Can LLM's Generate Human-Like Wayfinding Instructions? Towards Platform-Agnostic Embodied Instruction Synthesis

Anonymous ACL submission

Abstract

We present a novel approach to automatically synthesize *"wayfinding instructions"* for an embodied robot agent. In contrast to prior approaches that are heavily reliant on human-annotated datasets designed exclusively for specific simulation platforms, our algorithm uses *in-context learning* to condition an LLM to generate instructions using just a few references. Using an LLM-based Visual Question Answering strategy, we gather detailed information about the environment which is used by the LLM for instruction synthesis. We implement our approach on multiple simulation platforms including Matterport3D, AI Habitat and ThreeDWorld, thereby demonstrating its platform-agnostic nature. We subjectively evaluate our approach via a user study and observe that 83.3% of users find the synthesized instructions accurately capture the details of the environment and show characteristics similar to those of human-generated instructions. Further, we conduct zero-shot navigation with multiple approaches on the REVERIE dataset using the generated instructions, and observe very close correlation with the baseline on standard success metrics ($<$ 1% change in SR), quantifying the viability of generated instructions in replacing human-annotated data. To the best of our knowledge, ours is the first LLM-driven approach capable of generating *"human-like"* instructions in a platform-agnostic manner, without requiring any form of training.

1 Introduction

In embodied navigation tasks, language is primarily used to convey *wayfinding instructions* to an agent operating in a simulation platform. These instructions convey the path that the agent should take to reach a target location. Generating these instructions usually takes place in the form of creating datasets that require several human annotation hours (Qi et al., 2020a; Anderson et al., 2018a; Padmakumar et al., 2022). In addition, the current datasets are exclusive to the embodied simulation platform in which the agent operates, preventing the transfer of instruction-following approaches across platforms. For instance, instructions based on the Matterport3D simulator (Chang et al., 2017; Ramakrishnan et al., 2021), which is the most commonly used platform for indoor datasets (Gu et al., 2022) cannot be directly used with other indoor simulators such as ThreeDWorld (Gan et al., 2020) and Ai2-thor (Kolve et al., 2017) because the environment layouts are different. As a result, evaluating embodied navigation methods across the

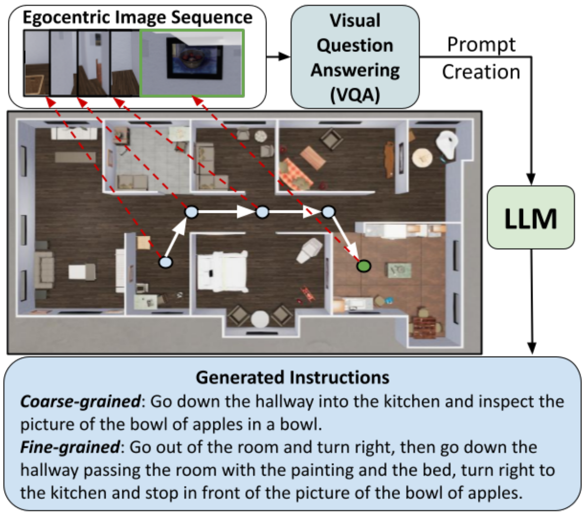


Figure 1: **Overview**: We use *in-context learning* with an LLM to generate multiple styles of *wayfinding instructions* for embodied navigation. Given **any** environment, we first gather a set of egocentric images along a path (white arrows), and obtain spatial knowledge via Visual Question Answering. We then condition an LLM on different styles of instructional language (coarse as well as fine grained) via reference texts. The figure highlights wayfinding instructions for this environment generated without training on any datasets.

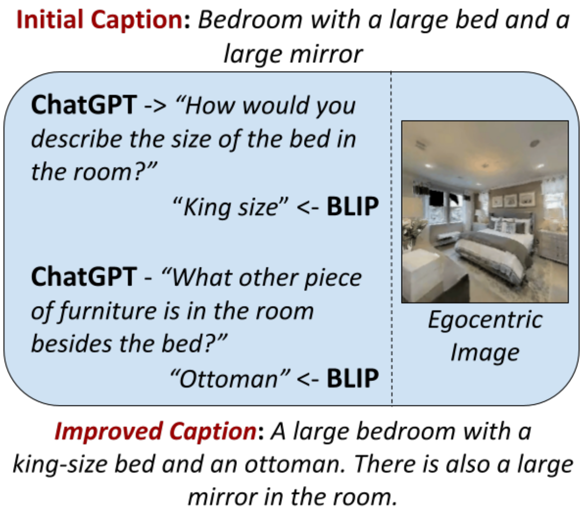


Figure 2: **Extracting Spatial Knowledge**: We use the GPT-3.5-turbo along with BLIP to maximize knowledge captured from an image, similar to ChatCaptioner (Zhu et al., 2023). We notice that adding more detail to the captions helps improve the quality the final instruction by filtering out unnecessary information. More details about this are in Appendix A.

simulators is rather difficult, which hinders experiments on their generalizability. It is important to design platform-agnostic wayfinding instruction synthesizers to help alleviate these issues.

