# Modular Framework for Visuomotor Language Grounding

Kolby Nottingham Litian Liang Daeyun Shin Charless C. Fowlkes Roy Fox
Sameer Singh
University of California Irvine
{knotting,litianl1,daeyuns,fowlkes,royf,sameer}@uci.edu

## Abstract

*Natural language instruction following tasks serve as a valuable test-bed for grounded language and robotics research. However, data collection for these tasks is expensive and end-to-end approaches suffer from data inefficiency. We propose the structuring of language, acting, and visual tasks into separate modules that can be trained independently. Using a Language, Action, and Vision (LAV) framework removes the dependence of action and vision modules on instruction following datasets, making them more efficient to train. We also present a preliminary evaluation of LAV on the ALFRED task for visual and interactive instruction following.*

## 1. Introduction

Many state of the art natural language systems are conditioned solely on language input [5, 7, 8]. However advanced language understanding requires that language is grounded in vision and interaction [3, 4]. Interactive and visual instruction following tasks provide a test-bed for developing methods that ground language in vision and actions. These types of tasks are also interesting from a robotics perspective. Ideally robots that interact with humans in the real world will support a natural language interface. Thus interactive and visual natural language instruction following tasks also work towards accomplishing this goal.

Typical approaches to instruction following tasks perform end-to-end learning with a deep neural network [2, 10]. However, gathering expert demonstrations paired with natural language instructions is costly and datasets are typically small. End-to-end baselines have performed poorly on interactive instruction following tasks, and many approaches to improve on baselines incorporate some modularization. Corona et al. [6] train separate modules for each type of high level task to be completed (e.g., go to, pick up), and Singh et al. [11] train a perception module separate from their action module. However, all modules in both of these methods are still dependent on the instruction following dataset. Recently, Saha et al. [9] introduced a method called Modular Vision and Language (MoViLan) that is most similar to our method. MoViLan also trains modules independently to work together at test time, but it is evaluated on a simpler version of the ALFRED task.

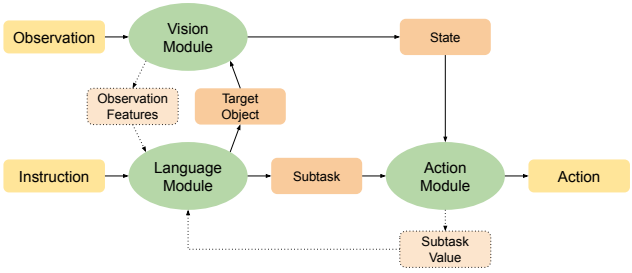


Figure 1. The Language Action Vision (LAV) framework. Modules are trained independently to minimize dependence on natural language instructions and expert demonstrations. Items outlined with a dotted line are suggested improvements.

Simple end-to-end approaches for solving instruction following tasks ignore the fact that much can be learned about actions and vision independent of instruction datasets. We make this separation explicit and propose a Language, Action, and Vision (LAV) framework that can train each module independently, thus removing the action and vision modules' dependence on any instruction following dataset. Our evaluation of LAV on the ALFRED task indicates that it is able to significantly outperform end-to-end baselines.

## 2. LAV Framework

Key to our LAV framework (see Figure 1) is that the action and vision modules can be trained independently of natural language instructions and expert trajectories. This usually means defining a set of possible action subtasks $g$ and target objects $o$. Given the natural language instruction $x_{lang}$, the main task of the **Language Module** $L$ is to identify the subtask and target object for the action and vision modules $L : x_{lang} \rightarrow o, g$.

With a set of possible objects, the LAV **Vision Module** $V$ can be trained to extract state features $s$ given visual ob-

servations $x_{obs}$ and a target object $o$, independent of the instruction following dataset $V : x_{obs}, o \rightarrow s$. The vision module benefits from this independence because computer vision datasets are typically cheaper to collect and more plentiful than instruction following datasets. Additionally, many instruction following tasks run in simulation making vision datasets especially simple to collect and label.

Finally the LAV **Action Module** $A$ can learn to complete subtasks with arbitrary target objects in a multi-task learning setting. Given a subtask $g$, target object $o$, and state featues $s$, it learns to predict actions $a$ to complete the subtask $A : g, o, s \rightarrow a$. Multi-task robot learning is a popular research area and many approaches exist for solving this problem. For example, if the tasks are running in a simulator, the action module can be trained via multi-task reinforcement learning.

Note that the output from each independent module can also augment the input to other modules to improve test time performance as illustrated by the dotted items in Figure 1. For example, the language module can be informed by what objects are visible in the scene ($L : x_{lang}, s' \rightarrow o, g$ where $V' : x_{obs} \rightarrow s'$). Additionally, the language module can use the value of the current state under a specific subtask to determine the output with the highest chance of success ($L : x_{lang}, v \rightarrow o, g$ where $A' : s' \rightarrow v$).

