TRAVEL: Training-Free Retrieval and Alignment for Vision-and-Language Navigation

Navid Rajabi
George Mason University
nrajabi@gmu.edu

Jana Košecká
George Mason University
kosecka@gmu.edu

Abstract

In this work, we propose a modular approach for the Vision-Language Navigation (VLN) task by decomposing the problem into four sub-modules that use state-of-the-art Large Language Models (LLMs) and Vision-Language Models (VLMs) in a zero-shot setting. Given navigation instruction in natural language, we first prompt LLM to extract the landmarks and the order in which they are visited. Assuming the known model of the environment, we retrieve the top-k locations of the last landmark and generate k path hypotheses from the starting location to the last landmark using the shortest path algorithm on the topological map of the environment. Each path hypothesis is represented by a sequence of panoramas. We then use dynamic programming to compute the alignment score between the sequence of panoramas and the sequence of landmark names, which match scores obtained from VLM. Finally, we compute the nDTW metric between the hypothesis that yields the highest alignment score to evaluate the path fidelity. We demonstrate superior performance compared to other approaches that use joint semantic maps like VLMaps [11] on the complex R2R-Habitat [1] instruction dataset and quantify in detail the effect of visual grounding on navigation performance.

1. Introduction

Vision-and-Language Navigation (VLN) task involves controlling an agent, either in simulation or in the physical world, to navigate through an environment by following natural language instructions. Consider an example in Fig. 1 where the agent is required to follow the instructions in a specific environment. This task requires parsing the language input (e.g., "Turn left in the hallway, go to the kitchen, and stop by the sink"), grounding the phrases to visual concepts such as scenes, landmarks, and actions (e.g., turn left) as well temporal cues (e.g., turn before).

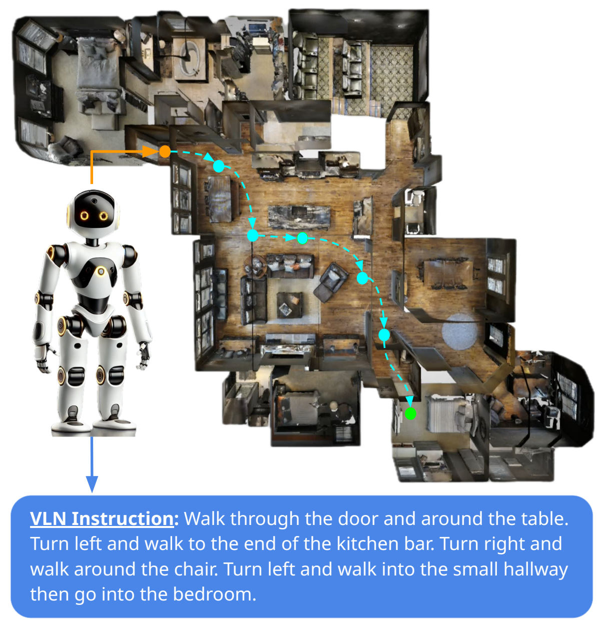


Figure 1. Bird's Eye View visualization of a sample VLN episode from R2R dataset [1].

One class of approaches formulates the Vision-Language Navigation task as a supervised multi-modal sequence-to-sequence learning task, where the learner is given episodes of natural language instructions, along with visual observations and navigation actions. These approaches were supported by large-scale datasets of navigation instructions, e.g., Room-2-Room (R2R) [1], in Matterport3D [4] indoor environments, providing the agent with panoramic images from different locations. The sequence-to-sequence methods varied in their multi-modal language and vision architectures, training techniques, and choices of representations, gradually improving the benchmark performance. Despite these improvements, non-negligible gaps still exist between machines' and human performance on existing benchmarks, the performance suffers in novel environments and in the presence of more complex variation of instructions.

Another class of methods pursued a more modular approach, using or learning separate modules for processing natural language inputs, using semantic segmentation or detection to ground noun phrases images and in the map and integrating these with more traditional map-based navigation. These methods, however, use simple natural language instructions and are typically evaluated on small-scale datasets [11, 18].

