ℓGym: Natural Language Visual Reasoning with Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

1 Introduction

The inclusion of natural language context in reinforcement learning (RL) provides both opportunities and challenges. Language provides an expressive and accessible conduit for task specification, so that RL agents can address a broad set of tasks in a given environment, rather than learn a single behavior. For language learning, RL is a promising avenue for language use and acquisition through world interaction. Fulfilling this potential requires addressing core reasoning challenges: the RL agent must reason about both high-level language concepts, low-level actions, and the relations between them.

Despite significant interest and promising approaches, it has been challenging to create expressive RL benchmarks with natural language. Existing benchmarks often make various simplifications, such as using synthetic language that does not reflect the complexity of natural language [Shinagawa et al., 2018, El-Nouby et al., 2018, 2019] and using reward functions that do not accurately capture language semantics [Misra et al., 2017]

We present ℓGym, a reinforcement learning benchmark for natural language visual reasoning. ℓGym is designed to address the above issues. It includes the semantically diverse natural language of the NLVR corpus [Suhr et al., 2017], which has highly compositional human-written language and requires complex grounded reasoning. ℓGym provides an executable evaluation function for every statement in the NLVR corpus, and these annotations align the reward function with the language semantics of the agent's underlying reasoning task. We experiment with standard on-policy RL algorithms. Our experimental results show that while existing methods are able to achieve non-trivial performance, the complex visual reasoning required by ℓGym forms a challenging open problem.

2 The ℓGym Benchmark

ℓGym consists of a collection of environments that share a common backbone. The backbone is a 2D plane that is manipulated by placing and removing objects of different types. Each environment instance is a Markov Decision Process (MDP) created by pairing a natural language statement and a target boolean value with a configuration of the shared backbone. The goal of the agent in each environment is to manipulate it by adding and removing objects so that the truth-value of the statement with regard to the environment is the target boolean.

The learning problem ℓGym presents is to induce a policy that generalizes across MDPs. We split the MDPs to training, development, and held-out testing sets. The training environments are to be used for parameter estimation, while the two other sets are for testing during development and for final held-out testing to report approach performance.

There are two dimensions of configuration: appearance and starting condition. The appearance determines the state space, transition function, and action space. The starting condition determines

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

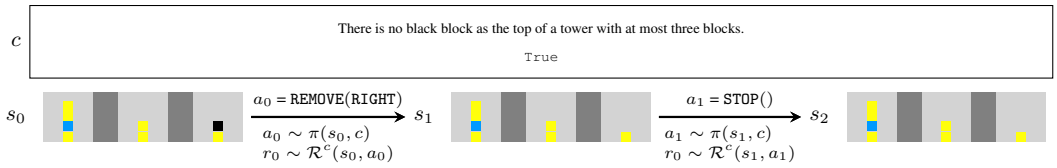


Figure 1: Overview of an example for one CMDP, TOWER-FLIPIT. The context c consists of a text statement and a target boolean. The leftmost image depicts the initial state s_0. The agent π is presented with (s_0, c), and chooses an action $a_0 \sim \pi(\cdot|s_0, c)$. The environment transitions to the next updated state s_1, the context remaining the same.

the agent's goal. The appearance of the environment can be (a) TOWER: the objects include squares only, and they can be stacked into towers only; or SCATTER: objects of different types can be freely distributed. The starting condition and agent objective can be: (a) SCRATCH: the environment starts without any objects and the goal is to modify it so that the statement's truth-value is True; or (b) FLIPIT: the environment starts with a set of objects and the agent's goal is to flip the truth-value of the statement.

