Exploring the effectiveness of Small-Scale Vision and Language Models for Vision and Language Navigation Tasks in Continuous Environments

Wesley Chiu, Abdulrahman Altahhan

University of Leeds, School of Computing, ODL MSc in AI, UK.
{od22wc, a.altahhan}@leeds.ac.uk

Abstract. Vision-and-Language Navigation (VLN) is a rapidly evolving field of research that aims to enable an embodied agent to follow textual instructions given in natural language to navigate through an unseen environment to a goal position. Existing approaches to this ask all into two main categories: "specialist" models that have been constructed and trained specifically to solve this task, and zero-shot or few-shot models that aim to leverage the implicit knowledge within Large Language Models (LLMs) and Vision Language Models (VLMs). Prior work in the latter approach have used the most powerful models (70B+ parameters). This work aims to evaluate the suitability of using a lighter-weight variant of an existing open-sourced model (Qwen2-VL) as the primary VLM, which would allow it to be run "on-device" rather than being connected to a broader network. The experiments in a simulated environment demonstrates that the current ability of the lighter-weight model is not yet fit for purpose, failing to reach the success rates seen in approaches that use the full-sized variants.

Keywords: Vision-and-Language Navigation, Vision Language Model, Large Language Model, Prompt Engineering, Prompting

1 Introduction

Vision and Language Navigation (VLN) is a rapidly evolving field of research, which tasks an embodied agent to follow a set of textual instructions given in natural language to navigate a previously unseen 3D indoor environment. Whilst initial research leveraged models such as RNNs and CNNs to process visual and natural language data, the introduction of Transformer-based models [42] has accelerated capabilities in Natural Language Processing (NLP) and Computer Vision (CV) - both key components in the VLN task. The rapid development of Large Language Models (LLMs) and Vision-Language Models (VLMs) has further intensified the volume and pace of research in VLN, with each advance in LLM and VLM competency directly benefiting the VLN task. The traditional VLN task, as proposed by Andersen, et al. in 2017 [2], has the agent navigate between pre-defined nodes in the environment (sometimes referred to as a navigation graph), which essentially reduces the VLN task to a vision-based

graph-search problem; models trained in this manner have difficulty translating to performance in the real-world [1]. To help combat this gap, the VLN Continuous Environment (VLN-CE) task was introduced, and has no such predefined navigation points, requiring agents to take low-level actions to reach a navigable point in the environment. This introduces new challenges for the models to overcome, such as avoiding getting stuck on obstacles and dealing with distances [20].

Approaches to the VLN task (whether graph-based VLN or VLN-CE) that leverage existing LLMs and VLMs typically fall into two broad categories: 'generalist' models or 'specialist' models.

Generalist models [**chen2024mapgptmapguidedpromptingadaptive**, 28, 47] use 'off-the-shelf' models such as OpenAI's ChatGPT [32] and aim to leverage the implicit knowledge, NLP ability, and strong reasoning skills within LLMs and VLMs [19, 34] to navigate the environment. These approaches rely on specific prompting to generate historical trajectories, make navigation decisions, and to monitor progress.

Specialist models [17, 48, 7, 16] are built from the ground-up, typically with specialist sub-models, to tackle the VLN task. These approaches exhibit stronger performance in VLN tasks compared to the generalist approach, but are less unable to take advantage of more competent LLMs or VLMs as they are released in the same 'plug-and-play' manner of generalist models.

Both approaches typically leverage larger, complex language and vision models, requiring powerful hardware with high amounts of memory to run during inferencing. The development of Low Rank Adaptation (LoRA) finetuning [18] combined with model quantization citation required allows for a LLM or VLM to be trained and run on hardware with more limited memory. This is particularly relevant to the VLN task, as lighter-weight models can be run more readily on smaller robots with more limited hardware, broadening the application of these models.

This work explores the effectiveness of using a small, finetuned, quantized open-sourced model as the primary VLM to tackle the VLN-CE task and examines its performance against other generalist approaches.