Contributions. In the presented work, we pursue a modular approach, where we exploit zero-shot capabilities of the state-of-the-art LLMs for understanding and parsing navigation instructions and VLMs for grounding landmark names in the visual observations. The navigation component is carried out by finding a path in the topological map of the environment that is best aligned with the navigation instructions. The map is acquired using the training episodes from R2R dataset [1], and the alignment score is computed using dynamic programming, where the costs of individual steps are obtained from the state-of-the-art Vision-Language Model. The presented modular approach demonstrates superior performance over occupancy map-based approaches and reveals current strengths and weaknesses of the state-of-the-art LLMs and VLMs for vision-language instruction following.

2. Related Work

For the purpose of our exposition, the existing works on Vision Language Navigation can be partitioned into end-to-end and modular approaches. The end-to-end methods take the natural language instructions, visual observations, and actions and train a multi-modal sequence-to-sequence model, and in the inference stage, given the instruction and initial view, the model generates the sequence of actions while ingesting additional views. The modular approaches integrate LLMs, VLMs, or both with more traditional map-based representations along with a common robotics navigation stack comprised of basic navigation skills that are not learned.

End-to-end approaches. These methods typically adopt a sequence-to-sequence model, taking as an input the language instruction and visual information and outputs the sequence of low-level navigation actions (move, turn left/right) or local waypoints. During the forward pass, the entire instruction is processed by the Language Encoder (e.g., LSTM/transformer). The aggregation of the context vectors, plus the encoded current view of the agent, is then fed to the Action Decoder (e.g., LSTM/transformer) that generates the next `action`. The decoder continues to predict actions until it generates the `STOP` action. The mixture of Reinforcement Learning (RL) and Imitation Learning (IL) has been commonly used for training these models [24]. The existing approaches proposed different variations of model architectures, training strategies and choice of representations [2, 6, 8–10, 15, 19, 24, 26] typically using the Room-to-Room (R2R) [1] and Room-Across-Room (RxR) [16] benchmarks for training and evaluation. The natural language instructions in these benchmarks are quite complex, with an average length of ~ 26 words. These approaches have made substantial improvements in past years, mostly thanks to increasing the number of training episodes and auxiliary tasks that support grounding [25] and instruction generation [8, 14]. It has been shown [29] that the performance of the existing methods continues to be severely compromised by the inability to ground landmarks, understand spatial relationships, as well as grounding of action phrases. The ability to ground landmarks is more critical for indoor environments, while in outdoor settings, the grounding of actions in navigation instructions is more critical. Furthermore, RL & IL require a large number of high-quality training episodes, in addition to the extra computational complexity of RL due to the online interaction of the agent with the simulator/environment that makes it more difficult to scale the training [14].

LLM and VLM based modular approaches. Language Models were used in the past as zero-shot planners, where [12] introduced the idea of utilizing the knowledge learned by LLMs, like OpenAI GPT-3 [3] and Codex [5], for decomposing high-level tasks (e.g. ”make breakfast”) to sequences of lower level skills executable by the agent. For navigation tasks, CLIP-Nav [7] utilized CLIP VLMs [20] for grounding instruction phrases and GPT-3 [3] for decomposition of complex natural language instructions into phrases. In CLIP-Nav, the language instruction is decomposed using GPT-3 [3], and then each sub-instruction, along with a panorama comprised of four egocentric views, is ranked by CLIP [20] to determine the closest heading direction. The major limitations of CLIP-Nav are the dependency on the existence of a navigable graph of the environment and the poor ability of CLIP to associate landmarks with images. Another decomposition of the navigation task was adopted by the VLMaps [11] approach, which first builds a global joint vision-language semantic occupancy map by exploring the environment. The cells of the map are populated by LSeg/CLIP embeddings [17, 20], projected onto the grid from images. The navigation instructions are simpler, often resorting to point and object goal navigation, which are further translated into robotic navigation skills in the form of executable code.

