On the Limits of Evaluating Embodied Agent Model Generalization Using Validation Sets

Hyounghun Kim[1] **Aishwarya Padmakumar**[2] **Di Jin**[2]
Mohit Bansal[1,2] **Dilek Hakkani-Tur**[2]
[1]UNC Chapel Hill [2]Amazon Alexa AI
{hyounghk, mbansal}@cs.unc.edu {padmakua, djinamzn, hakkanit}@amazon.com

Abstract

Natural language guided embodied task completion is a challenging problem since it requires understanding natural language instructions, aligning them with egocentric visual observations, and choosing appropriate actions to execute in the environment to produce desired changes. We experiment with augmenting a transformer model for this task with modules that effectively utilize a wider field of view and learn to choose whether the next step requires a navigation or manipulation action. We observed that the proposed modules resulted in improved, and in fact state-of-the-art performance on an unseen validation set of a popular benchmark dataset, ALFRED. However, our best model selected using the unseen validation set underperforms on the unseen test split of ALFRED, indicating that performance on the unseen validation set may not in itself be a sufficient indicator of whether model improvements generalize to unseen test sets. We highlight this result as we believe it may be a wider phenomenon in machine learning tasks but primarily noticeable only in benchmarks that limit evaluations on test splits, and highlights the need to modify benchmark design to better account for variance in model performance.

1 Introduction

Language guided embodied task completion is an important skill for embodied agents requiring them to follow natural language instructions to navigate in their environment and manipulate objects to complete tasks. Natural language is an easy medium for users to interact with embodied agents and effective use of natural language instructions can enable agents to navigate more easily in previously unexplored environments, and complete tasks involving novel combinations of object manipulations. Vision and language navigation benchmarks (Anderson et al., 2018; Thomason et al., 2019; Ku et al., 2020) provide an agent with natural language route instructions and evaluate their ability to follow these to navigate to a target location. It requires agents to have a deep understanding of natural language instructions, ground these in egocentric image observations and predict a sequence of actions in the environment. Other benchmarks study the manipulation and arrangement of objects (Bisk et al., 2016; Wang et al., 2016; Li et al., 2016; Bisk et al., 2018) - another crucial skill to complete many tasks that users may desire embodied agents to be able to complete. These tasks additionally require agents to reason about the states of objects and relations between them. Language guided embodied task completion benchmarks (Shridhar et al., 2020; Kim et al., 2020; Padmakumar et al., 2022) combine these skills – requiring agents to perform both navigation and object manipulation/arrangement following natural language instructions.

In this paper, we explore a challenging navigation and manipulation benchmark, ALFRED (Shridhar et al., 2020), where an agent has to learn to follow complex hierarchical natural language instructions to complete tasks by navigating in a virtual environment and manipulating objects to produce desired state changes. The ALFRED benchmark provides a training dataset of action trajectories taken by an embodied agent in a variety of simulated indoor rooms paired with hierarchical natural language instructions describing the task to be accomplished and the steps to be taken to do so. For validation and testing of models, there are two splits each - seen and unseen splits. The seen validation and testing splits consist of instructions set in the same rooms as those in the training set, while the unseen splits consist of instructions set in rooms the agent has never seen before, with rooms in the unseen test set being different from those in the train and unseen validation set. Performance on the unseen validation and test sets are considered to be the best indicators of whether a model can really solve the task as the agent must operate in

Proceedings of the Third Workshop on Insights from Negative Results in NLP, pages 113 - 118

a completely novel floorplan, and cannot rely for example on memorized locations of large objects such as a fridge or a sink. Additionally, the ground truth action sequences are not publicly available for the seen and unseen test sets, and participants must submit prediction acted sequences on the test sets to an evaluation server where they are privately evaluated to obtain test performance. The evaluation server limits the number of submissions that can be made from an account to one per week to discourage directly tuning hyperparameters of a model on the test set. It is expected that following standard procedure in training machine learning models, one may use the validation sets to evaluate models trained with different hyperparameters, or ablating different components on the validation sets and only evaluate the best model on the test sets. Since ideally we would want a model to perform well on the unseen test set, it is reasonable to use success rate on the unseen validation set as a metric to choose which model is to be submitted for evaluation on the unseen test set.

