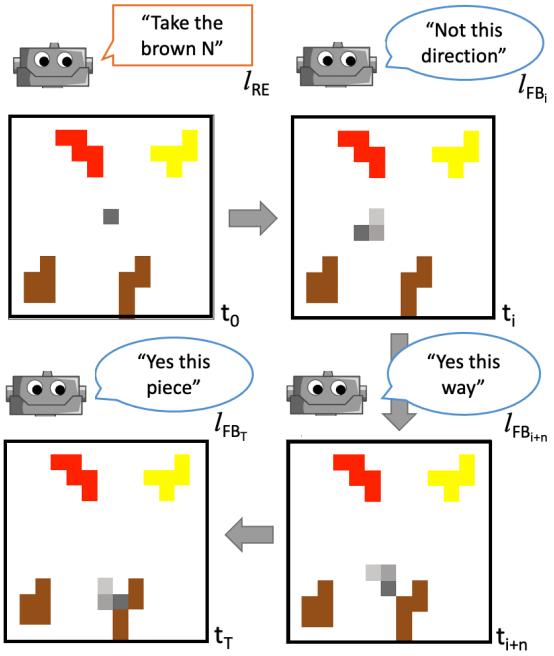

Figure 1: An exemplary interaction between a teacher and a follower that controls the gripper (the grey square). After an initial referring expression l_{RE} at t_0, the teacher provides feedback l_{FB_t} based on the follower's actions until the correct piece is selected at time step T.

1 Introduction

The communicative acts of humans in collaborative situations can be described as two parts of a joint act: signalling and recognizing. In such joint activities, these signals work as coordination devices to increment on the current common ground of the participants (Clark, 1996). The ability to act on these language signals is crucial for future machine learning models to naturally collaborate and interact with humans (Lemon, 2022; Fernández et al., 2011). Such a collaborative interaction with humans usually happens fluently, where one communicative act is performed after the other. The framework of reinforcement learning (RL) (Sutton and Barto, 2018) describes such mechanics where an agent is exposed in steps to observations of an environment with dynamic factors such as the position of objects or language expressions. The goal is that the agent learns to behave generally well in a particular environment solely based on the observations it makes and rewards it gets.

A key challenge here is the variability of expressions in language that can be said to the agent during an interaction. Even in relatively simple environments, there might arise an overwhelming amount of situations for an agent to handle (Chevalier-Boisvert et al., 2019). Recent work on collaborative agents focuses on large pre-collected datasets for imitation learning to learn agents in complex simulated visual environments (Gao et al., 2022; Padmakumar et al., 2022; Pashevich et al., 2021) or frames the learning as a contextual bandit problem (Suhr and Artzi, 2022; Suhr et al., 2019). Nevertheless, other work has shown that intermediate language inputs are a valuable signal to improve the agent's learning performance in task-oriented visual environments (Co-Reyes et al.,

Findings of the Association for Computational Linguistics: ACL 2023, pages 9228–9239
July 9-14, 2023

2019; Mu et al., 2022).

In this paper, we present an initial study that evaluates a follower's learning success given a teacher's intra-episodic feedback in a collaborative setting. We use a referential language game (in English) as a controllable example of a task-oriented collaborative joint activity (see Figure 1). In this game one player (the follower) is supposed to select a piece based on the another player's directives (the teacher). We assume a teacher that utters referring expressions as initial instructions and then responds to the follower's actions with intra-episodic feedback. We frame this as a RL problem with sparse rewards where the intermediate feedback is not part of the reward function but its potential usefulness is learnt by the follower alone.[1]

2 Related Work

Vision and language navigation. In vision and language navigation, an agent is given a natural language instruction which is to be understood to navigate to the correct goal location in a visually observed environment (Gu et al., 2022). The follower can usually ask an Oracle for further information, if necessary (Nguyen et al., 2019; Nguyen and III, 2019; Fried et al., 2018). We extend on this idea and aim for an ongoing interaction with corrections that loosens the turn-based paradigm by letting the Oracle choose when to speak as part of the environment. Hence, in our reference game, the language back-channel for the follower is cut, so that we force the follower to rely more on the visual observations for task success.