2 Literature Review

Vision and Language Navigation (VLN) The ability for an embodied agent to navigate a previously unseen environment purely by natural language instruction is a field of research that has seen increasing interest since the task was formally introduced by Andersen et al. in 2017 [2]. The VLN task requires that an embodied agent be able to navigate an unstructured and previously unseen environment by following navigation instructions provided in natural language. Early research focused on the use of Recurrent Neural Networks (RNNs) [2, 29, 44, 24]; however, with the introduction of the Transformer archicture [42] and the subsequent development of the Large Language Model (LLM) [37, 41], recent research has focused on leveraging the NLP capabilities of LLMs; these approaches

can be broadly categorised into two different approaches: a text-based approach and an integrated approach.

Textual vs Integrated Approaches The text-based approach seeks to address the challenge of multi-modality by translating visual features from the agent's observations into textual space. LangNav [33] and NavGPT [47], for example, used image captioning models such as BLIP [23, 22] and DETR [49] to caption images and identify objects in those images; this textual information was then passed to GPT4 for decision making and other navigation tasks; OpenNav [36] takes a similar approach with open-sourced models. Variations on this approach include DiscussNav [28], which uses a similar mechanism to enable discussions between its multiple domain experts, as well as NavCOT [25], which asks the LLM to predict information about observations from future states.

Integrated approaches also leverage LLMs and image captioning or vision transformer models [11], but integrates both models in the latent space through pre-training. These models are typically more performant than the text-based approaches, as the richness of the image data remains captured and transferred between the vision and language models [33]. Early works in VLN typically required the fusion of text and image encoding via pretraining, such as PREVALENT [15], the CMA model [20], and Airbert [14]. Recent models build on this work and also grow in complexity as researchers work to capture more proficiency in the VLN task. HAMT [7] combined BERT [10] encodings with a custom-trained Vision Transformer [11] and a form of spatial-temporal encoding. Nav-GPT2 [48] integrates LLMs with InstructBLIP [9], a VLM, via a vision encoder [13]. In a slightly different approach, SayCan [19] connects an LLM with a model trained via Reinforcement Learning to robotic affordance. These models, at time of release, were all State of the Art (SOTA) models, and continue to be referenced as benchmarks against new models; however, to obtain such proficiency, extensive pretraining and finetuning was required.

Specialist Modules Independent of the approach, VLN models typically contain multiple "modules" that specialise in particular tasks [45]; these are either separate models or different types of prompts fed to a single model. Common modules include the action-decision (or policy) module and a progress monitor, which usually neccesitates a navigation history module. LangNav [33] uses GPT-4 to both select the action and update historical trajectory by using different prompt templates. NavGPT [47] takes a similar approach by a Prompt Manager to dynamically change the prompt in response to changes in the agent's state. Some approaches [7, 16] have improved progress monitoring by creating specialist architectures to enhance the usefulness of the historical trajectory, whilst other work [8, 6] seeks to retain historical trajectory as a graph.

Vision Language Models Early research in VLN required researchers to tackle the task of multi-modal input by integrating separately trained NLP and CV models. Further research in multi-modal models have developed more tightly

integrated foundation models such as InstructBLIP [9], which allow researchers to finetune a pretrained multi-modal model. More recent research has expanded the competency of multi-modal foundation models, with notable closed-sourced models such as OpenAI's GPT4-V [32] and o1 [31] models, as well as Google's Gemini [40], whilst open-sourced models, such as LLaVa [27] and Qwen2-VL [43], provide a lower-cost alternative.

Challenges in Continuous Environments ... (talk about Sim2Real challenges) (speak about work that used node-based navigation)
However, these models typically rely on translating visual data into purely textual information, which risks a loss of information in the process [33]; as a result, these models typically underperform against specialist models. The strength of these models lie in the fact that no pre-training is required, and as LLMs and VLMs continue to improve, this increase in competency can directly translate into improved VLN performance without the need for pre-training or adaptation of a complex model structure.