## 3. Evaluation

To test our LAV framework, we develop a prototype implementation to evaluate on the ALFRED task [10].

### 3.1. Implementation

The LAV language module for our implementation is finetuned from a T5 language model [8]. We train the model to generate sequences of (subtask, target object) pairs based on natural language goal instructions. The subtasks we use consist of the seven high level actions defined by ALFRED (pick up, place, toggle, clean, cool, heat, and slice), and the objects consist of all of the object types used as target objects in the training set.

We train three vision networks that make up our LAV vision module. Given an RGB observation, these networks output an object type segmentation, a depth map, and an obstacle indicator. All three models were trained via supervised learning from datasets collected from the AI2THOR simulator. Depth maps are used in with segmentations to estimate the relative position to a target object.

Using the target object's position estimated by the vision module, the LAV action module first navigates toward the target object and then executes low level actions corresponding to the current subtask predicted by the language module. Navigation is a simple depth first search around obstacles toward the target object's estimated position.

| | **Test Data Seen** | | | | **Test Data Unseen** | | | |
|---|---|---|---|---|---|---|---|---|
| | SR | PWSR | GC | PWGC | SR | PWSR | GC | PWGC |
| Baseline | 4.0 | 2.0 | 9.4 | 6.3 | 0.4 | 0.1 | 7.0 | 4.3 |
| LWIT [1] | 30.9 | 25.9 | 40.5 | 36.8 | 9.4 | 5.6 | 20.9 | 16.3 |
| **LAV** | 13.4 | 6.3 | 23.2 | 13.2 | 6.3 | 3.1 | 17.3 | 10.5 |

Table 1. Percent success rate (SR), path weighted success rate (PWSR), goal condition success rate (GC), and path weighted goal condition success rate (PWGC) from ALFRED's public leaderboard. Seen test data indicates scenes included in the training set but with novel tasks while unseen data used novel scenes.

| | SR | PWSR | GC | PWGC |
|---|---|---|---|---|
| L & V Oracles | 25.2 | 8.5 | 32.5 | 11.3 |
| V Oracle | 18.4 | 6.3 | 26.3 | 8.6 |
| L Oracle | 15.4 | 7.3 | 24.8 | 15.3 |
| **LAV** | 12.7 | 5.9 | 23.4 | 13.7 |

Table 2. Metrics comparing versions of LAV on ALFRED's validation data (seen). Oracles replace a module with ground truth.

For example, after picking up a dirty bowl the language module predicts the "clean" subtask and the "sink" target object. The vision module identifies the sink and provides an estimated position. The action module navigates toward the sink. Once the estimated position is in range, the action module performs the actions "place in sink", "toggle sink on", "toggle sink off", and "pick up bowl".

### 3.2. Results

We compare our LAV framework to the ALFRED baseline and the current state of the art (LWIT [1]) in Table 1. We also provide results when replacing our language and vision modules with ground truth oracles in Table 2. The LAV framework significantly outperforms the baseline across all metrics. While it doesn't outperform LWIT in its current form, LAV suffers less from the transfer to novel test scenes than LWIT does. Also note that LAV only achieves 25.2% SR while using language and vision oracles. This indicates that the current point navigation search is a major weak point of our implementation. We plan to replace the current action module with a reinforcement learning agent in the future which will further improve performance.

## 4. Conclusion

The LAV framework demonstrates the advantage of training vision and action modules independent of instruction datasets. Doing so allows those modules to train on much cheaper and more abundant data. The language module is able to predict subtasks and target objects from instruction data without needing to learn vision and low level actions as well. In the future we plan to apply LAV to other tasks such as iGibson [13] and AI2-THOR Rearrangement [12] and improve upon the current LAV implementation.

## References


[1] Alfred leaderboard. https://leaderboard.allenai.org/alfred/submissions/public. Accessed: 2021-05-14. 2

[2] 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*, pages 3674–3683, 2018. 1

[3] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, 2020. 1

[4] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. Experience grounds language. *arXiv preprint arXiv:2004.10151*, 2020. 1

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[6] Rodolfo Corona, Daniel Fried, Coline Devin, Dan Klein, and Trevor Darrell. Modularity improves out-of-domain instruction following. *arXiv preprint arXiv:2010.12764*, 2020. 1

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 1, 2

[9] Homagni Saha, Fateme Fotouhif, Qisai Liu, and Soumik Sarkar. A modular vision language navigation and manipulation framework for long horizon compositional tasks in indoor environment. *arXiv preprint arXiv:2101.07891*, 2021. 1

[10] 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*, pages 10740–10749, 2020. 1, 2

[11] Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. Moca: A modular object-centric approach for interactive instruction following. *arXiv preprint arXiv:2012.03208*, 2020. 1

[12] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[13] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020. 2