Some recent works have looked at synthesizing instructions from input visual landmarks (Wang et al., 2022b; Kurita and Cho, 2020; Tan et al., 2019). These approaches however are not easily generalizable and require training a separate model for each instruction dataset to infer synthetic instructions. Moreover, they only focus on the Matterport3D environment, as indoor instruction datasets are scarce on other platforms.

Main Results: We present a novel approach to synthesize wayfinding instructions for an embodied robot agent. Figure 1 presents an overview of our approach. Given a set of egocentric images captured from a simulator, we perform Visual Question Answering to gather information about the scene, and use this to condition an LLM with reference texts to generate different styles of instructions. The novel components of our work include:

- We present a novel platform-agnostic, non-training based approach to synthesize wayfinding instructions of multiple styles.
- We use the *in-context learning* capabilities of LLMs to perform instruction synthesis in a few-shot manner. Our method only requires a few samples of reference wayfinding text to produce human-like instructions in multiple simulation platforms.
- We subjectively validate generated instructions across multiple simulation platforms via a user study and infer that 83.3% of users find the instructions accurately capture details of the environment, and exhibit human-like characteristics.
- Finally, we evaluate the effectiveness of our generated instructions on the REVERIE vision-and-language navigation (VLN) task. The performance of three zero-shot VLN approaches, evaluated using standard VLN success metrics, was comparable to established baselines, highlighting the efficacy and practical utility of LLM-generated instructions in navigation tasks.

In contrast to prior work which is limited to a single simulation platform and instruction style, we use in-context learning in LLMs to achieve *instruction synthesis* of multiple styles on different embodied simulation platforms, including Matterport3D, AI Habitat and ThreeDWorld. Our evaluation both via a user study and navigation performance indicates that the synthesized instructions are sufficiently representative of human-like texts for them to be used as a scalable alternative for generating instructions for embodied navigation tasks.

2 Approach

Our approach consists of two components. First, we perform Visual Question Answering (VQA) on egocentric images taken along an agent's path in a simulation environment. This gives us spatial knowledge about the scene. Next, we combine this spatial knowledge with a few reference *wayfinding instructions* in an in-context learning (Liu et al., 2023b) prompt to condition an LLM for synthesizing instructions that would lead the agent to the target location.

2.1 Extracting Spatial Knowledge: LLM + BLIP

Paths in simulated environments describe a navigable route for an embodied agent to get from one point to another. In our approach, given any embodied simulator, we first generate random paths. We then obtain a discrete set of egocentric images $\mathcal{I}$ uniformly sampled on this path.

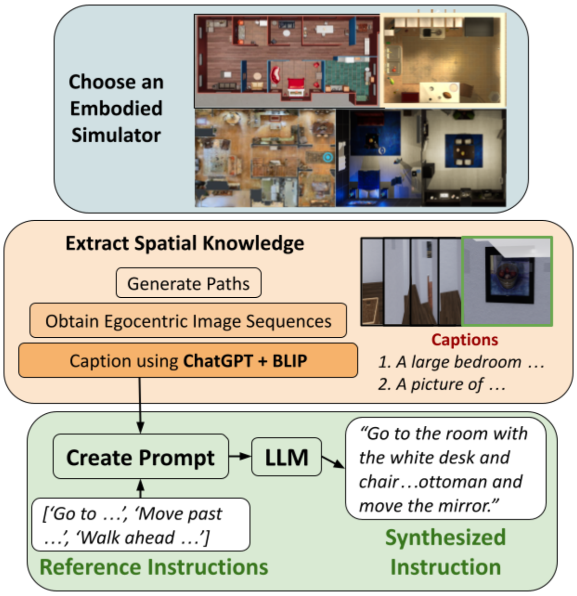


Figure 3: Given any embodied simulator, we synthesize multiple styles of wayfinding instructions for agents. Spatial knowledge is first mined from egocentric images $\mathcal{I}$ captured using the LLM and BLIP. These captions are fed into a prompt along with a few reference examples representing the desired instruction style. Finally, the LLM is conditioned with this prompt to generate a human-like instruction in the style of the reference text, using the captioned information.

We then perform VQA on the images in $\mathcal{I}$, to gather information about the environmental artifacts on the path. Following a similar approach presented in ChatCaptioner (Zhu et al., 2023), we maximize the knowledge obtained from each image by gathering insights via a conversation in a Chain of Thought manner (Wei et al., 2022) between GPT-3.5 (OpenAI, 2020) and BLIP (Li et al., 2023) (Figure 2). We notice that this gives us more detailed descriptions of each image, improving the quality of the generated instruction.

2.2 Synthesizing Wayfinding Instructions via In-Context Learning

We condition GPT-3.5-turbo-instruct to generate suitable wayfinding instructions for navigation. Figure 3 illustrates this approach. Captions obtained for images in $\mathcal{I}$ along with *reference texts* providing context on the desired instruction style are used to create a prompt for the LLM. We experiment with reference instructions taken from two datasets with contrasting styles; **R2R** (Anderson et al., 2018a), which has more detailed, *fine-grained* human annotations, and **REVERIE** (Qi et al., 2020a), which has instructions that are abstract and *coarse-grained*.