3 Methodology

The approach taken in this work, as illustrated in Figure 1, comprises of using a single VLM as the primary component of the model to help the model understand where it located in the environment, track its historical trajectory, and to make decisions on its next action. A small component of this architecture leverages an LLM to break down the natural language instructions into discrete "Navigation Checkpoints", which is in line with prior work [6, 33, 47].

3.1 Problem Formulation

The VLN-CE task asks an embodied agent to autonomously navigate from a specified starting point to a goal location in an unseen environment by following a path described by natural language instructions. That is, given a natural language navigation instruction $\mathcal{I}$ and a starting state S_0, at each step t the agent must determine its current location in the environment and its current progress to take an action a_t that will lead it closer to the goal state S_G. The agent determines through observations O_t obtained from its sensors. In this work, O_t is comprised of colour images I from N different views, each set at a different angle, from the egocentric perspective of the agent. Additionally, the agent is also able to observe the environment through an egocentric panoramic image $I_{p,t}$. That is, the observations at each step is represented by $O_t = \{o_{p,t}, o_{0,t}, ..., o_{N,t}\}$. At each step, the agent must choose to take an action which corresponds to an image direction (excluding $o_{p,t}$), i.e. $a_t \in \{o_{0,t}, ..., o_{N,t}\}$, unless a specific direction is determined to result in a collision with the environment, in which case that particular direction is removed from the action set.

The agent is deemed to have successfully reached S_G if the final state of the agent S_T, where T is the total number of steps taken, is within 1.0 metres of S_G.

3.2 Framework and Prompts

Unlike prior work that used separate Vision and Language models [47, 19] which require pre-training to combine the different models, this work proposes the use of a singular VLM to manage both the image and textual inputs.

The VLM is used for inference for three different purposes: to track Navigation History, Decide the Next Checkpoint to work toward, and to Decide the Next Step. Only one task, creating Inferred Navigation Checkpoints, is not given to the VLM, and is instead given to the LLM.

Inferred Navigation Checkpoints The navigation instructions provided by the RxR dataset is given in natural language, and can include discourse markers that may not be relevant to the navigation itself (e.g. `"Okay, now ..."`) and may be confusing or difficult to track progress against. To address this, this work follows prior work [36, 48] an LLM is used to convert $\mathcal{I}$ into a list of Navigation Checkpoints, which is easier for the model to track progress against. That is, for each $\mathcal{I}$, a series of M Navigation Checkpoints $C = \{C_0, ..., C_M\}$ are created such that by progressing from c_0 (the starting point of the instruction) to c_M will result in a successful trajectory (i.e. $S_T \leq S_G \pm 1.0$ metres).

Navigation History A key component to the success of VLN agents is that at the time of deciding the optimal a_{t+1} the agent has the capability to retain in its context a history of its navigational trajectory [7, 16].

In this work, at each step, the VLM is prompted to generate h_t, which is a summary of the reasoning given for selecting a_t and the agent's current location based on $o_{p,t}$. The VLM is provided with C, $o_{p,t}$, a_t and the reasoning associated with its selection as the best action, as well as the Navigation History $H_{t-1} = \{h_{t'}, ..., h_{t-1}\}$, where $t' = \max(t - 20, 0)$. Once generated, h_t is appended to the Navigation History, such that $H_t = \{h_{t'+1}, ..., h_t\}$. The Navigation History is limited to only the past 20 steps to manage VRAM limitations in the hardware; however, this also prevents actions taken much earlier, which are unlikely to be relevant to the current action, from entering the VLM's context window, thereby removing some noise from the data [16].

Deciding Next Checkpoint Each Navigation Checkpoint c_m acts as an medium-term goal for the agent to navigate toward, which is particularly important in trajectories where S_G is not visible from S_0. At each step, the VLM is provided with C, O_t, and H_t and is then prompted to state the list of all the Navigaton Checkpoints the agent has already achieved (c_0 to c_{m-1}), the Navigation Checkpoint the agent should now be working toward c_m, and the instructions associated with c_m.