Continual learning from human feedback. Suhr and Artzi (2022) let humans instruct the follower and then ask them to rate the agent's behaviour (thumbs up or down). This binary feedback is used for further training as the reward signal in a contextual bandit framework. They show that the agent improves over several interactions with humans. Similarly we evaluate the learning process in the context of RL because it imposes "weaker constraints on the regularity of the solution" (Nguyen et al., 2019), but take a broadly available, off-the-shelf learning algorithm (Schulman et al., 2017) to directly study the effects of different kinds of feedback. The feedback given to our agent is of natural language and not directly bound to the reward; the follower needs to learn the meaning of the language feedback itself.

Language-guided policy learning. Chevalier-Boisvert et al. (2019) compared the sampling complexity of RL and imitation learning (IL) agents on various language-conditioned tasks. They proposed a 2-dimensional visual environment called *Minigrid* in which an agent is given a single mission statement that instructs the agent to achieve a specific state, e.g. "Take the red ball". In contrast to them we intentionally do not use IL approaches, because then the agent would have already learnt how to ground the language signals. We want to test if the agent can pick-up on the language from the interaction alone. For this, we similarly propose a diagnostic environment to directly control for the distributions of target objects (cf. skewed distribution of target objects in CVDN (Thomason et al., 2019)) and feedback signals.

Other work uses the *Minigrid* environment to propose a meta-training approach that improves the learning via natural language corrections, e.g. "Pick up the green ball" (Co-Reyes et al., 2019). The agent is given an episodic correction if a specific task cannot be solved. In this way, the agent must not only ground the mission statement but also ground the corrections into actions. Mu et al. (2022) improve policy learning with intra-episodic natural language sub-goals e.g. "Pick up the ball". These sub-goals are provided by a trained teacher policy when a previous sub-goal has been reached. In contrast, we rather follow earlier work (Engonopoulos et al., 2013) on monitoring execution and use a heuristic teacher which provides intra-episodic language feedback whenever it appears feasible. The agent has to learn that certain pairs of feedback and behaviour at a specific time-step lead to the task's success and others to failure.

3 The CoGRIP environment

We use a **Co**llaborative **G**ame of **R**eferential and **I**nteractive language with **P**entomino pieces as a controllable setting. A teacher instructs a follower to select a specific piece using a gripper. Both are constrained as follows: The teacher can provide utterances but cannot move the gripper. The follower can move the gripper but is not allowed to provide an utterance. This asymmetry in knowledge and skill forces them to work together and coordinate. Zarrieß et al. (2016) found that this settings leads to diverse language use on the teacher's side.

[1] Code is publicly available at https://github.com/clp-research/intra-episodic-feedback.

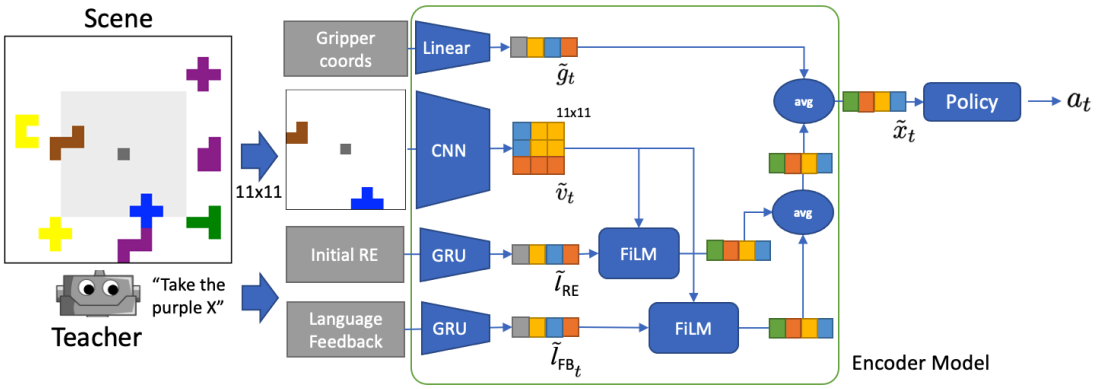

Figure 2: The information flow through our encoder model which produces the features $\tilde{x}_t$ as an input to the policy.