There are four configurations: TOWER-SCRATCH, TOWER-FLIPIT, SCATTER-SCRATCH, and SCATTER-FLIPIT. Each configuration forms a Contextual Markov Decision Process [CMDP; Hallak et al., 2015]. CMDP is an abstraction over a set of Markov Decision Processes (MDPs) to account for a context that remains constant throughout the interaction with an MDP. We set the context to include the statement and the target boolean the interaction is conditioned on. A CMDP is a tuple $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \mathcal{M}(c))$, where $\mathcal{C}$ is the context space, $\mathcal{S}$ the state space, $\mathcal{A}$ the action space, and $\mathcal{M}$ a function mapping a context $c \in \mathcal{C}$ to an MDP $\mathcal{M}(c) = (\mathcal{S}, \mathcal{A}, T, R^c, \beta^c)$. Here, $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a transition function, $R^c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and β^c an initial state distribution. This means that a CMDP is a collection of MDPs that share the same state space, action space and transition function. Table 2 in Appendix A shows the number of MDPs under each configuration. The policy takes as input both the current state and the context that created the MDP. The learning problem is to estimate parameters θ for a policy $\pi_\theta : \mathcal{S} \times \mathcal{C} \to \mathcal{A}$.

Contexts A context $c \in \mathcal{C}$ is a pair $c = (\bar{x}, b)$, where $\bar{x}$ is a natural language statement and $b \in \{\texttt{True}, \texttt{False}\}$ is a target boolean value for the statement $\bar{x}$ with respect to the state s. The target boolean value in SCRATCH is always True. In FLIPIT, the target boolean value can either be True or False.

States A state $s \in \mathcal{S}$ is an RGB image. Images in ℓGym are divided into three box regions of identical dimensions by two gray separators (Figure 1). The objects in ℓGym have three properties, each can take multiple values: shape (CIRCLE, SQUARE or TRIANGLE), color (BLACK, BLUE, or YELLOW), and size (SMALL, MEDIUM or LARGE). In TOWER, states are constrained to have stacks of up to four SQUAREs of MEDIUM size and any color at the center of each box. SCATTER states support all object shapes, sizes, and colors, and they may be positioned freely. In both conditions, objects cannot cross image boundaries or into the separators. The choice between SCRATCH or FLIPIT does not influence the state space.

Actions and Transitions There are three action types STOP, ADD, and REMOVE. STOP terminates the episode and does not require any parameters. The truth-value of the statement is only evaluated and compared to the target boolean after the STOP action is taken. ADD adds objects to the environment, and REMOVE removes objects. They take different arguments for TOWER and SCATTER:

TOWER: Similar to the state space of TOWER, the actions are also more constrained. Both ADD and REMOVE take a position argument, which has three possible values corresponding to the three box regions. Objects are always added or removed at the top of the stack. Adding an object on top of a stack of four objects or removing an object from an empty box are both invalid actions. ADD also takes a color argument. Including STOP, there are $1 + (3 + 1) \times 3 = 13$ actions.

SCATTER: Unlike TOWER, objects of any type can be placed freely in the box regions. Both ADD and REMOVE take 2D coordinates that specify the pixel location. Adding an object places it

so that its top-left coordinates is the given coordinates. Removing an object will remove the object at the given coordinates. Adding also requires specifying the shape, color, and size. If adding results in objects overlap or boundary crossing with the separators or image boundaries, the action is invalid. Removing from a position that does not include an object is also an invalid action. The native resolution of images is 380×100 pixels. Including STOP, there are $1+(380\times100)\times((3\times3\times3)+1) = 1{,}064{,}001$ actions. In our experiments (Section 4), we use a grid of 19×5, giving a total number of 2,661 actions.

The environment transitions are controlled by the transition function $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. Because the choice between TOWER and SCATTER configurations determines the action space, it also determines the transition function. The transition function does not modify the context, which is fixed for a given MDP.

Reward Function The reward function R^c is computed with respect to the context pair $c = (\bar{x}, b)$, where $\bar{x}$ is a natural language statement and b is the target boolean value. The reward is based on evaluating the truth-value of the natural language statement $\bar{x}$ with respect to a state s, and comparing it to the target boolean b. ℓGym includes an executable evaluation function $\mathcal{E}^{\bar{x}} : \mathcal{S} \times \mathcal{A} \rightarrow \{\texttt{True}, \texttt{False}\}$ for all $\bar{x}$. We describe how we create these evaluation functions in Appendix A.