Lang2LTL [18] represents another line of work that has been proposed to use LLMs to translate free-form

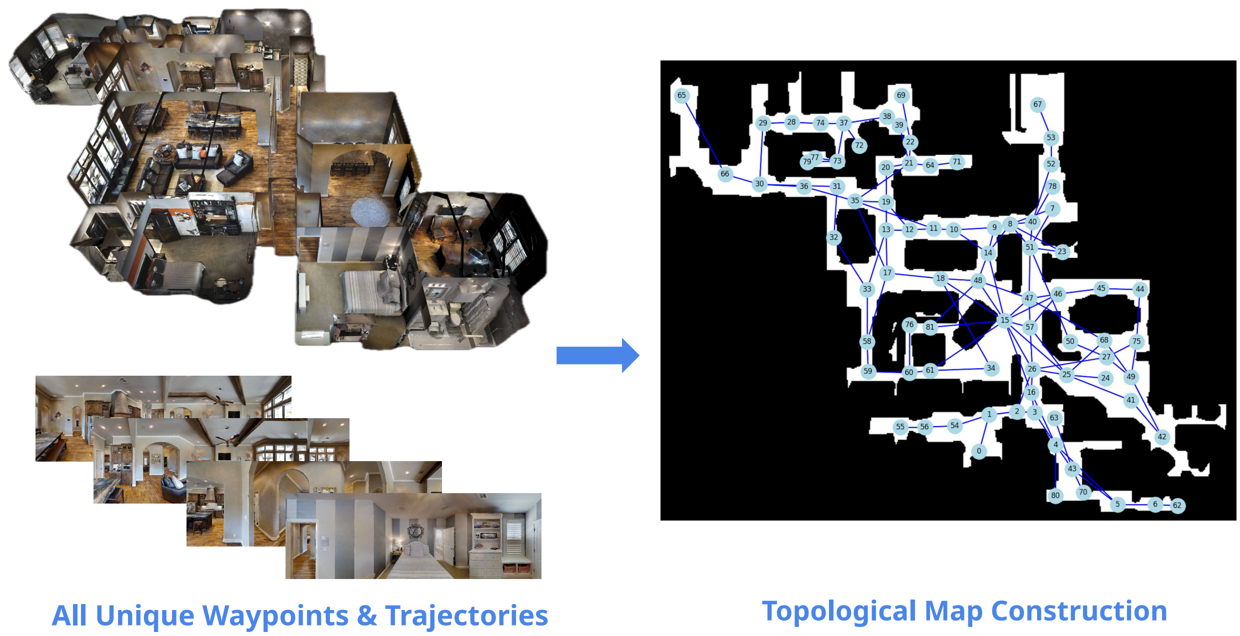


Figure 2. Topological Map Construction

natural language instructions into linear temporal logic (LTL). Lang2LTL is advantageous because it disambiguates the goal specification and facilitates incorporating temporal constraints. The limitations of Lang2LTL are the need for a parallel dataset of natural language instructions and their corresponding fixed set of LTL formulas for fine-tuning the LLMs for the translation stage and the limited level of complexity of the instructions, compared to R2R [1] and RxR [16] benchmarks. Authors in LM-Nav [23] propose a zero-shot approach for outdoor instruction following. They utilize a visual navigation system called ViNG [22], to construct a topological map G from a set of observations, followed by extraction of landmarks L from the free-form navigation instruction using GPT-3. CLIP is then used to infer a joint probability distribution over the nodes in G and landmarks in L, followed by a graph search algorithm to find the optimal path that is executed by local navigation policy. The approach in LM-Nav can only navigate to a sequence of unique landmarks by design, discarding complexities like spatial clauses and fine-grained grounding of landmark and action phrases.

3. Our Approach

We introduce a modular approach for solving the VLN task using the pre-trained state-of-the-art language and vision and language models in a zero-shot setting, focusing on complex instructions in the R2R-Habitat dataset [1]. The agent first builds a topological map of the environment using the train split episodes of the dataset. We used all the available unique waypoints and trajectories of the environment to build the graph G, where each node v is represented by a 360° RGB panorama and each edge e has a weight of 1, representing the connectivity between each pair of nodes, as shown in Figure 2. In this way, we ensure consistency in our evaluation process as every node of the ground-truth waypoints from the training episodes has a corresponding node in the topological map.