We also observe that adding more information about the instruction style itself helps further fine-tune the outcome. For instance, in the REVERIE dataset (Qi et al., 2020a), almost all instructions end by describing a task with the target object (*'turn the faucet'* for example). Adding this information as an additional constraint helps further finetune the LLM output. More details about this are provided in appendix A.

3 Evaluation & Results

In this section, we discuss our evaluation strategy and present results.

3.1 Qualitative: User Study

We conduct a user study to evaluate the quality of the generated instructions. Participants are first shown a video of a random path taken from one of 3 different simulators (Matterport3D, AI Habitat, ThreeDWorld). Using an instruction of either a REVERIE or R2R style as reference they are asked to come up with a stylistically similar instruction for the video. We then show them the generated instruction, and ask them a few questions about correlation. We infer that 83.3% of users believe that the generated instruction captured details of the environment to more than a decent level of accuracy, and that a majority of 73.3% believed that the agent could reach the target room by following the generated instruction. More details are in Appendix B.2.

3.2 Quantitative: Embodied Navigation

Our evaluation setup is simple. We first implement a zero-shot navigation scheme using the original instructions provided in REVERIE, a popular VLN dataset. We then replace the original instructions with instructions generated by our approach, and run the navigation scheme again. A similar performance would indicate that the generated instructions can indeed serve as a replacement to human-annotated data.

REVERIE is based on the Matterport3D simulator, which contains real-world captures of household environments. We look at 3 zero-shot VLN approaches - 1) **CLIP-Nav** (Dorbala et al., 2022), which uses CLIP (Radford et al., 2021) to ground target instructions to a scene to drive the agent's navigation policy, 2) **Seq-CLIP-Nav**, an extension of this approach that also performs backtracking (see Appendix B.3), and 3) **GLIP-Nav**, which we

Approach	Original			Generated (Central)			Generated (Panoramic)		
	SR ↑	OSR ↑	SPL ↑	SR ↑	OSR ↑	SPL ↑	SR ↑	OSR ↑	SPL ↑
Clip-Nav	6.57	28.68	0.06	5.98	26.69	0.05	5.57	26.09	0.05
Seq-CLIPNav	14.92	24.46	0.15	13.94	21.51	0.14	11.35	23.10	0.13
GLIP-Nav	16.87	32.56	0.18	16.32	33.23	0.18	14.18	29.87	0.15

Results: We evaluate zero-shot VLN models by replacing REVERIE's human-annotated instructions with instructions generated by our approach. Notice the similar performance on each VLN model across all metrics. There is a noticeable drop in using panoramic frames over central frames, and this could be attributed to condensing copious amounts of scene information into a single sentence (See Appendix B.3.2). We can positively infer from the minimal difference in SR, OSR, and SPL values that our approach can generate instructions that can indeed serve as a good replacement to human-annotated data.

introduce as a GLIP (Li* et al., 2022) based variant of Seq-CLIP-Nav. More details about these approaches are in Appendix B.3.

As Matterport3D provides panoramic images, we consider two possibilities for extracting spatial knowledge (see Appendix B.3.2); The **Central Caption**, where only the images in the direction of the agent's heading are captioned, and the **Panoramic Caption**, where the entire panorama (4 images) is captioned and summarized to obtain an instruction.

Experiment Details: We employ 3 standard VLN evaluation metrics (Zhao et al., 2021) to measure performance across each navigation approach - 1) **SR**, which is the **S**uccess **R**ate determining when the agent has successfully reached the target location; 2) **OSR**, the **O**racle **S**uccess **R**ate, for when the agent successfully reached the target location once, but overshot and stopped elsewhere, and 3) **SPL**, which measures efficiency of **S**uccess weighted by **P**ath **L**ength. The results table compares the performance of the generated instructions with the original ones on the zero-shot VLN approaches.

We make the following key **inferences** -

Automated Instruction Generation: A key observation is that embodied agents equipped with LLM-generated instructions perform almost equally well compared to when they are provided with human annotated instruction. This has practical implications for researchers working on embodied navigation, where such instruction data is limited and hard to annotate. Creating large-scale instruction datasets is challenging, often needing simulator-specific annotation tools, which cannot be easily transferred. To this end, our study presents a good alternative in leveraging off-the-shelf LLMs as a wayfinding instruction generation tool.

Cross-Platform Scalability: Our approach is platform-agnostic, and can be applied to generate instructions across embodied simulation platforms, whether they are discrete, continuous, photorealistic, or not. The user study validates this, where users across simulator types believed that the generated instructions captured details of the environment and could lead the agent to the target location. We believe that the embodied navigation community can significantly benefit from this, enabling researchers to conduct cross-platform generalizability experiments without relying on the availability of platform-specific human-annotated data.