One technique previously demonstrated to improve performance on ALFRED is the use of a multi-view setup (Nguyen et al., 2021; Kim et al., 2021) where an agent turns or moves its head in place at every time step to obtain additional views before deciding what action to take. In contrast to current models that simply concatenate features from each view, we use view-action matching - explicitly aligning embeddings of actions with embeddings of corresponding views - and using a score from fusing these aligned embeddings to select the next action to be taken. This is inspired by a dominant paradigm for modeling visual navigation tasks called viewpoint selection (Fried et al., 2018) where an agent predicts the next action by examining the resultant views each of those would produce and selecting the desired future view. Viewpoint selection is possible in some simulators such as R2R where the environment does not get altered by the agent's actions and the agent's movement is confined to a fixed grid. The ALFRED dataset uses the AI2-THOR simulator which supports a wider action space, physics modeling for movement and a more dynamic environment including irreversible actions. Hence, it is not possible to obtain the view that would result from an action without taking it, preventing direct application of viewpoint selection. Additionally, the agent must decide at each time step whether to perform navigation or manipulation actions. In contrast to prior work that uses a single classifier layer over all possible actions treating them equally, we propose a gate module which gives a higher weight to actions of a more relevant action type.

We follow standard experimental procedure training our modified models on the train split and using success rate on the unseen validation split to compare to baselines and perform ablation studies. On this set, the proposed model equipped with the aforementioned modules outperforms the state-of-the-art multi-view setup approaches and the ablation study shows each proposed module helps improve the model's performance.

However, we observe an unexpected and large performance gap between the unseen validation and test data splits. Our model outperforms state-of-the-art baseline models on the unseen validation split, but performs worse than them on the unseen test split. We hypothesize that it may be possible to overfit hyperparameters and design choices to one set of unseen environments (the unseen validation) and hence success on one such set of unseen environments is insufficient to guarantee that a model will generalize to another set of unseen environments (the unseen test). We report this finding as we believe this situation is likely more common during development on machine learning benchmarks, but such intermediate results are unlikely to be published. Instead after a poor result on a test set, it is likely that researchers continue further model modifications until a model setting is obtained that performs well on the test set. We believe that such models are likely overfitting to the test set of the benchmark and may not generalize well to a new test set.

2 Dataset & Environment

In this paper, we focus on improving models for the ALFRED (Shridhar et al., 2020) benchmark. ALFRED is built using the AI2-THOR simulator (Kolve et al., 2017) which consists of 120 indoor scenes across 4 types of rooms. Scenes also contain a diverse set of objects that are rearranged in different configurations for each trajectory in the dataset. In ALFRED, a agent is given a high level natural language goal statement (*"Put a chilled pan on the counter"*) as well as step by step natural language instructions corresponding to subgoals to be completed in order for achieving the goal (*"Turn around and cross the room and then go right and*

turn to the left to face the stove ... Put the pan down on the counter to the right of the toaster"). An agent has access to all these instructions at the start of the task and then has to iteratively predict navigation and manipulation actions in the environment based on egocentric image observations to complete subgoals in order. An agent must predict between a discrete set of possible navigation and manipulation actions, and predict a segmentation mask for the object to be manipulated if a manipulation action is predicted. The performance for an agent is evaluated by comparing the final states of the objects at the end of the action trajectory executed by the agent to the states of the objects at the end of the ground truth trajectory.

3 Model

We employ a vision-language transformer, LXMERT (Tan and Bansal, 2019) as the base architecture for our model. We encode the language input using a learned word embedding and transformer layer, and action history using a linear layer. Following Pashevich et al. (2021), we extract image features using a faster R-CNN (Ren et al., 2015) pretrained on images from the AI2-THOR simulator, and average-pool features of regions into a single vector. The visual and action features are first combined via a liner layer, and then fused with language features through a cross modal transformer layer.

View-Action Matching. We collect the multiple views (front, left, right, up, down) and go through the aforementioned process to obtain a feature V_i from the cross modal transformer for each view, and compute its matching score M_i with the corresponding action embedding A_i using a feedforward network.

Action-Type Gate. We additionally learn a gate vector using a linear layer over features of all views at the current time step to better distinguish between navigation and non-navigation actions. This layer is trained to predict high weights for actions of the same type as the ground truth action and low weights otherwise. The predicted weights are multiplied pointwise with match scores M_i and the action with the highest resultant score is selected. For example, if the ground truth action at a particular time step is `Move forward`, the gate will ensure that a prediction of `ToggleOff` which is a non-navigation action will receive a higher loss than a prediction of `Turn Right`, which is also an incorrect action but of the same type as the ground truth action (navigation).