Then, we extract the sequence of landmarks from the natural language instruction using a pre-trained LLM, `LLama-3.1-8B-Instruct` in our case. We identify the last landmark phrase and search panoramas for the top-k most likely goal nodes. Suppose that the last landmark is *bedroom*, we can locate the goals by recognizing whether the *bedroom* can be found in the panoramic images associated with the graph nodes. In this way, we will narrow down the set of possible paths that lead to the goal locations.

We use the state-of-the-art vision language model SigLIP [28] for goal/final landmark recognition, as shown in Figure 3. SigLIP training is similar to the CLIP model, replacing the contrastive loss with sigmoid binary prediction. The recognition is carried out by computing cosine similarity between panorama images and the textual description of the landmark. In order to compare the effectiveness of this choice with an open-vocabulary semantic map such as VLMaps [11] that endows the occupancy map with CLIP embeddings, we ran the landmark localization experiment on all 127 landmarks and reported the mean Precision@10 in Table 1. The superiority of our approach stems from recognizing the landmarks in the panoramic views and replacing CLIP [20] with SigLIP [28], instead of using open-vocabulary semantic occupancy maps.

Model	# Landmarks	Precision@10 (%)
VLMaps [11] w/ CLIP [20]	127	34.4
Ours w/ SigLIP [28]	127	**70.0**

Table 1. SigLIP vs. VLMaps Quantitative Results for Last Landmark Indexing

Given the top-k goal locations, we compute the BFS shortest path from the starting pose to the goal nodes, obtaining k paths hypotheses. In the next stage, we quantify the alignment of the instruction with each of the paths and select the one with the highest alignment score. We introduce two approaches for path-instruction alignment and ranking. In Approach I, we formulate this problem as a sequence-to-sequence alignment, where the sequence of panoramas is $X = [X_0, X_1, ..., X_p]$, and the sequence of landmark phrases $Y = [Y_0, Y_1, ..., Y_l]$, as shown in Figure 5. Considering $X \times Y$ as a matrix A, where A_{ij} is the binary grounding scores of landmark being present in the panorama associated with the waypoint. We use the state-of-the-art VLM, GPT-4o, in our case, as shown in Figure 4 to obtain these scores. We first discard the path hypotheses where the number of nodes is smaller than the number of landmarks. Then, given the A matrix we compute for each path the normalized alignment score using Dynamic Programming (DP) formulation similar to the Longest Common Subsequence (LCS) problem, named Pano2Land described in Algorithm 14.

Figure 5 shows the alignment matrix A for three path hypotheses, comprised of 8, 7, and 6 nodes. The left example yields a score of 5/8, corresponding to 5 of the landmark names being successfully grounded in the right order in 8 consecutive panoramas. The middle example yields a score of 5 by grounding landmarks in panoramas 2, 3, 4, 5, and 7, where the final score would be $5/7 = 0.71$. The right example demonstrates the perfect way of aligning all the panoramas to the corresponding landmarks without skipping, yielding the top score of $6/6 = 1$.

Alternatively, we introduce Approach II for path ranking by prompting GPT-4o to rate each path on a scale of 1 to 5 given the sequence of panoramas in order, original natural language instruction, and the extracted sequence of landmark phrases, as shown in Figure 6. This approach bypasses the individual landmark grounding stage and alignment score computation done by Pano2Land algorithm. The performance of this approach is slightly worse than Approach I. Furthermore, the results are less interpretable since the internal ranking mechanism of GPT-4o is unknown.

Finally, for each approach's output, we compute the normalized dynamic-time warping (nDTW) metric between the ground truth and the best-aligned path to measure the path fidelity; nDTW is more aligned with our task objective compared to the Success Rate (SR), which only considers an episode to be successful if the agent's last position is within 3 meters of the ground-truth goal and it does not explicitly consider the intermediate alignments with the landmarks that were supposed to be visited in order by the agent [13].