The agent receives a positive reward for terminating the episode using the STOP action with the statement evaluation $\mathcal{E}^{\bar{x}}(s)$ equal to the target boolean value b. If the statement boolean value $\mathcal{E}^{\bar{x}}(s)$ does not equal the target boolean b value when taking the STOP action, the agent receives a negative reward. If the episode terminated because the current time step t reached the action horizon H or because of an invalid action, the agent also receives a negative reward. Action validity depends on the current state s and on the configuration, because TOWER and SCATTER have different action spaces. There is also a verbosity penalty of δ for every other action. Formally, the reward is:

$$R^c(s,a) = \begin{cases} 1.0 & a = \texttt{STOP} \wedge \mathcal{E}^{\bar{x}}(s) = b \\ -1.0 & a = \texttt{STOP} \wedge \mathcal{E}^{\bar{x}}(s) \neq b \\ -1.0 & (a \text{ is invalid in } s) \vee (t = H) \\ -\delta & \text{otherwise} \end{cases} .$$

Initial State Distribution The initial state distribution β^c is parameterized by the context $c \in \mathcal{C}$, which is different between SCRATCH and FLIPIT. In SCRATCH, the agent modifies an empty environment to satisfy the truth-condition of the statement $\bar{x}$ in the context c, so the initial state s_0 is always an empty image. The set of initial states β^c for every context $c \in \mathcal{C}$ is the set of images associated with the statement $\bar{x}$ in the NLVR data. In practice, for FLIPIT, this set includes between 1 to 43 images.

3 Experimental Setup

We consider two RL algorithms paired with two models for a total of four algorithm-model pairs.

3.1 Models

In our experiments, we consider two models: CNN+BERT and ViLT. These two models learn a joint visuo-linguistic representation, which is necessary to solve the proposed CMDP configurations.

CNN+BERT. We combine a CNN network [Fukushima and Miyake, 1982, LeCun et al., 1989] with BERT text features [Devlin et al., 2019] to learn a joint visuo-linguistic representation. The BERT features are fixed and used without fine-tuning. To account for the context target boolean, we train two embeddings for FLIPIT, and combined it with the other two modalities representations. For SCRATCH, we ignore the context target boolean because it is always True (see Section 2).

ViLT. We also evaluate a modern pretrained Transformer [Vaswani et al., 2017] model, Vision-and-Language Transformer (ViLT) [Kim et al., 2021]. Unlike CNN+BERT, ViLT handles both modalities, vision and language, inside a single Transformer instead of separating both components. In preprocessing, we concatenate the sentence with the text-format target boolean separated by a special token <TARGET>. ViLT's pretrained preprocessor is used to preprocess both text and image.

Table 1: Accuracy for all the four CMDP, with both models (CNN+BERT and ViLT) and both algorithms (PPO and PPO+SF). Evaluation is done without stop forcing (i.e. with PPO).

		TOWER-SCRATCH		TOWER-FLIPIT		SCATTER-SCRATCH		SCATTER-FLIPIT	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
PPO	CNN+BERT	71.78	63.27	35.95	34.78	**39.08**	**48.39**	0.00	0.00
	ViLT	**79.76**	**75.93**	**60.36**	**60.09**	35.63	38.71	**3.51**	**6.09**
PPO+SF	CNN+BERT	80.98	78.70	27.22	26.75	**70.12**	**74.84**	8.31	8.46
	ViLT	**82.82**	**82.10**	**64.79**	**62.26**	57.77	65.81	**24.92**	**30.29**