	Model	Wide View	View-Act Matching	Act-Type Gate	Success Rate (%)
1	Base LXMERT Architecture	✗	✗	✗	4.7
2	VAM (Ours)	✓	✗	✗	9.3
3	VAM (Ours)	✓	✓	✗	11.8
4	VAM (Ours)	✓	✓	✓	13.8

Table 1: Performance improvement from wide view, view-action matching and action type gate modules on the ALFRED validation unseen split.

Loss. The model is trained via cross-entropy losses for action (teacher-forcing) and object type.

4 Experiments

Implementation & Training Details. We use 2 language and 2 cross-modal LXMERT layers for the model, and use 768 as the hidden size. We use AdamW (Loshchilov and Hutter, 2018) as the optimizer with the learning rate 1×10^{-5}. All of the experiments are run on AWS 'p3.16xlarge' EC2 instances running Ubuntu 18.04. We employ PyTorch (Paszke et al., 2017) to build our models.

Data Splits. Following Shridhar et al. (2020), we train our models on the train split and use success rate on the unseen validation split to perform model selection, and determine whether our model changes are likely to improve over existing state of the art models. We used the validation splits to evaluate the efficacy of variants of the transformer architecture, number of layers and number of epochs of training to use. We then submitted predictions from the best performing model on the unseen validation split to the evaluation server to obtain scores on the test sets.

Evaluation Metrics. We report two evaluation metrics from Shridhar et al. (2020) on validation and test splits. Success rate (SR) measures the fraction of episodes whether the predicted model trajectory results in all object state changes produced by the ground truth action trajectory. Goal Condition Success Rate (GC) measures the fraction of such desired state changes across all episodes that were accomplished by model-predicted trajectories.

Model Comparison. Recently, the best performing models on the ALFRED benchmark make use of semantic map representations of the environment (Blukis et al., 2021). However, these rely on pre-exploration of the environment to build a semantic map, rather than utilizing language instruc-

Subgoals	Wide View	(+) View-Act Matching	(+) Act-Type Gate
CleanObject	81.4	89.4	91.2
CoolObject	100.0	100.0	100.0
GotoLocation	62.0	66.2	67.1
HeatObject	100.0	100.0	98.5
PickupObject	69.2	68.5	68.5
PutObject	66.6	71.2	68.3
SliceObject	62.2	61.3	69.4
ToggleObject	51.4	42.2	41.6

Table 2: Success rate (%) of the sub-goal tasks on the ALFRED validation unseen split.

tions to directly navigate to target objects. Therefore, we focus on comparing our model with other multi-view setup models that are the state-of-the-art among non-SLAM models. LWIT (Nguyen et al., 2021) predicts an initial actions from an selected instruction alone and integrates the actions sequence with visual information to generate final actions to take. ABP (Kim et al., 2021) factorizes the model into interactive perception and action policy modules for adapting to two different tasks (the former needs a pixel-level and the latter requires a global information). However, although they employ multi-view setup, the information from each view collapses into one integrated feature. On the other hand, our model exploit each view directly to keep the useful clues without any loss.

5 Results

We first evaluate the utility of each modeling change on the unseen validation set of ALFRED. As shown Table 1, we gain 4.6% on success rate from adding a wider field of view, an additional 2.5% from view-action matching and a further 2% from action type gating. We observe a variance of 3% in success rate of the same type of model trained with different random seeds so we consider a 4.6% improvement to be sufficiently large to be unlikely from pure variance.

Sub-Goal Performance. Considering the proportion of GotoLocation to the total number of sub-goal tasks (i.e., 48%) and its role of bridging other sub-goal tasks, navigation is very crucial ability for a agent to successfully perform this challenging ALFRED task. As shown in Table 2, our full view-action matching (VAM) model improves the performance of GotoLocation task by 5.1% while also improves performance for some of other sub-goal tasks. This performance boost could attribute to the agent's ability to figure out where to go (View-Action Matching) and what to do (Action-Type Gate).

Validation-Test Performance Gap. When we compare to other baselines in Table 3, although our model outperforms other state-of-the-art models on the unseen validation split by a large margin, its performance on the unseen test split is poorer, whereas the reverse trend is seen with ABP (Kim et al., 2021). This suggests that good performance from a model on an unseen validation set may not be a good method to determine whether model changes are likely to generalize to another unseen test set.

This lack of generalization is more likely in current embodied learning tasks such as vision-and-language navigation or embodied task completion in comparison to other machine learning tasks due to the way unseen test sets are defined in embodied learning tasks. While ALFRED in particular does not introduce new object categories at test time, both validation and test unseen environments are visually different, by design from the training environment and from each other. When we compare models on the validation set, we hope that an increase in performance denotes a model that is more capable of generalizing to *any* unseen environment. However, it may only be the case that the model only generalizes better to the particular visual differences present in the unseen validation environment.