Deciding Next Step The VLN-CE task requires the agent to decide at each step which of the low-level actions would be most likely to lead to c_m. At each step, the VLM is provided with O_t and the reasoning associated with c_m described in the "Deciding Next Checkpoint" section above. The VLM is then

prompted to repeat the instructions for c_m and to select the o_n that is most aligned with that Navigation Checkpoint. During prompting, the list of available actions for that step (A_t) is modified to remove the action `"Forward"` if the simulation detects a collision if that step were to be taken; this is an emulation of a Laser Distance Scanner providing collision information to the agent and provides some environmental feedback to the agent.

3.3 Finetuning Data Generation for Continuous Environment

Each natural language navigation instruction provided by the RxR dataset is paired with a set of waypoints that describes the path that the original annotator used to create the instructions. This path can be referred to as the "gold label" trajectory that the agent should follow to reach the goal in as direct an approach as possible. As the instructions were created using the Matterport3D simulator [5], a navigation-graph based simulator, the provided waypoints do not describe a path that is relevant to the VLN CE task.

Gold Label Trajectory The original trajectories from the RxR *train* split were used to determine a new "gold label" trajectory for use in the continuous environment (CE-Trajectory). The agent was placed at the starting state S_0, and at each step t, all potential actions $\{a'_{t,0}, ..., a'_{t,N}\}$ were assessed by determining the state that each action would result in $\{S'_{t+1,0}, ..., S'_{t+1,N}\}$. The next best action was determined in a greedy manner by the state s_{t+1} that was closest to the next waypoint in the original trajectory; this was repeated until $S_T \leq S_G \pm 1.0$ metres.

Some of the RxR annotations and waypoints describe trajectories and actions that the agent specified in this work cannot take, or would require a material deviation from the original trajectory to achieve. For example, some of these instructions may include: `"Hop over the coffee table"`, `"Hop over the table"`, or navigating to an outdoor staircase. To avoid these trajectories, any trajectories where $T > 50$ were rejected, resulting in 3,304 eligible CE-Trajectories.

Generating Training Prompts The training data required for finetuning the VLM requires that the training data include the prompts for the `user` and `assistant` roles. These prompts were generated by recreating the experience described in the "gold label" trajectories and capturing the prompts used in the Navigation History and Progress Monitor modules. A slightly adjusted version of the Action Decider module, the Action Rationalizer, was created to *rationalise* the decision behind the provided action a_t at each step. The same process was used to create the calibration dataset for use in quantization.

4 Experiment

4.1 Datasets and Environment Simulation

Dataset The Matterport3D [5] dataset is a well known and commonly used dataset in VLN research. It comprises of 194,400 colour and depth (RGB-D) images to create panoramas at 10,800 different (internal) locations from 90 different residential buildings. These panoromas allows a Matterport3D Simulator user to view the environment in full in 360°, and also allows users to "teleport" from one panoramic "node" to another. These panoramic nodes form the "navigation-graph" used in traditional or discrete VLN models. The Room-across-Room (RxR) dataset [3, 21] was used for the natural language instructions, which comprises of over 126,000 natural language instructions (annotations) across three different languages to describe traversal through 16,500 different paths in the Matterport3D dataset. For this work, only the en-US guide annotations were considered, which comprises of 13,992 annotations across 13,992 unique paths. Finally, the annotations are separated into the following splits: *train*, *seen val*, and *unseen val*. In order to align with prior work, the *train* split was used for generating finetuning data, whilst the *unseen val* split was used to test the model.