3.1 Problem Formulation

The follower has to navigate a gripper to select a piece described by the teacher. We frame this task as a RL problem with sparse rewards. At each time-step t, given an observation $o_t \in \mathcal{O}$ of the environment, the agent has to select an action $a_t \in \{\texttt{LEFT}, \texttt{RIGHT}, \texttt{UP}, \texttt{DOWN}, \texttt{WAIT}, \texttt{GRIP}\}$ such that the overall resulting sequence of actions $(a_0, ..., a_t, ..., a_T)$ maximizes the sparse reward $\mathcal{R}(o_T) = r$. An episode ends when the GRIP action is chosen, and the gripper position g_t is in the boundaries of a piece. An episode also ends when t reaches $T_{max} = 100$. Following Chevalier-Boisvert et al. (2019), the reward function returns a basic reward minus the movement effort $\mathcal{R} = 1 - 0.9 * (T/T_{max})$. We extend this formulation and give an additional bonus of $+1$ if the correct piece has been taken or a penalty of -1 when the wrong or no piece has been taken at all.

3.2 Environment

The environment exposes at each time-step t an observation o_t that contains the gripper coordinates $g_t = (x, y)$, the initial referring expression l_{RE}, the language feedback l_{FB_t} (which might be empty) and a partial view v_t of the scene. While the scene as a whole is represented as a 2-dimensional image (with RGB colour channel), the partial view represents a 11×11-sized cut out, centered on the gripper position (see Figure 2). The teacher generates the initial and feedback statements.

3.3 Teacher

For the teacher, we assume a heuristic behaviour (a fix policy) that has been shown to lead to collaborative success with humans (Götze et al., 2022) and leave the complexity of learning in a multi-agent setting (Gronauer and Diepold, 2022) for future work. The teacher produces an initial referring expression $l_{\text{RE}} = (w_0, ..., w_N)$ where N is the message length and w_i is a word in the vocabulary. The production rule is implemented following the Incremental Algorithm (IA) (Dale and Reiter, 1995) that is given the symbolic representations of the pieces on the board (see Appendix A.1). The teacher provides a feedback message $l_{\text{FB}_t} = (w_0, ..., w_N)$ at a time-step $t_{>0}$ when the gripper's position g_t has exceeded a pre-defined distance threshold $D_{\text{dist}} = 3$ compared to the gripper's last position of feedback $g_{\text{FB}_{\text{last}}}$ or it is over a piece. The generated feedback is of positive sentiment ("Yes this way/piece") when the gripper is then closer to or over the target piece and negative otherwise ("Not this direction/piece"). Alternatively, suppose the follower does not exceed the distance threshold after $D_{\text{time}} = 6$ time-steps the feedback message is the same as the initial statement. Overall, the property values and sentence templates lead to a small vocabulary of 33 words.

3.4 Follower

The follower agent has to move the gripper and successfully grip a piece solely based on the observations. The observations $o_t = (v_t, g_t, l_{\text{RE}}, l_{\text{FB}_t})$ are mapped to 128-dimensional features $\tilde{x}_t \in \mathbb{R}$ using the encoder model (see Figure 2). Following Chevalier-Boisvert et al. (2019), the word embeddings (which are learned from scratch) of the language inputs are fed through a Gated Recurrent Unit (GRU) (Cho et al., 2014) and then combined with the embedded visual features using a Feature-wise Linear Modulation (FiLM) layer (Perez et al., 2018). These language conditioned visual features are then max pooled, averaged and again averaged with the gripper position. Given the resulting features $\tilde{x}_t$, we learn a parameterised policy

$\pi(\tilde{x}_t; \theta) \sim a_t$ that predicts a distribution over the action space. We use the Proximal Policy Optimization (PPO) (Schulman et al., 2017) implementation of *StableBaselines3* v1.6.2 (Raffin et al., 2021) to train the policy in our environment.

3.5 Tasks

The follower has to grip an intended target piece among several other pieces (the distractors). Thus a task is defined by the number of pieces, the target piece and the map size. The pieces for the tasks are instantiated from symbolic representations: a tuple of shape (9), color (6) and position (8) which leads to 432 possible piece symbols. For our experiments we use all of these symbols as targets, but split them into distinct sets (Appendidx A.4). Therefore the targets for testing tasks are distinct from the ones in the training tasks. We ensure the reproducibility of our experiments by constructing 3300 training, 300 validation, 720 testing tasks representing scenes with a map size of 20×20 and 4 or 8 pieces.