Improved Instruction Quality: We notice that human-annotated instructions in REVERIE sometimes tend to be unnatural and lacking in terms of sentence construction. As these annotations are crowdsourced, this can be attributed to human error. It is often in these cases that the embodied agent fails to reach it's target location, due to poor annotation leading to inferior grounding scores. LLM-generated instructions on the other hand are almost always well structured, containing specific objects and waypoints leading up to a target location; a direct consequence of our prompting strategy. Some of these cases are discussed in appendix B.3.3.

4 Conclusion

We present a simple, cross-platform approach to synthesize multiple styles of wayfinding instructions for embodied navigation. Our approach operates under zero-shot setting, and instead utilizes an LLM with in-context learning to produce instructions across multiple simulation platforms. We verify the quality of the instructions generated both via a user study and by evaluating zero-shot VLN performance. We positively infer that the generated instructions are usable, and that our approach provides for a scalable and accessible solution for creating wayfinding instructions.

5 Limitations and Future Work

While our approach is platform-agnostic, the quality of the generated instructions is very sensitive to the individual modules that drive our scheme. Poor spatial knowledge extracted from performing VQA would directly affect the quality of the caption. In some preliminary experiments, we notice this behavior on some images taken from the VirtualHome (Puig et al., 2018) embodied simulator, which has non-photorealistic environments. Using LLaVA (Liu et al., 2023a) for VQA seems to create ghost objects and artifacts when asked to describe a scene leading to poor instructions. In contrast, it performs well with real world images taken from Matterport3D. We believe this poor performance might be because large captioning models such as LLaVA are trained on an abundance of real world data, and may contain fewer if not any simulation or non-photorealistic images. Secondly, during the synthesis stage, we present the LLM with examples from the instruction style that we wish to obtain. The generated instructions can sometimes contain the direct words or language used in these reference examples. As such, we believe it is necessary to explicitly specify in the prompt that the LLM uses only the captions and not the reference texts for generation. In the future, we intend to use our approach to implement a *generalist navigation agent* and study its performance in terms of *consistency* across various embodied simulation platforms.

6 Ethics Statement

Equipping embodied agent with LLM-generated instructions to perform navigational tasks is a step towards cohesive human-robot collaboration. While the end goal is to make such systems fault-tolerant and error-free, we may not want an agent to perform certain actions that it is unsure of. However, currently there seems to be a gap in the language interpretation capabilities of the agent especially in complex scenarios.

Our user study protocol was approved by Institutional Review Board and we do not collect, share or store any personal information of the participants.

References

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

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Vishnu Sashank Dorbala, James F Mullen Jr, and Dinesh Manocha. 2023. Can an embodied agent find your" cat-shaped mug"? llm-based zero-shot object navigation. *arXiv preprint arXiv:2303.03480*.

Vishnu Sashank Dorbala, Gunnar Sigurdsson, Robinson Piramuthu, Jesse Thomason, and Gaurav S Sukhatme. 2022. Clip-nav: Using clip for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2211.16649*.

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. 2020. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*.

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)*. Association for Computational Linguistics.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. 2022a. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2022b. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. 2023. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10813–10823.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 104–120. Springer.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*.

Shuhei Kurita and Kyunghyun Cho. 2020. Generative language-grounded policy in vision-and-language navigation with bayes' rule. *arXiv preprint arXiv:2009.07783*.

Jialu Li, Hao Tan, and Mohit Bansal. 2022. Envedit: Environment editing for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15407–15417.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. 2022. Grounded language-image pre-training. In *CVPR*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2023. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *arXiv preprint arXiv:2305.15021*.

OpenAI. 2020. Language models are unsupervised multitask learners. *OpenAI Blog*, 23(6). https://openai.com/blog/chatgpt.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020a. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020b. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.

Yanyuan Qiao, Yuankai Qi, Zheng Yu, Jing Liu, and Qi Wu. 2023. March in chat: Interactive prompting for remote embodied referring expression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15758–15767.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. 2021. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Dhruv Shah, Błażej Osiński, Sergey Levine, et al. 2023. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning*, pages 492–504. PMLR.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*.

Hanqing Wang, Wei Liang, Jianbing Shen, Luc Van Gool, and Wenguan Wang. 2022a. Counterfactual cycle-consistent learning for instruction following and generation in vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15471–15481.

Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. 2022b. Less is more: Generating grounded navigation instructions from landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15428–15438.

Xiaohan Wang, Wenguan Wang, Jiayi Shao, and Yi Yang. 2023. Lana: A language-capable navigator for instruction following and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19048–19058.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. 2023. L3mvn: Leveraging large language models for visual target navigation. *arXiv preprint arXiv:2304.05501*.

Ming Zhao, Peter Anderson, Vihan Jain, Su Wang, Alexander Ku, Jason Baldridge, and Eugene Ie. 2021. On the evaluation of vision-and-language navigation instructions. *arXiv preprint arXiv:2101.10504*.

Gengze Zhou, Yicong Hong, and Qi Wu. 2023a. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*.

Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. 2023b. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. *arXiv preprint arXiv:2301.13166*.

Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. 2023. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv preprint arXiv:2303.06594*.

A In-Context Learning Strategies

In this section, we discuss some strategies we employ to get the best possible wayfinding instruction. A prompt template is presented to the LLM as -

> *"A robot agent at home sees a sequence of egocentric images with the following frame descriptions.*
> *Frame 0: <Caption 1>*
> *Frame 1: <Caption 2>*
> *...*
> *Frame n: <Caption n>*
> *Reference Texts: ['Go to ...', 'Move past ...', Walk ahead ...]*
> *Write an concise instruction in the style of the Reference Texts that would get the robot from Frame 0 to Frame n.*
> *<Additional constraints on the instruction style>"*

Each caption in this template is obtained using the LLM + BLIP strategy outlined in section 2.1.

A.1 Influence of LLM + BLIP

An example egocentric image sequence of a path taken in the TDW simulator (Gan et al., 2020) is shown in figure 4. Using the LLM + BLIP approach discussed in section 2.1, we get the following captions for each image in 4.

1. *The image depicts a computer screen showing a colorful video of a man that is being displayed on a television. There is also a chair visible in the image besides the television.*

2. *The image contains a small chair made of fabric, in colors of red, white and gray. There is another object present in the image, but it is not clear what it is.*

3. *The image is of a living room with brown furniture and no decorations on the walls. There are no people present in the living room.*

4. *The image depicts a room with a gray couch located against a wall. There is a small television mounted on the wall.*

5. *The image features a computer screen displaying a website, with a couch visible in the background. A plant is placed on a table next to the computer. No other objects are visible on the table.*

Passing these Frame descriptions to the prompt in A gives us a REVERIE-like instruction as,

> *Go to the living room, then move to the room with the gray couch and turn off the television mounted on the wall.*

and R2R-like instruction as,

> *Go from the computer screen to the chair, then past the object in the background and into the living room. Walk past the blue furniture and turn right towards the gray couch. Finally, stop in front of the table with the plant and view the website on the computer screen.*

Notice the level of detail added by *insights* that LLM gathers by conversing with BLIP. This inturn gives more information for GPT-3.5-Turbo-Instruct to use for generating a final instruction from the frames.

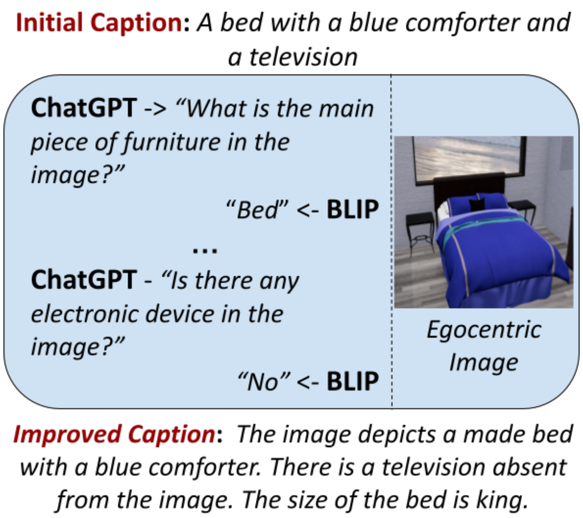


Figure 7: **LLM + BLIP**: Notice the initial BLIP caption mentions a television in the image, even when it is not present. When the LLM asks for the presence of electronic items in the room, BLIP answers no, which leads to the refined caption preventing misinformation.

We also experiment with using only BLIP captions with object and room queries, without the LLM. The REVERIE-like instruction in this case looks like,

> *Go to the living room on level 0 and turn off the television by the couch and the table.*

and the R2R-like instruction is,

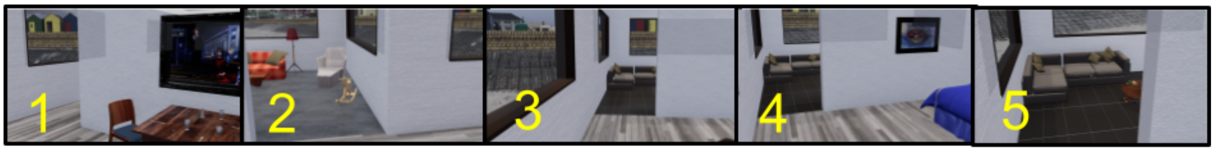


Figure 4: Egocentric Image Sequence from a path in ThreeDWorld (Gan et al., 2020)

Figure 5: Egocentric Image Sequence from a path in AI Habitat (Ramakrishnan et al., 2021)