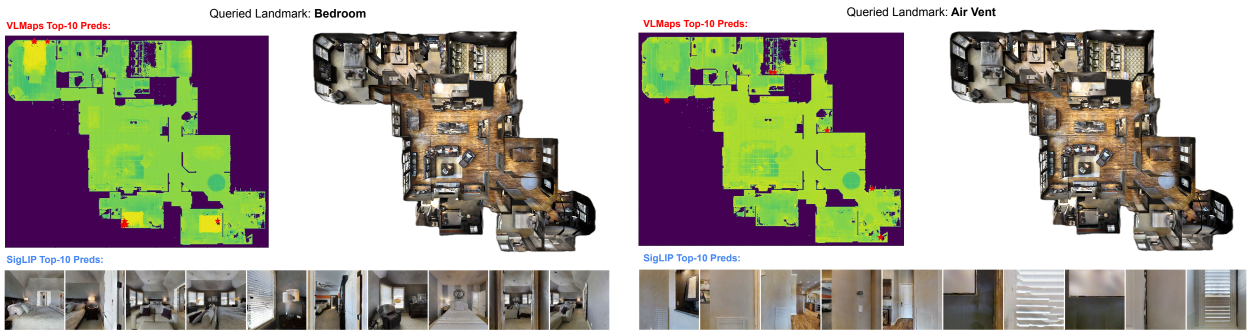
Algorithm 1 - Pano2Land algorithm for calculating path alignment using grounding scores, similar to DP/LCS.

Require: Binary grounding matrix $\mathbf{M} \in \{0,1\}^{R\times C}$
Ensure: Alignment score S

$R \leftarrow$ number of rows (landmarks) in $\mathbf{M}$
$C \leftarrow$ number of columns (panoramas) in $\mathbf{M}$
Initialize matrix $\mathbf{dp} \in N_0^{(R+1)\times(C+1)}$ with zeros
for $r \leftarrow 1$ to $R+1$ **do**
  for $c \leftarrow 1$ to $C+1$ **do**
    if $M_{r,c} == 1$ **then**
      $\mathbf{dp}[r][c] \leftarrow (\mathbf{dp}[r-1][c-1]) + 1$
    else
      $\mathbf{dp}[r][c] \leftarrow \max(\mathbf{dp}[r-1][c],\ \mathbf{dp}[r][c-1])$
    end if
  end for
end for
$S \leftarrow \mathbf{dp}[R][C]$
return S

In Table 2, Hypo Path Gen accuracy indicates the fraction of episodes where the ground-truth path or a highly similar one is among the selected path hypotheses. There might be multiple reasons why the correct path couldn't be retrieved, including but not limited to (1) not being able to ground the last landmark, (2) encountering a dramatically different landmark that has been part of the train samples, (3) a highly-frequent last landmark which exists in multiple locations (e.g., door) where the ground-truth landmark location may not fall into the top-3 retrieved ones, etc.

The Approach I nDTW shows the average nDTW score for all 21 episodes in each environment. If the nDTW is above 87%, we consider the path successful, where this threshold is based on our empirical analysis. We applied the same evaluation metrics to Approach II Both numbers are higher in the Approach I. While in Approach II the full natural language instruction, including the action phrases (e.g. turn left) is given to GPT-4o, we hypothesize that the model has trouble grounding actions and/or landmarks and Approach I benefits from explicit decomposition of different stages. Since there were cases in which multiple top grounding



(a) Bedroom

(b) Air Vent

Figure 3. SigLIP vs. VLMaps Query Result for Last Landmark Indexing

Prompting for Landmark Grounding Score Extraction Per Panorama

USER PROMPT:

Given the panorama image and the following landmark phrase sequence, please give me a list of binary scores, with the same length as the landmark phrase sequence.

Each score is 1 if you are confident the landmark is nearby and visible in the panorama image; otherwise, it is 0.

Only return the list of scores without any other output.