3.2 Algorithms

We couple each model with two algorithms: PPO and PPO+SF. We choose the on-policy algorithm PPO [Schulman et al., 2017] to optimize each of the CMDP proposed in this benchmark because of its empirical success with large models [Stiennon et al., 2020]. In our CMDP configurations pairs, the reward is very sparse because the agent has to select the correct sequence of actions and choose the STOP action as the last action if statement evaluation $\mathcal{E}^{\bar{x}}(s)$ is equal to the target boolean value b. To combat this sparse reward issue, we mask out all actions except for the STOP action when reaching a goal state, and the agent needs to select the STOP action. This makes the reward function less sparse and allows the PPO agent to gain some positive reward sooner. We denote the policy with masked actions as PPO+Stop Forcing (PPO+SF).

4 Experiments

For the experiments we focus on two questions.

How do the algorithm-model pairs perform? For every CMDP, we report accuracy on the development and test sets for both PPO and PPO+SF and both CNN+BERT and ViLT. All models are randomly initialized. For every configuration, we randomly sample 10% of the training data as a held-out validation set kept unchanged throughout the experiments. We stop training using this validation set with early stopping and select the model with the best validation accuracy.

Table 1 shows the accuracy after training all algorithm-model pairs. We see that ViLT is performing better than CNN+BERT on all configurations except for SCATTER-SCRATCH. ViLT is especially outperforming CNN+BERT on the configurations using FLIPIT, for both TOWER and SCATTER, showing that having models with more representation capacity is important.

What types of mistakes do PPO models make? We sample 50 erroneous development examples from configurations in the SCATTER-FLIPIT and SCATTER-SCRATCH CMDPs, and analyze their mistakes. In SCATTER-SCRATCH trained with PPO, we found that for CNN+BERT, 76% of the mistakes are due to invalid actions, and 24% due to early termination. Among the invalid actions, 58% are due to trying to put an item that cannot fit in the box, 24% are due to trying to perform an action on a separator, and 18% due to trying to remove an object from a position that does not include an object. For other configurations, the error causes are similar.

5 Conclusion

We introduce the ℓGym reinforcement learning benchmark for natural language visual reasoning, based on a set of Python program annotations for thousands of human-written natural language statements grounded in visual environments. The statements present a high degree of compositionality and semantic diversity, and require complex relational reasoning about displayed sets of objects. The collection of environments in ℓGym are modular and present increasing levels of difficulty. We will release our annotated data and code.

References

Peter Anderson, Abhishek Das, and Qi Wu. Connecting language and vision to actions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 10–14, 2018.

Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. *arXiv preprint arXiv:1806.01946*, 2018.

Raffaella Bernardi and Sandro Pezzelle. Linguistic issues behind visual question answering. *Language and Linguistics Compass*, 15(6):e12417, 2021.

Valts Blukis, Nataly Brukhim, Andrew Bennett, Ross A Knepper, and Yoav Artzi. Following high-level navigation instructions on a simulated quadcopter with imitation learning. *arXiv preprint arXiv:1806.00047*, 2018.

Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. *arXiv preprint arXiv:2107.05612*, 2021.

Ben Bogin, Shivanshu Gupta, Matt Gardner, and Jonathan Berant. Covr: A test-bed for visually grounded compositional generalization with real images. *arXiv preprint arXiv:2109.10613*, 2021.

Satchuthananthavale RK Branavan, Harr Chen, Luke Zettlemoyer, and Regina Barzilay. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 82–90, 2009.

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. *arXiv preprint arXiv:1810.08272*, 2018.

Cédric Colas, Ahmed Akakzia, Pierre-Yves Oudeyer, Mohamed Chetouani, and Olivier Sigaud. Language-conditioned goal generation: a new approach to language grounding for rl. *arXiv preprint arXiv:2006.07043*, 2020.

Pradeep Dasigi, Matt Gardner, Shikhar Murty, Luke Zettlemoyer, and Eduard Hovy. Iterative search for weakly supervised semantic parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2669–2680, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1273. URL `https://aclanthology.org/N19-1273`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Keep drawing it: Iterative language-based image generation and editing. *arXiv preprint arXiv:1811.09845*, 2, 2018.

Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10304–10312, 2019.

Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

Omer Goldman, Veronica Latcinnik, Udi Naveh, Amir Globerson, and Jonathan Berant. Weakly-supervised semantic parsing with abstract examples. *arXiv preprint arXiv:1711.05240*, 2017.

Omer Goldman, Veronica Latcinnik, Ehud Nave, Amir Globerson, and Jonathan Berant. Weakly supervised semantic parsing with abstract examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1809–1819, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1168. URL https://aclanthology.org/P18-1168.

Nitish Gupta, Sameer Singh, and Matt Gardner. Enforcing consistency in weakly supervised semantic parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 168–174, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.22. URL https://aclanthology.org/2021.acl-short.22.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

Austin W Hanjie, Victor Y Zhong, and Karthik Narasimhan. Grounding language to entities and dynamics for generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 4051–4062. PMLR, 2021.

Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018.

Drew A Hudson and Christopher D Manning. Gqa: a new dataset for compositional question answering over real-world images. *arXiv preprint arXiv:1902.09506*, 3(8), 2019.

Minqi Jiang, Jelena Luketina, Nantas Nardelli, Pasquale Minervini, Philip HS Torr, Shimon Whiteson, and Tim Rocktäschel. Wordcraft: An environment for benchmarking commonsense agents. *arXiv preprint arXiv:2007.09185*, 2020.

Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. Language as an abstraction for hierarchical deep reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kim21k.html.

Julia Kiseleva, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, et al. Interactive grounded language understanding in a collaborative environment: Iglu 2021. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 146–161. PMLR, 2022.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *arXiv preprint arXiv:1611.00020*, 2016.

Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. Visually grounded reasoning across languages and cultures. *arXiv preprint arXiv:2109.13238*, 2021a.

Nan Liu, Shuang Li, Yilun Du, Joshua B Tenenbaum, and Antonio Torralba. Learning to compose visual relations. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021b.

Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

Dipendra Misra, John Langford, and Yoav Artzi. Mapping instructions and visual observations to actions with reinforcement learning. *arXiv preprint arXiv:1704.08795*, 2017.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*, 2015.

Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019a.

Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019b.

Khanh X Nguyen, Dipendra Misra, Robert Schapire, Miroslav Dudík, and Patrick Shafto. Interactive learning from activity description. In *International Conference on Machine Learning*, pages 8096–8108. PMLR, 2021.

Juan Pavez, Héctor Allende, and Héctor Allende-Cid. Working memory networks: Augmenting memory networks with a relational reasoning module. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1000–1009, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1092. URL `https://aclanthology.org/P18-1092`.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Rajkumar Ramamurthy, Rafet Sifa, and Christian Bauckhage. Nlpgym–a toolkit for evaluating rl agents on natural language processing tasks. *arXiv preprint arXiv:2011.08272*, 2020.

Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 290–298, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Seitaro Shinagawa, Koichiro Yoshino, Sakriani Sakti, Yu Suzuki, and Satoshi Nakamura. Interactive image manipulation with natural language instruction commands. *arXiv preprint arXiv:1802.08645*, 2018.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, 2017.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1644. URL https://aclanthology.org/P19-1644.

Hao Tan and Mohit Bansal. Object ordering with bidirectional matchings for visual reasoning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 444–451, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2071. URL https://aclanthology.org/N18-2071.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Guangyu Robert Yang, Igor Ganichev, Xiao-Jing Wang, Jonathon Shlens, and David Sussillo. A dataset and architecture for visual reasoning with a working memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 714–731, 2018.

Yiqun Yao, Jiaming Xu, Feng Wang, and Bo Xu. Cascaded mutual modulation for visual reasoning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 975–980, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1118. URL https://aclanthology.org/D18-1118.

Table 2: Data statistics per CMDP configuration and data split. The number of MDPs corresponds to the number of contexts under each configuration. For `FLIPIT`, "Init." corresponds to the total number of initial states.