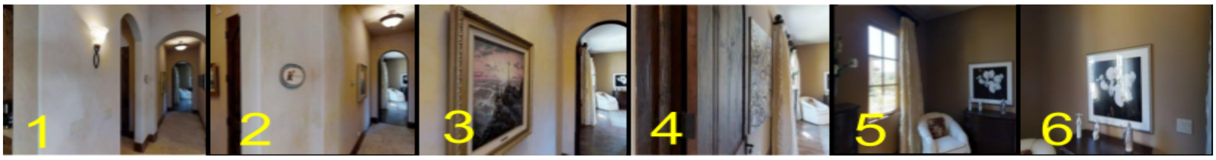


Figure 6: Egocentric Image Sequence from a path in Matterport3D (Chang et al., 2017)

> *Start in the kitchen and go up the stairs on the left. Turn right at the top of the stairs and then go past the round table and chairs and stairs. Keep walking until you see the two small tables on the rug and then turn left. Go down the hallway keeping the wall on your left and stop in front of the door on your right with the treadmill. Turn left and you will see the living room with a computer screen containing a picture of a couch and a table.*

While the REVERIE-like instruction is still usable, notice the R2R-like instruction tends to be nonsensical with ghost objects such as *stairs* and *treadmill* in the caption. It also contains bad directions. We observe this phenomenon in multiple cases, and Figure 7 showcases how the conversation with the LLM improves the initial captions to remove ghost objects and prevent misinformation.

Thus, we infer that using an LLM with BLIP to provide more detail about the environment is important when it comes to finally generating more meaningful instructions.

A.2 Empirical Information on Instruction Styles

We utilize factual knowledge about R2R and REVERIE instruction styles to finetune the LLM prompt.

A.2.1 Additional Constraints for R2R

Upon inspection, we observe that R2R instructions are usually 2 or more sentences long, attributed to longer path lengths. Further, in the R2R paper, the authors mention that they ask annotators to “write directions so that a smart robot can find the goal location after starting from the same start location", and are told that it is not necessary to follow the path, but only to reach the goal. We incorporate this information to append our prompt:-

> “Write directions so a smart robot can find the final frame after starting from the same starting frame. You do not have to use information in the frames, and just need to reach the goal location."

A.2.2 Additional Constraints for REVERIE

REVERIE instructions are concise, and talk only about the goal location. Clip-Nav (Dorbala et al., 2022) studies REVERIE in detail and empirically deduces that most instructions can be broken down into *navigation* and *activity* components, with the conjunction *and* between them. We utilize this information to add the following to our prompt:-

> "The instruction must be a single sentence long, ending with a task related to an object in the final frame, and must be less than 20 words."

B Evaluation Details

B.1 Simulator Implementations

We implement our approach on 3 different simulation platforms, namely AI Habitat (Ramakrishnan et al., 2021), Matterport3D (Chang et al., 2017) and ThreeDWorld (TDW) (Gan et al., 2020). Egocentric image sequences for these simulators are presented in Figure 4, Figure 5 and Figure 6 respectively. Depending on the type of simulator, we revise our strategy for extracting sequences as listed below -

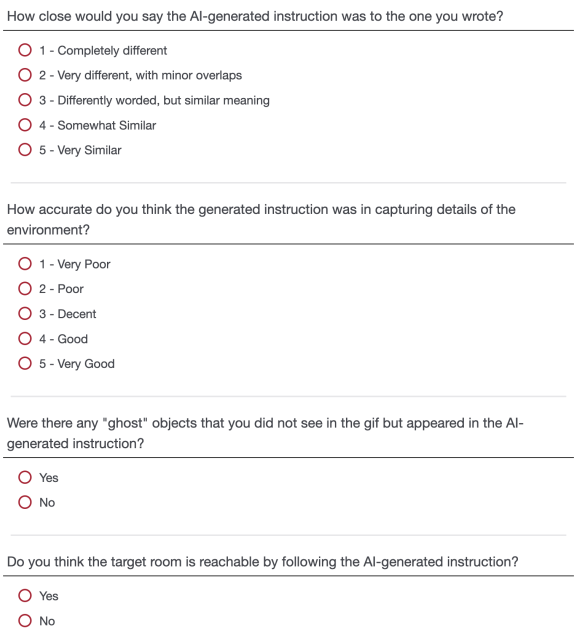
- Environments in the **Matterport3D** simulator are taken from real world scenes and provide fully connected graphs whose nodes represent 360 panoramas. Given two nodes from the connected graph, we compute a path between them as a sequence of nodes. To compute captions, we either consider the central frame or the entire panorama (described in Appendix B.3.2). The path contains discrete "hops" of in the form of images, which gives us our image sequence.
- **AI Habitat** has continuous 3D reconstructions of real world household environments. To obtain a path, we first sample two navigable points in the environment and compute the shortest distance between them. Then, to obtain a discrete sequence of images, we sample images at a uniform interval along the path.
- **TDW** is a photorealistic simulator that is capable of procedurally generating new environments. We make use of this simulator to test the robustness of our approach in non-real world environments. We obtain our image sequence in the same manner as AI Habitat.

For the user study, we sample 100 paths of varying lengths from each of these simulators, randomly choosing from environments they offer. We then use our approach on these paths to generate instructions in a platform-agnostic manner.