4 Experiments

In this section we explore the effects of the teacher's language and intra-episodic feedback on the follower's success and ask whether the follower generalizes on aspects of scene complexity.

4.1 Which referential language is most beneficial for the agent's learning success?

As suggested by Madureira and Schlangen (2020) we explore the question of which language is most effective. The IA constructs the initial reference by following a preference order over object properties (Krahmer et al., 2012). We hypothesize that a particular order might be more or less suitable depending on the task. Thus we conduct a series of experiments *without* the feedback signal where the preference order is varied as the permutation of color, shape and position. Our results indicate that such orders perform better that prioritize to mention positional attributes as distinguishing factors of the target piece (see Table 1). This is reasonable as the directional hint reduces the agent's burden for broader exploration. The follower is able to pick up early on these positional clues and performs overall better during training (see Figure 3).

4.2 What is the agent's performance gain with intra-episodic feedback in our setting?

We conduct the same experiments as above *with* intra-episodic language feedback to measure its effect on the follower's success rate. Our results show that the follower achieves higher success rates with intra-episodic feedback among all preference orders (see Table 1). We also notice that the gain is higher for the low-performing preference orders. This shows that the intra-episodic feedback is a valuable signal for the follower to overcome miss-

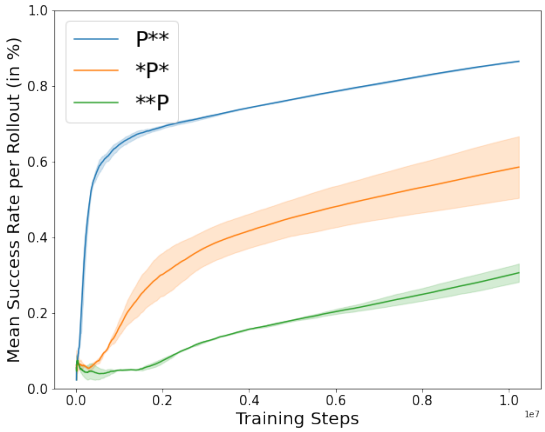

Figure 3: The mean success rates per rollout (during training) grouped by the teacher's preference of position i.e. teacher's with P** start with the position description to rule out distractors and teacher's with **P use the position only, when color or shape are not enough to distinguish the target piece from others. The curves show that a preference for position descriptions lead to faster training success of the follower.

	only initial RE		with intra-episodic Feedback	
Pr.Or.	mSR	mEPL	mSR	mEPL
C-P-S	39.44	59.71	87.64 (+48.2)	22.12 (-37.6)
S-P-C	25.42	68.49	78.75 (+53.3)	32.44 (-36.0)
P-C-S	**73.19**	**23.06**	93.89 (+20.7)	15.39 (-7.7)
P-S-C	70.56	23.10	**94.44** (+23.9)	**14.80** (-8.3)
C-S-P	15.14	81.18	79.86 (+64.7)	33.06 (-48.1)
S-C-P	13.33	85.30	70.69 (+57.4)	42.71 (-42.6)

Table 1: The mean success rates (mSR in %) and episodes lengths (mEPL) of the agent on the test tasks when the teacher follows a particular preference order over target piece properties (color (C), shape (S), position (P)) with language feedback and without it (only initial RE). A shortest path solver reaches 10.96 mEPL.

Generalization Tasks	P-C-S	w/ FB	# Tasks
Test 30x30 (12P,18P)	39.17	80.56	360
Test 30x30 (4P,8P)	61.94	91.39	360
Holdout 20x20 (4P,8P)	63.31	94.44	864
Test 20x20 (4P,8P)	73.19	93.89	720

Table 2: The mean success rates (mSR in %) of the best agent (a teacher with pref. order P-C-S) on the generalization tasks. The agent with the intra-episodic feedback (w/ FB) performs much better on these more complex scenes. Number of pieces abbreviated, for example 4P means 4 pieces. Map sizes given by NNxNN.

ing directives in the initial referring expressions. The agent can learn strategies incorporating the feedback signals. This is an interesting finding because language feedback is not part of the reward function and could be empty.