Landmark phrase sequence: ['door', 'table', 'kitchen bar', 'chair', 'small hallway', 'bedroom']

GPT-4o RESPONSE:

[[1, 1, 1, 1, 0, 0],
[1, 1, 1, 1, 0, 0],
[1, 1, 1, 1, 0, 0],
[1, 1, 1, 1, 0, 0],
[1, 1, 1, 1, 0, 0],
[1, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 0]]

Figure 4. GPT-4o Landmark Grounding Score Extraction

	NUM	HYPO PATH GEN	APPROACH I		APPROACH II	
	Episodes	Accuracy (%)	nDTW (%)	Accuracy (%)	nDTW (%)	Accuracy (%)
`8WUmhLawc2A`	21	66.7	88.68±0.0	57.10±0.0	87.34±0.52	52.38±6.73
`JeFG25nYj2p`	21	61.9	87.51±0.13	52.38±0.0	88.92±0.64	49.20±5.93
`mJXqzFtmKg4`	21	66.7	91.21±0.07	66.70±0.0	90.08±0.21	57.14±0.0
`r1Q1Z4BcV1o`	21	57.1	87.96±0.30	49.20±2.24	88.69±0.42	39.68±2.24
`sT4fr6TAbpF`	21	76.2	89.25±1.05	61.90±3.88	86.68±0.34	52.38±6.73
`Average`	105	**65.72**±6.33	**88.92**±1.28	**57.45**±6.31	88.34±1.20	50.15±5.81

Table 2. Full Pipeline Quantitative Results

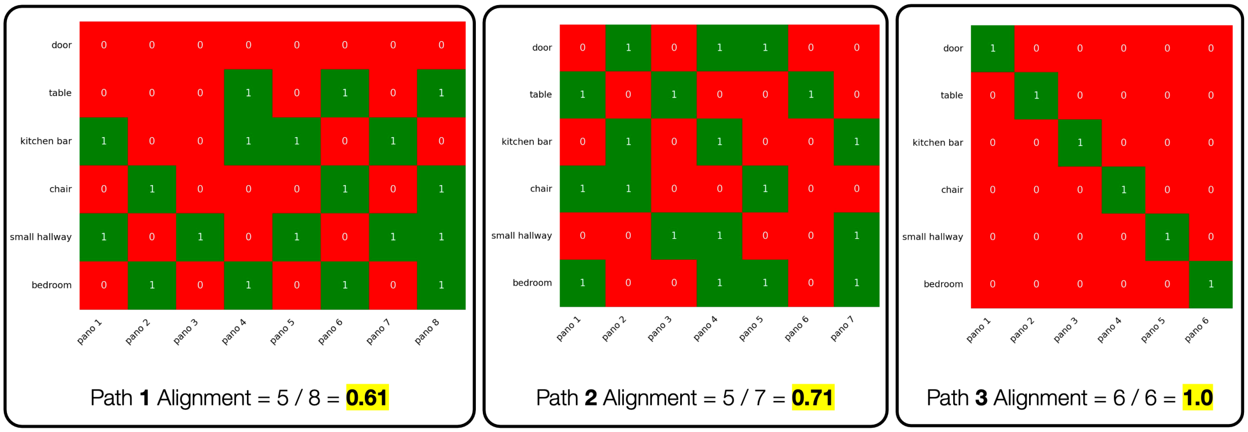


Figure 5. Sequence Alignment for Path Ranking (Pano2Land)

Prompting for Entire Sequence Scoring

USER PROMPT:

You have been given a series of RGB panorama observations and a navigation instruction.

Instruction: "Walk through the door and around the table. Turn left and walk to the end of the kitchen bar. Turn right and walk around the chair. Turn left and walk into the small hallway then go into the bedroom."

Please provide a rating from 1 to 5 on how well this sequence aligns with the instruction, where:

1 = Very Poor Alignment
2 = Poor Alignment
3 = Fair Alignment
4 = Good Alignment
5 = Excellent Alignment.

Note that the excellent alignment needs to visit all the following landmarks in order:

['door', 'table', 'kitchen bar', 'chair', 'small hallway', 'bedroom']

Please first answer in the format of [RATING] = rating, then explain in detail why you give this rating.

GPT-4o RESPONSE:

[RATING] = 5

The navigation instruction outlines a clear sequence that involves moving through specific landmarks in a detailed order: the door, table, kitchen bar, chair, small hallway, and finally into the bedroom.