B.2 Qualitative Analysis - User Study Details

Each user is presented with a random image sequence chosen from a bank of sequences gathered from the 3 different environments. This allows for us to evaluate the generated instruction across multiple platforms. We observe a consistent performance across simulators, leading us to establish the platform-agnostic nature of our instruction synthesizer.

Our study was aimed at quantifying the usability of generated instructions in guiding an embodied agent in the environment. In this direction, we first presented the user with video of an egocentric image sequence chosen from a random simulation platform. After being shown examples of fine or coarse grained instructions, the users were asked to provide an instruction describing the robot's path in that style. Finally, the participant is shown the synthesized instruction for the same sequence and is asked comparative questions highlighted in figure below.


How close would you say the AI-generated instruction was to the one you wrote?

- 1 - Completely different
- 2 - Very different, with minor overlaps
- 3 - Differently worded, but similar meaning
- 4 - Somewhat Similar
- 5 - Very Similar

How accurate do you think the generated instruction was in capturing details of the environment?

- 1 - Very Poor
- 2 - Poor
- 3 - Decent
- 4 - Good
- 5 - Very Good

Were there any "ghost" objects that you did not see in the gif but appeared in the AI-generated instruction?

- Yes
- No

Do you think the target room is reachable by following the AI-generated instruction?

- Yes
- No

Our User Study. The participant is asked questions on the quality of the generated instructions and about how much it compares with the instruction that they wrote.

Each question aims to tackle a different comparative perspective. The first question seeks to find out if the generated instructions are similar to what the user has written down. The second question asks if the generated instructions accurately capture details of the environment. The third queries about the robustness of generation by asking if the participant has noticed any ghost objects or artifacts. Finally, we ask if the user thinks an embodied agent could reach the target location by following the generated instruction.

Out of a total of 30 participants, 83.3% believed the instruction captured details of the environment to a more than decent level of accuracy. A majority (73.3%) of these users also believed that the agent could reach the target room by following the generated instruction. A lower percentage of participants (16.5%) reported seeing ghost objects, which indicates either that some people may have missed objects in the video, or that the generated instruction is sensitive to the captioning scheme.

Conversely, 43.3% of participants believed that the instructions generated were either very different from what they wrote, or had minor overlaps. We can infer from this that the vocabulary people use to describe a path may significantly vary from the generated instruction. However, this does not necessarily mean that the agent would not be able to follow the generated instruction to reach the target location, as it would use alternate references or landmarks to get there.

Our study was determined exempt by our institution's IRB. All of the participants voluntarily chose to participate in it.

B.3 Quantitative Study - Zero-Shot Embodied Navigation

B.3.1 Dataset and Navigation Setup Details

We run navigation experiments on the REVERIE dataset, which tackles vision-and-language navigation (VLN) using coarse-grained instructions. Instructions in REVERIE have been human-annotated, where the annotator is asked to write a high-level instruction describing how to get to the target location after being shown a path in the Matterport3D environment. Each path is discrete, i.e., it consists of a set of panoramic images or nodes along which the agent "*hops*". The nodes inturn consist of 4 views covering a 360 degree view of the agent.

We consider a generalizable, zero-shot case, where the agent is dropped in an environment that it has no knowledge of, and is given an instruction that it must follow to get to a target location. This setting is in line with our ultimate goal of developing a generalist embodied navigation agent, which is able to function without any supervision in an unseen environment. We opt to use the unseen validation split of the REVERIE dataset for evaluation, which contains environments that the agent would not see in the training split. It contains 504 paths, which was deemed sufficient for showcasing zero-shot navigation prowess using the generated instructions.

CLIP-Nav (Dorbala et al., 2022) uses CLIP to make grounding decisions for navigation. The instruction is first broken down into a Navigation Component (NC) and an Activity Component (AC). The NC contains information about getting to the target location, while the AC containing the activity that the agent is expected to perform is disregarded. The NC is further broken down into noun phrases using GPT-3.5-turbo, which are then grounded using CLIP with each of the 4 images captured by the agent from its panoramic view. The agent takes the direction of the highest CLIP grounding score.
Seq-CLIP-Nav extends this to incorporate backtracking. Backtracking refers to when the agent falls back or "backtracks" a few nodes when it determines that it has taken the wrong path.

We also ablate with **GLIP-Nav**, a variant of Seq-CLIP-Nav we introduce, where CLIP is replaced with GLIP (Li* et al., 2022) for obtaining grounding scores.

B.3.2 Matterport3D: Frame Selection

REVERIE provides a set of panoramic images taken from Matterport3D that forms a path corresponding to each instruction. The annotator is provided with this whole panoramic view at each step. To incorporate our generation approach here, we consider two variations.

Central Caption: We hypothesize that the central frame contains the most immediate and critical information required for the embodied agent to perform its next set of actions. To this end, we caption only the central frames (i.e., the image in the direction of the agent's heading) of the entire path sequence to generate the instruction.