4.3 Does intra-episodic feedback help the agent to generalize on scene complexity?

As a proxy for generalization capabilities, we take the best performing follower and raise the complexity of the *testing* scenes along two dimensions (i) we increase the map size to 30×30 and (ii) put up to 18 pieces on the board. In addition, we hold out 72 combinations of piece shapes and colors that have never been seen during training. Our results show that the agent trained with intra-episodic feedback is able to perform better (i) on the larger map size, (ii) the higher number of pieces and (iii) the new target pieces compared to the one without (see Table 2).

5 Conclusion

In this work, we studied the effects of a teacher's language and intermediate interventions (the feedback) towards a learner's success and whether the learner generalizes on aspects of scene complexity. Our results show that there is a most beneficial language for the teacher. Its intra-episodic feedback allows the learner to learn faster and generalize better than without intermediate help. An exciting direction for further work is to show the benefits of language feedback for other reinforcement learning problems, to overcome the limits of the heuristic teacher strategy and to reduce the need for feedback after successful training.

6 Limitations

Limits on visual variability and naturalness. The Pentomino domain can only serve as an abstraction for referring expression generations in visual domains. The amount of objects is limited to 9 different shapes and the number of colors is reduced to 6 as well. The positions are chosen to be discrete and absolute while real-world references might include spatial relations. Furthermore, the pieces show no texture or naturalness, but are drawn with a solid color fill. We choose this simplified domain to focus on the interaction between the follower and the teacher and left the evaluation of the proposed models on more realistic looking scenes for further work. Nevertheless, we think our approach can also be applied to photo-realistic environments (Ramakrishnan et al., 2021; Kolve et al., 2017).

Limits on variability of the referring expressions. We only explored expressions that are generate by the Incremental Algorithm. Moreover, we choose a fix property value order (color is mentioned before shape is mentioned before position) for the realisation of the template's surface structure and left the exploration for a higher variability to further work.

Limits on variability of the feedback signal. In this work we used a heuristic teacher with a fixed behavior to provide the intermediate feedback to the follower. We choose this Oracle speaker for better control over the experiments and to focus on the research questions of which feedback is most helpful and how it should be presented (contain which information). We are aware that in natural interaction the teacher's responses might be more dynamic and can be potentially learnt in a much more complex multi-agent RL settings which would go beyond our focused contribution here. Still this is an interesting prospect for future research.

7 Ethics Statement

For now, we see no immediate threats regarding this work, because the experiments are performed in a controlled setting of an abstract domain. But since this research has collaborative agents in prospect people might use more advanced stages of this technique to train agents on possibly other tasks. Thus we encourage everyone to apply such a technology only for good use and to avoid harmful applications.

Acknowledgements

We want to thank the anonymous reviewers for their comments. This work was funded by the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) – 423217434 ("RECOLAGE") grant.

References

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2019. Babyai: A platform to study the sample efficiency of grounded language learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Herbert H. Clark. 1996. *Using Language*. 'Using' Linguistic Books. Cambridge University Press.

John D. Co-Reyes, Abhishek Gupta, Suvansh Sanjeev, Nick Altieri, Jacob Andreas, John DeNero, Pieter Abbeel, and Sergey Levine. 2019. Guiding policies with language via meta-learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cogn. Sci.*, 19(2):233–263.

Nikos Engonopoulos, Martin Villalba, Ivan Titov, and Alexander Koller. 2013. Predicting the resolution of referring expressions from user behavior. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1354–1359. ACL.

Raquel Fernández, Staffan Larsson, Robin Cooper, Jonathan Ginzburg, and David Schlangen. 2011. Reciprocal Learning via Dialogue Interaction: Challenges and Prospects. In *Proceedings of the IJCAI 2011 Workshop on Agents Learning Interactively from Human Teachers (ALIHT 2011)*.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3318–3329.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S. Sukhatme. 2022. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics Autom. Lett.*, 7(4):10049–10056.

Sven Gronauer and Klaus Diepold. 2022. Multi-agent deep reinforcement learning: a survey. *Artif. Intell. Rev.*, 55(2):895–943.

Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. 2022. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7606–7623. Association for Computational Linguistics.

Jana Götze, Karla Friedrichs, and David Schlangen. 2022. Interactive and Cooperative Delivery of Referring Expressions: A Comparison of Three Algorithms. In *Proceedings of the 26th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Virtually and at Dublin, Ireland. SEMDIAL.

Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474.

Emiel Krahmer, Ruud Koolen, and Mariët Theune. 2012. Is it that difficult to find a good preference order for the incremental algorithm? *Cogn. Sci.*, 36(5):837–841.