In evaluating the sequences, the images provided depict a coherent layout where:

1. **Door**: The entrance is visible and can be assumed to be accessible from the starting point.
2. **Table**: Following through the door, there is a dining or kitchen table which aligns with the instruction to "walk around the table."
3. **Kitchen Bar**: Turning left leads directly to this area, allowing the next part of the navigation to occur seamlessly.
4. **Chair**: As instructed, there's a movement around the chair, which should be located near or around the kitchen area.
5. **Small Hallway**: Following the instructions to turn left into a small hallway is feasible once you navigate around the chair.
6. **Bedroom**: The progression leads directly from the small hallway into the bedroom.

Therefore, since all landmarks are visited in the correct order as specified in the instruction, I rated the sequence as an "Excellent Alignment" with a score of 5.

Figure 6. GPT-4o Full Prompting for Entire Sequence Scoring

scores or GPT-4o rating scores would exist, we repeated the process of picking the path with the highest score randomly up to 3 times and reported the mean and standard deviation.

4. Limitations

There are specific limitations to our approach that we'd like to elaborate on. Firstly, our approach only works in the previously explored environments, given the topological map. Secondly, it only works in cases where the natural language instruction is landmarks-based and is not heavily based on spatial and temporal phrases, action phrases, and absolute metric distances. Since our pipeline is modular and not trained end-to-end, drawbacks of each module, especially the early stages of the LLM landmark extraction and VLM retrieval, propagate the errors to later stages of PANO2LAND alignment or GPT-4o ranking. The quality of the path hypotheses eventually determines the upper bound on the ranking computed by GPT-4o or any other VLM being used.

5. Conclusion

In this work, we introduced a modular approach for the vision-and-language navigation (VLN) task based on the R2R-Matterport3D dataset [1, 4] within the Meta Habitat Simulator [21, 27]. Our approach assumes that the agent has built a topological map in the exploration stage. We then use LLM to extract the sequence of landmarks the agent needs to visit, retrieve the top-k goal locations, and rank the path hypotheses to select the one with the highest alignment with the natural language instructions as the final answer. For the task, the approach demonstrates the superiority of the topological map with per-node panoramas to an open-vocabulary semantic occupancy map for land-mark grounding and goal retrieval. The overall performance on this benchmark is mainly affected by the zero-shot capabilities of VLM's to ground special landmark names in the panoramas. Future improvements can be achieved by fine-tuning the existing VLMs on navigation tasks and deploying the agent in previously unseen environments by seamlessly integrating the exploration and navigation part.

References

[1] 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, 2, 3, 7

[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. 2

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020. 2

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 1, 7

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 2

[6] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021. 2

[7] Vishnu Sashank Dorbala, Gunnar Sigurdsson, Robinson Piramuthu, Jesse Thomason, and Gaurav S Sukhatme. Clip-nav: Using clip for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2211.16649*, 2022. 2

[8] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. *Advances in neural information processing systems*, 31, 2018. 2

[9] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15460–15470, 2022.

[10] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1643–1653, 2021. 2

[11] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 1, 2, 3, 4

[12] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022. 2

[13] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the

path: Instruction fidelity in vision-and-language navigation. *arXiv preprint arXiv:1905.12255*, 2019. 4

[14] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. 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, 2023. 2

[15] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 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, 2020. 2

[16] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2020. 2, 3

[17] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2

[18] Jason Xinyu Liu, Ziyi Yang, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah. Grounding complex natural language commands for temporal tasks in unseen environments. In *Conference on Robot Learning*, 2023. 2

[19] Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene-and object-aware transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34: 7357–7367, 2021. 2

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 4

[21] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 7

[22] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13215–13222. IEEE, 2021. 3

[23] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023. 3

[24] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*, 2019. 2

[25] Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. 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, 2022. 2

[26] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6629–6638, 2019. 2

[27] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. *ArXiv*, abs/2210.05633, 2022. 7

[28] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 3, 4

[29] Wanrong Zhu, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Xin Eric Wang, Qi Wu, Miguel Eckstein, and William Yang Wang. Diagnosing vision-and-language navigation: What really matters. *arXiv preprint arXiv:2103.16561*, 2021. 2