	`TOWER-SCRATCH`	`TOWER-FLIPIT`		`SCATTER-SCRATCH`	`SCATTER-FLIPIT`	
	MDPs	MDPs	Init.	MDPs	MDPs	Init.
Train	989	1,910	5,704	1,241	2,340	6,696
Dev	163	317	676	87	164	313
Test	324	619	1,383	155	285	591
Total	1,476	2,846	7,763	1,483	2,789	7,600

A The ℓGym Data

The data used for ℓGym is based on the Natural Language Visual Reasoning corpus [NLVR; Suhr et al., 2017]. The NLVR data was initially collected as a supervised learning benchmark. We formalize an interactive task on top of the NLVR data and collect additional annotations for reward computation.

A.1 The NLVR Corpus

NLVR includes human-written natural language statements paired with synthetic images. Each pair is annotated with the boolean truth-value of the statement with regard to the image (i.e., `True` if the statement is true with regard to the image, or `False` otherwise). The images are designed to support complex reasoning, including about spatial and set relations. The original learning task posed by NLVR is to classify statement-image pairs as `True` to indicate the statement is true with regard to the image, or `False` otherwise. Various approaches were developed to address the NLVR challenge [Suhr et al., 2017, Tan and Bansal, 2018, Goldman et al., 2018, Pavez et al., 2018, Yao et al., 2018, Hudson and Manning, 2018, Perez et al., 2018, Dasigi et al., 2019, Gupta et al., 2021], and a separate version using photos was also released [Suhr et al., 2019].[1] NLVR is roughly balanced, and the current state-of-the-art using the raw images is 69.7% accuracy [Tan and Bansal, 2018], leaving significant room for improvement and illustrating the reasoning challenges the NLVR data presents.

Qualitative analysis released with the data shows a more diverse representation of semantic and compositional phenomena compared to similar corpora (Table 2 in [Suhr et al., 2017]), including requiring joint visual-linguistic reasoning about spatial relations, quantities, and sets of objects. NLVR also provides an underlying structure representation for every image, which supports easy manipulation of images. The combination of simple interface for image manipulation with complex reasoning via natural language makes NLVR a strong candidate to support an interactive benchmark environment.

The original dataset is divided into four splits for training, development, public testing, and hidden testing. We follow the original splits for the training and development sets. Following the rest public release of the hidden testing set, we merge the public and hidden testing sets into a single public test split.

A.2 Annotations for Automatic Evaluation

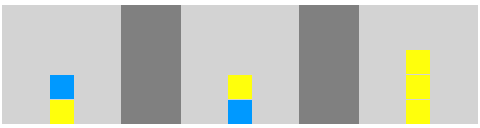
The NLVR annotations include the truth-value of each statement with regard to the images paired with it in the data. Once we manipulate the image (i.e., change the state in our interactive environment), the truth-value annotation does hold. A key challenge for creating an interactive environment using the NLVR data is the need for accurate evaluation of the natural language statement for *every* possible state (i.e., image), as required for reward computation (Section 2).

We address this challenge by annotating all statements with executable Python programs representing their meaning, denoted as $\mathcal{E}^{\bar{x}}$ in Section 2. The Python programs operate on the underlying

[1] We do no use the photographic NLVR2 in this work.

structured representation. Each program returns `True` for every image that satisfies the constraints specified in the corresponding statement, and `False` otherwise. In general, there are many states that satisfy any given statement, many more than provided with the original NLVR images.

The programs are written using an API defined over the structured representations. We base the API design on the logical ontology designed for NLVR's structured representations by [Goldman et al., 2017]. Figure 2 shows two examples of logical forms paired with the corresponding text statement and an image.