Oliver Lemon. 2022. Conversational grounding as natural language supervision – the need for divergent agent data. In *ACL Workshop on Learning with Natural Language Supervision*.

Brielen Madureira and David Schlangen. 2020. An overview of natural language state representation for reinforcement learning. In *Proceedings of the ICML Workshop on Language in Reinforcement Learning*.

Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah D. Goodman, Tim Rocktäschel, and Edward Grefenstette. 2022. Improving intrinsic exploration with language abstractions. *CoRR*, abs/2202.08938.

Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. 2019. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12527–12537. Computer Vision Foundation / IEEE.

Khanh Nguyen and Hal Daumé III. 2019. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 684–695. Association for Computational Linguistics.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gökhan Tür, and Dilek Hakkani-Tür. 2022. Teach: Task-driven embodied agents that chat. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 2017–2025. AAAI Press.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. 2021. Episodic transformer for vision-and-language navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15922–15932. IEEE.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3942–3951. AAAI Press.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. 2021. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Alane Suhr and Yoav Artzi. 2022. Continual learning for instruction following from realtime feedback. *CoRR*, abs/2212.09710.

Alane Suhr, Claudia Yan, Jacob Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2119–2130. Association for Computational Linguistics.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, second edition. The MIT Press.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2019. Vision-and-dialog navigation. *CoRR*, abs/1907.04957.

Kees van Deemter. 2016. *Computational Models of Referring*, chapter 4.6. The MIT Press.

Sina Zarrieß, Julian Hough, Casey Kennington, Ramesh Manuvinakurike, David DeVault, Raquel Fernández, and David Schlangen. 2016. PentoRef: A Corpus of Spoken References in Task-oriented Dialogues. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 125–131, Portorož, Slovenia. European Language Resources Association (ELRA).

A Appendix

A.1 Teacher Details

Hyperparameters

- $D_{\text{dist}} = 3$
- $D_{\text{time}} = 6$
- preference_order

The distances between two coordinates (p_1, p_2) are calculated as the euclidean distance.

The Incremental Algorithm (IA) The Algorithm 1, in the formulation of (Dale and Reiter, 1995), is supposed to find the properties that uniquely identify an object among others given a preference over properties. To accomplish this the algorithm is given the property values $\mathcal{P}$ of distractors in M and of a referent r. Then the algorithm excludes distractors in several iterations until either M is empty or every property of r has been tested. During the exclusion process the algorithm computes the set of distractors that do *not* share a given property with the referent and stores the property in $\mathcal{D}$. These properties in $\mathcal{D}$ are the ones that distinguish the referent from the others and thus will be returned.

The algorithm has a meta-parameter $\mathcal{O}$, indicating the *preference order*, which determines the order in which the properties of the referent are tested against the distractors. In our domain, for example, when *color* is the most preferred property, the algorithm might return BLUE, if this property already excludes all distractors. When *shape* is the preferred property and all distractors do *not* share the shape T with the referent, T would be returned. Hence even when the referent and distractor pieces are the same, different preference orders might lead to different expressions.

Algorithm 1 The IA on symbolic properties as based on the formulation by van Deemter (2016)

Require: A set of distractors M, a set of property values $\mathcal{P}$ of a referent r and a linear preference order $\mathcal{O}$ over the property values $\mathcal{P}$

$\mathcal{D} \leftarrow \emptyset$
for P in $\mathcal{O}(\mathcal{P})$ **do**
  $\mathcal{E} \leftarrow \{m \in M : \neg P(m)\}$
  if $\mathcal{E} \neq \emptyset$ **then**
    Add P to $\mathcal{D}$
    Remove $\mathcal{E}$ from M
return $\mathcal{D}$

Referring Expression Templates There are 3 expression templates that are used when only a single property value of the target piece is returned by the Incremental Algorithm (IA):

- *Take the [color] piece*
- *Take the [shape]*
- *Take the piece at [position]*

Then there are 3 expression templates that are selected when two properties are returned:

- *Take the [color] [shape]*
- *Take the [color] piece at [position]*
- *Take the [shape] at [position]*

And finally there is one expression templates that lists all property values to identify a target piece:

- *Take the [color] [shape] at [position]*

Feedback Expression Templates We use two templates to give positive or negative feedback on the direction of the follower