When the benchmark limits access to the test set, as in ALFRED, when dealing with a model that demonstrates variance when trained with different random seeds, hyperparameters and across training epochs, it is natural to choose the setting that results in the highest performance on the unseen validation set. However, a different setting may in fact be optimal for the unseen test set due to visual differences. While such a design is likely significantly more computationally expensive, it may be necessary to redesign benchmarks to take an average of performance from a few different variants of a model to reliably rank different modelling methods, instead of using scores from individual runs. We may also want to re-evaluate the value of keeping a test set private, as in the case of ALFRED that avoids prevents allowing models to overfit on the test set, but also makes it difficult to analyze the robustness of model performance between the validation and test sets. We would also like to encourage the reviewing community to enable the publication of modelling techniques whose performance is in the same ball-

Split	Model	Seen		Unseen	
		SR	GC	SR	GC
Val	LWIT	33.70	43.10	9.70	23.10
	ABP	42.93	50.45	12.55	25.19
	VAM (Ours)	40.9	47.9	13.8	28.1
Test	LWIT	29.16	38.82	8.37	19.13
	ABP	44.55	51.13	15.43	24.76
	VAM (Ours)	35.42	43.98	8.57	20.69

Table 3: Success rate (%) on the ALFRED evaluation splits (GC: Goal-Condition). Our model outperforms the state-of-the-art multi-view setup models on validation splits but not test splits.

park as existing state-of-the-art models, but novel in some way, as opposed to solely relying on a model achieving a top score on a leaderboard as a criterion for publication, as this limits the development that could be made using these alternative modeling approaches.

6 Conclusion

We attempted to improve a transformer model for embodied task completion by enabling it to effectively uses multiple views via view-action matching and action-type gating. Our view-action matching module computes a matching score between each a view and the embedding of the action used to generate it, and the gate module gives a higher weight to a more appropriate action type. While our model outperformed relevant baselines on the ALFRED unseen validation split, the trend was reversed on the unseen test split, suggesting that it may not be possible to over-utilize a validation split when making model selection choices so that the resultant model does not perform well on the test split. We choose to publish this result as we believe this phenomenon is likely more common than reported with machine learning benchmarks, but only noticeable to researchers when working on a benchmark with limited access to the test set. We additionally hope that our work encourages the publication of promising modelling approaches that do not work as reliably as expected, so that these can act as a guide to researchers to better inform their future directions.

Acknowledgments

We thank the reviewers for their helpful comments. This work was partially done while Hyounghun Kim was interning at Amazon Alexa AI and later extended at UNC, where it was supported by NSF Award 1840131 and DARPA KAIROS Grant FA8750-19-2-1004. The views contained in this article are those of the authors and not of the funding agency.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018. 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.

Yonatan Bisk, Daniel Marcu, and William Wong. 2016. Towards a dataset for human computer communication via grounded language acquisition. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*.

Yonatan Bisk, Kevin J Shih, Yejin Choi, and Daniel Marcu. 2018. Learning interpretable spatial operations in a rich 3d blocks world. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. 2021. A persistent spatial semantic representation for high-level natural language instruction execution. In *5th Annual Conference on Robot Learning*.

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 *NeurIPS*.

Byeonghwi Kim, Suvaansh Bhambri, Kunal Pratap Singh, Roozbeh Mottaghi, and Jonghyun Choi. 2021. Agent with the big picture: Perceiving surroundings for interactive instruction following. In *Embodied AI Workshop CVPR*.

Hyounghun Kim, Abhaysinh Zala, Graham Burri, Hao Tan, and Mohit Bansal. 2020. Arramon: A joint navigation-assembly instruction interpretation task in dynamic environments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3910–3927.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412.

Shen Li, Rosario Scalise, Henny Admoni, Stephanie Rosenthal, and Siddhartha S Srinivasa. 2016. Spatial references and perspective in natural language instructions for collaborative manipulation. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 44–51. IEEE.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Van-Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. 2021. Look wide and interpret twice: Improving performance on interactive instruction-following tasks. *IJCAI*.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2022. Teach: Task-driven embodied agents that chat. *AAAI*.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. 2021. Episodic Transformer for Vision-and-Language Navigation. In *ICCV*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2019. Vision-and-dialog navigation. In *Conference on Robot Learning (CoRL)*.

Sida I Wang, Percy Liang, and Christopher D Manning. 2016. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2368–2378.