There are two towers with the same height but their base is not the same in color.

```
exist(filter_obj(all_boxes, lambda x:  x.is_tower() and exist(filter_obj(all_boxes,
lambda y:  y.is_tower() and count(x.all_items_in_box()) == count(y.all_items_in_box())
and get_set_colors(filter_obj(y.all_items_in_box(), is_bottom)) !=
get_set_colors(filter_obj(x.all_items_in_box(), is_bottom))))))
```

There is a box with all 3 different colors and a black triangle touching the wall with its top.

```
exist(filter_obj(all_boxes, lambda x:  count(get_set_colors(x.all_items_in_box())) == 3
and exist(filter_obj(x.all_items_in_box(), lambda y:  is_black(y) and is_triangle(y)
and is_touching_wall(y, Side.TOP)))))
```

Figure 2: Example sentences, a corresponding logical form (Python program), and an example image from the dataset. The sentence and logical form are `True` for the top statement, and `False` for the bottom statement.

We use the freelancing platform Upwork[2] for annotation. We recruit three annotators based on a preliminary screening of their fluency in English and competency in Python programming. We de-duplicate the naturally occurring sentences in the data, collect 2,666 annotations at a total cost of $3,756, and keep 2,661 valid annotations.

All the sentences in the dataset are randomly distributed to annotators with an example pair of image and boolean. Every sentence is annotated with a logical form by one annotator. Each logical form is evaluated against a corresponding hidden validation set, and it needs to pass all the tests.

[2] `https://www.upwork.com`

B Related work

Interactive grounded language learning Grounding language learning in an multimodal and interactive framework has been a challenging and long-standing goal [Anderson et al., 2018]. The competition Kiseleva et al. [2022] focused on building collaborative agents provided with natural language instructions, and Shridhar et al. [2020] studied instruction following in an embodied environment. Nguyen et al. [2021] studies interactive learning for agents mapping language request to executions. Our work focuses on both vision and language modalities, while focusing on learning to reason interactively based on visual-linguistic environments displaying complex relations.

Reinforcement Learning and Imitation Learning with Language Recent work in Reinforcement Learning (RL) and Imitation Learning investigated a range of natural language-related tasks [Chevalier-Boisvert et al., 2018, Jiang et al., 2020], such as visual question answering (VQA) [Hudson and Manning, 2018], image captioning [Ren et al., 2017], semantic parsing [Liang et al., 2016], vision-language navigation [Nguyen et al., 2019b, Blukis et al., 2018], text games [Narasimhan et al., 2015], instruction following [Blukis et al., 2021, Nguyen et al., 2019a]. Several work focused on using language to represent instructions or to condition the goal [Branavan et al., 2009, Bahdanau et al., 2018, Colas et al., 2020]. There has also been a trend of reward modeling in language task [Bahdanau et al., 2018, Leike et al., 2018]. Language is also used to improve generalization [Hanjie et al., 2021]. Ramamurthy et al. [2020] is an unimodal framework for sequence tagging, multi-label classification and question answering. In Jiang et al. [2019], language is used to study compositional task learning. By nature, the compositional nature of language enables the study at different levels of abstraction and combinatorial generalization.

Visual-linguistic reasoning Grounded reasoning has been studied in a visual-linguistic context in a range of recent work, using synthetic images [Suhr et al., 2017, Johnson et al., 2017, Yang et al., 2018] or real images [Suhr et al., 2018, Liu et al., 2021a] with human annotations, or real images with synthetic language [Hudson and Manning, 2019]. More recently, there have been work focusing on investigating the compositionality of visually grounded natural language, either by generating and editing synthetic images [Liu et al., 2021b], or instead generating question-answer pairs about real images [Bogin et al., 2021]. There has been work investigating linguistic issues about current VQA systems [Bernardi and Pezzelle, 2021]. Current visual-linguistic systems still present a range of challenges, for instance reasoning at different abstraction levels, credit assignment [Minsky, 1961], dealing with language ambiguity, negation, etc. Our work incorporate these ideas and requires reasoning about the alignments of inputs parts across modalities.