Environment Habitat-Sim [38, 39, 35] was used as the simulated environment, which converts the navigation graph-based Matterport3D dataset into continous environments. The environment was also set to allow sliding, which allows the agent to slide along an obstacle rather than being stuck against it. The agent in the simulator was given a colour sensor, through which colour images could be captured for the agent to observe the envrionment in. The agent was able to observe its environment in 6 different directions: Forward (0°, straight ahead), Forward-left (45°to its left, as measured from the forward direction), Left (90°to its left), Forward-right (45°to its right), Right (90°to its right), and Behind (180°from its forward direction); each observation has a 90°horizontal and vertical field of view (FOV). The agent is able to take an action that corresponds to each of these observations, where "Forward" would result in a forward step of 1.0 metres, whilst all other actions would result in the agent turning to face that direction (e.g. "Forward-left" would result in turning 45°to the left). The agent was also provided a colour sensor to create a panoramic colour observation, with an effective horizontal FOV of 360°and a vertical FOV of 90°.

4.2 Implementation Details

This work used a finetuned and quantized open-sourced VLM as the primary model to navigate an environment. All finetuning, quantization and inferencing of the model was run on a desktop computer with an AMD Ryzen 7 7800x3D CPU, 32GB of RAM, and an NVIDIA RTX4090 with 24GB of VRAM.

Qwen2-VL-7B-Instruct [4, 43] was chosen as the base model for the VLM due to its strong performance relative to other open-sourced models [12, 30] of similar parameter size. The model was finetuned using Low Rank Adaptation

(LoRA) [18], and quantized using Activation-aware Weight (AWQ) optimisation [26]. Both model finetuning and qunatization were conducted through LLaMa-Factory [46].

OpenAI's ChatGPT 4 [32] was chosen as the LLM, which was used solely to convert the natural language navigation instructions from the RxR dataset into a structured list of Navigation Checkpoints.

4.3 Experiment Results

Results were ...

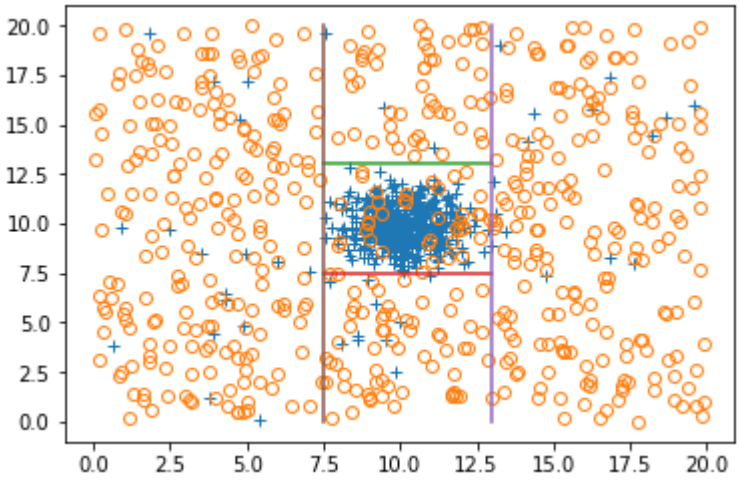

Fig. 1. Caption

In Fig. 2 that $\alpha = n^2$. In eq. (1) we have shown that the $1 + 2 + 3 + 4 = 4 \times 5/2 = 10$.

$$\sum_{i=1}^{n} y = 10$$
$$M = \beta^2$$
$$\boldsymbol{B}^\top = \boldsymbol{\Lambda}^2$$