Panoramic Caption: Here we caption each image of the entire panorama (4 frames), and summarize the individual captions using the LLM. We perform this over the entire path sequence to generate the instruction. Although the panoramic sequence contains more semantic information over the single (central) frame, note that each instruction is only a single sentence, and compressing all the information of a scene (be it the target or an image along the path) is non-trivial, if the instruction has to be of a suitable length.

During the panoramic-frame case, we use the LLM to summarize the set of captions obtained 4 90 degree views around the agent. Each caption in this set is obtained using the LLM + BLIP approach discussed in section 2.1. The prompt for this is -

> *"I see a panoramic view with the following descriptions.*
> *North: <Caption 1>*

East: <Caption 2>
South: <Caption 3>
West: <Caption 4>
Summarize these descriptions into a single description using less than 20 *words."*

B.3.3 Inferences on Generated Instructions

In addition to the results presented in section 3.2, we also measure the *average pairwise cosine similarity* using MiniLM-V6 (Reimers and Gurevych, 2019) between the human-annotated instructions and the generated instructions.

For the central-caption case, we get a score of 0.476, and for the panoramic-caption case, we get 0.433, on a scale of -1 to 1. From the overall positive correlation, we can infer that the generated instructions tend to be similar to the human-annotated ones on average. Some individual cases of extreme difference are discussed below.

In a low cosine similarity example, consider

> **Human-Annotated**: *"Walk to the bottom of the stairs leading to the level 1 hallway and find the bottommost stair"*
> **Generated**: *"Move from bedroom to kitchen, turn off faucet."*
> **Similarity**: 0.0850

Notice that the human-annotated instruction presents a unique situation to the agent where it is expected to find the *bottommost stair*. In contrast, the generated instruction asks the agent to move to the kitchen, which is near the vicinity of the staircase in this environment. While the cosine similarity might be low, a generalist agent would still be able to reach the target location with the given instruction since it references other elements (“the faucet" here) in the scene. Note that VLN tasks deal with the agent reaching a target location, and not with what it needs to do once it gets there.

In a high cosine-similarity example, consider,

> **Human-Annotated**: *"Go through the nearest bedroom to the bathroom on the first floor and turn on the faucet on the rightmost"*
> **Generated**: *"Go to the bedroom and turn off faucet."*
> **Similarity**: 0.820

Observe that a high cosine similarity does not necessarily mean that the generated instruction is of good quality. In this example, notice that the human annotator asks the agent to enter the bathroom after going through the bedroom to turn off the faucet. The generated instruction however entirely misses out on entering the bathroom, which would cause an agent to incorrectly look for a faucet in the bedroom.

These are however one-off cases; we observe that most generated instructions tend to closely follow or paraphrase human-annotations. For instance, consider,

> **Human-Annotated**: *"Go to the bathroom on level 1 and wipe off the faucet"*
> **Generated**: *"Go to the wooden room on level 1, turn off faucet in the bathroom."*
> **Similarity**: 0.885

Both these instructions ask the agent to go to the bathroom on level 1 to execute a task.

C Related Work

C.1 Embodied Instruction Synthesis

Embodied or Vision-and-Language Navigation deals with the problem of navigating an agent in unseen photorealistic environments and adhering to language instructions. These wayfinding instructions are usually human annotated as part of datasets (Ku et al., 2020; Qi et al., 2020b; Anderson et al., 2018b; Krantz et al., 2020), and can roughly be categorized into coarse and fine-grained (Gu et al., 2022) based on their level of detail. As these datasets are exclusive to the environments that they are created in, generalizing them to other new or procedurally generated environments presents a unique challenge. Most prior work on instructions synthesis (Li et al., 2022) has mostly been tailored toward data augmentation. (Wang et al., 2022a) presents a counterfactual reasoning approach to generate instructions, but ultimately requires the model to be trained on the R2R (Anderson et al., 2018a) dataset. (Wang et al., 2022b; Kamath et al., 2023) present imitation learning models that are trained on datasets, and use the augmented instructions to improve navigation performance. More recently Wang et al. (2023) presents a navigation agent which is able to not only execute human-written navigation commands, but also provide route descriptions to humans. These approaches are limited to a few datasets and have cumbersome training procedures. In contrast, our approach can generalize over multiple styles of instructions, over multiple simulation platforms without requiring a dataset.

C.2 LLMs for Embodied Robot Navigation

Recent work has used LLMs being for embodied robot navigation (Huang et al., 2022a; Zhou et al., 2023a), especially in a zero-shot setting (Yu et al., 2023; Dorbala et al., 2022). While (Shah et al., 2023) leverage GPT-3.5 (Brown et al., 2020) to identify landmarks, (Zhou et al., 2023b) and (Dorbala et al., 2023) use an LLM for commonsense reasoning between objects and targets to facilitate navigation. With LLMs being increasingly used in several embodied AI frameworks beyond navigation (Mu et al., 2023; Huang et al., 2022b), utilizing them for instruction generation allows for easier integration and testing at a system level. Finally, March-in-Chat (MiC) (Qiao et al., 2023) can talk to the LLM on the fly and plan the navigation trajectory dynamically.