- *Yes this way*
- *Not this way*

And we give a similar feedback when the follower is locating the gripper over a piece

- *Yes this piece*
- *Not this piece*

The vocabulary Overall, the property values and sentence templates lead to a small vocabulary of 33 words:

- 9 shapes: F, N, P, T, U, W, X, Y, Z
- 6 colors: red, yellow, green, blue, purple, brown
- 6 position words: left, right, top, bottom, center (which are combined to e.g., right center or top left)
- 8 template words: take, the, piece, at, yes, no, this, way
- 4 special words: <s>, <e>, <pad>, <unk>

The maximal sentence length is 11.

A.2 Follower Details

Agent Parameters: $9,456$

word_embedding_dim	128
feature_embedding_dim	128
actor_layers	2
actor_dims	128
vf_layers	2
vf_dims	128

Table 3: Agent hyperparameters

The max-pooling layer additionally downsamples the language conditionaed visual features from $11 \times 11 \times 128$ to $1 \times 1 \times 128$ dimensions. For this we use the `nn.AdaptiveMaxPool2d((1, 1))` layer from PyTorch v1.11.0. In addition, before we average the gripper coordinates features and the resulting language conditioned visual features, we apply a layer normalization (eps = 1e-5) on them.

Architecture Search We performed a little architecture search where we evaluated two methods for visual encoding (pixels, symbols), four methods for language encoding (word embeddings with GRU,one-hot word embeddings with GRU, one-hot sentence embeddings, pre-trained sentence embeddings) and two methods for the fusion (concatenate, FiLM). We found learnt word embeddings and FiLM perform best in regard of training speed and success rate. The visual encodings showed similar performance but we prefer the pixel encoder because it makes less assumptions about the world.

Learning Algorithm We apply a learning rate schedule that decreases the learning rate during training according to the training progress (based on the number of time steps) with $p \in [0, 1]$, but the learning rate is given a lower bound α_{min} so that it never reaches zero: $\alpha_t = \max(p \cdot \alpha_{\text{init}}, \alpha_{\text{min}})$

lr_init	2.5e-4
lr_min	2.5e-5
num_epochs	8
buffer_per_env	1024
clip_range	0.2
clip_range_vf	0.2
ent_coef	0.01
vf_coef	0.5
target_kl	0.015

Table 4: PPO hyperparameters

A.3 Environment Details

Board The internal representation of the visual state is a 2-dimensional grid that spans $W \times H$ tiles where W and H are defined by the map size. A tile is either empty or holds an identifier for a piece (the tile is then occupied). The pieces are defined by their colour, shape and coordinates and occupy five adjacent tiles (within a virtual box of 5×5 tiles). The pieces are not allowed to overlap with another piece's tiles. For a higher visual variation, we also apply rotations to pieces, but we ignore the rotation for expression generation, though this could be an extension of the task.

Name	HEX	RGB
red	#ff0000	(255, 0, 0)
yellow	#ffff00	(255, 255, 0)
green	#008000	(0, 128, 0)
blue	#0000ff	(0, 0, 255)
purple	#800080	(128, 0, 128)
brown	#8b4513	(139, 69, 19)

Table 5: The colors for the Pentomino pieces.

Gripper The gripper can only move one position at a step and can move over pieces, but is not allowed to leave the boundaries of the board. The gripper coordinates $\{(x, y) : x \in [0, W], y \in [0, H]\}$ are projected to $\{(x, y) : x, y \in [-1, +1]\}$ so that the coordinate in the center is represented with $(0, 0)$. This provides the agent with the necessary information about its positions on the overall board as its view field is shrinked to 11×11 tiles. In addition, to provide the agent with a notion of velocity, the environment keeps track of the last two gripper positions and applies a grey with decreasing intensity to these positions on the board:

- $\text{color}_{g_t} = (200, 200, 200)$
- $\text{color}_{g_{t-1}} = (150, 150, 150)$
- $\text{color}_{g_{t-2}} = (100, 100, 100)$

A.4 Task Details

We created training, validation, test and holdout splits of target piece symbols (a combination of shape, color and position) for the task creation (see Table 6). We split these possible target piece symbols so that each subset still contains all colors, shapes and positions, but different combinations of them. For example, the training set might contain a "red F" but this is never seen at the bottom left. Though this will be seen during validation or

testing. An exception is the holdout split where we hold out a color for each shape. This means that for example a "green T" is never seen during training, but a "green F" or a "blue T".

		# of Tasks					
		Map Size=20		Map Size=30			
	# TPS	N=4	N=8	N=4	N=8	N=12	N=18
training	275	1650	1650				
validation	25	150	150				
testing	60	360	360	180	180	180	180
holdout	72	432	432				

Table 6: The target piece symbols (TPS) distributed over the task splits with different map sizes (Size) and number of pieces (N) on the board. The total possible number of target piece symbols is $9 \cdot 6 \cdot 8 = 432$.

To create a task we first place the target piece on a board with the wanted map size. Then we sample uniform random from all possible pieces and place them until the wanted number of pieces is reached. If a piece cannot be placed 100 times, then we re-sample a piece and try again. The coordinates are chosen at random uniform from the coordinates that fall into an area of the symbolic description. We never set a piece into the center, because that is the location where the gripper is initially located.

A.5 Experiment Details

We trained the agents on a single GeForce GTX 1080 Ti (11GB) where each of them consumed about 1GB of GPU memory. The training spanned 10.24 million time steps executed with 4 environments simultaneously (and a batch size of 64). The training took between 9.24 and 12.32 hours (11.86 hours on average). The random seed was set to 49184 for all experiments. We performed an evaluation run on the validation tasks after every 50 rollouts (51, 200 timesteps) and saved the best performing agent according to the mean reward.

Pr.Or.	Step in K w/ FB	Step in K w/o FB
C-S-P	8,601	8,806
C-P-S	8,396	9,911
P-S-C	9,216	9,830
P-C-S	5,939	9,830
S-P-C	5,529	8,806
S-C-P	7,984	10,035

Table 7: The timesteps of the best model checkpoints.

As the evaluation criteria on the testings tasks we chose success rate which indicates the relative number of episodes (in a rollout or in a test split) where the agent selected the correct piece:

$$\text{mSR} = \frac{\sum^{N} s_i}{N} \text{ where } s_i = \begin{cases} 1, & \text{for correct piece} \\ 0, & \text{otherwise} \end{cases}$$

B Additional Results

In addition, we notice that the feedback has a positive effect on early success rates during *training* when we compare training runs of the same preference order groups with and without feedback (see 4). The intra-episodic feedback largely improves the early success rates of agents with teachers of preference orders `**P` (`SCP`, `CSP`) as well as those with preference orders `*P*` (`SPC`, `CPS`). There is also a noticable but lower effect on the preference orders `P**` (`PSC`, `PCS`) that perform already well early without the intra-episodic feedback. Though the latter seem to be confused by the feedback initially (until 10% of the training steps). The benefit of intra-episodic feedback is starting to decrease in later time steps, because the agent without that additional signal catch up on the success rates. Still these findings show that intra-episodic feedback is helpful to improve the learning in early stages.

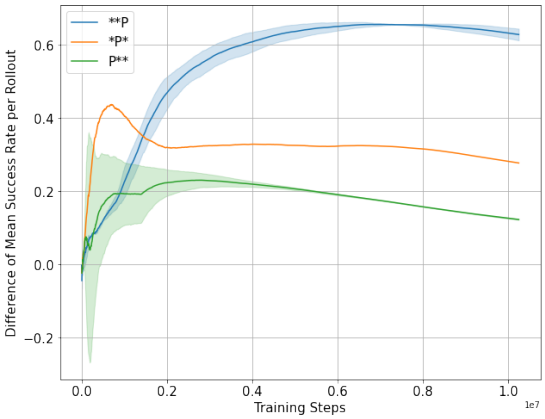

Figure 4: The difference of success rates during *training*, when we directly compare agents that are exposed to teacher's with the similar preference orders (`P**`, `*P*`, `**P`). The lines indicate the success rates of the agents with feedback minus the success rates of agents without feedback.

C Misc

Robot image in Figure 1 adjusted from `https://commons.wikimedia.org/wiki/File:Cartoon_Robot.svg`. That file was made available under the Creative Commons CC0 1.0 Universal Public Domain Dedication.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

A.4

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
A.4

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
A.2 A.5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
A.2 A.5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4 A.5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3.4 A.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.