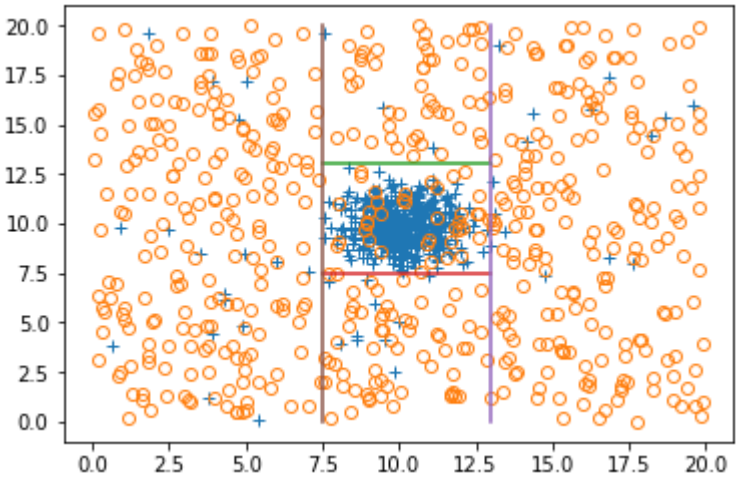

Fig. 2. Caption

$$\sum_{i=1}^{n} i = \frac{(n+1)n}{2} \tag{1}$$
$$\beta^2 = \alpha_i^n$$

5 Reference Code bit

This is how you reference stuff: 1

6 Experiment Results

6.1 Experiment 1

Exp Note that the figure and the tables might be laid out on another page. Do not worry about that, and do not attempt to change it. Leave this to Latex.

Table 1. This is a table

Year	World	Duration
8000 B.C.	5,000,000	10
50 A.D.	200,000,000	20
1650 A.D.	500,000,000	30

7 Conclusion and Future Work

We have conducted a study on ...

References

[1] Peter Anderson et al. "Sim-to-Real Transfer for Vision-and-Language Navigation". In: *Proceedings of the 2020 Conference on Robot Learning*. Ed. by Jens Kober, Fabio Ramos, and Claire Tomlin. Vol. 155. Proceedings of Machine Learning Research. PMLR, 16–18 Nov 2021, pp. 671–681. URL: https://proceedings.mlr.press/v155/anderson21a.html.

[2] Peter Anderson et al. "Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3674–3683. DOI: 10.1109/CVPR.2018.00387.

[3] Peter Anderson et al. "Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[4] Jinze Bai et al. "Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond". In: *arXiv preprint arXiv:2308.12966* (2023).

[5] Angel Chang et al. "Matterport3D: Learning from RGB-D Data in Indoor Environments". In: *International Conference on 3D Vision (3DV)* (2017).

[6] Jiaqi Chen et al. "MapGPT: Map-Guided Prompting with Adaptive Path Planning for Vision-and-Language Navigation". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 9796–9810. DOI: 10.18653/v1/2024.acl-long.529. URL: https://aclanthology.org/2024.acl-long.529/.

[7] Shizhe Chen et al. "History aware multimodal transformer for vision-and-language navigation". In: *Advances in neural information processing systems* 34 (2021), pp. 5834–5847.

[8] Shizhe Chen et al. "Think Global, Act Local: Dual-scale Graph Transformer for Vision-and-Language Navigation". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 16516–16526. DOI: 10.1109/CVPR52688.2022.01604.

[9] Wenliang Dai et al. *InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning*. 2023. arXiv: 2305.06500 [cs.CV]. URL: https://arxiv.org/abs/2305.06500.

[10] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423/.

[11] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[12] Hugging Face. *OpenVLM Leaderboard*. https://huggingface.co/spaces/opencompass/open_vlm_leaderboard. [Online]. Accessed: 2025-01-11. 2024.

[13] Yuxin Fang et al. "EVA: Exploring the Limits of Masked Visual Representation Learning at Scale". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 19358–19369. DOI: 10.1109/CVPR52729.2023.01855.

[14] Pierre-Louis Guhur et al. "Airbert: In-Domain Pretraining for Vision-and-Language Navigation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 1634–1643.

[15] Weituo Hao et al. "Towards Learning a Generic Agent for Vision-and-Language Navigation via Pre-Training". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 13134–13143. DOI: 10.1109/CVPR42600.2020.01315.

[16] Keji He et al. "Memory-Adaptive Vision-and-Language Navigation". In: *Pattern Recognition* 153 (2024), p. 110511. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2024.110511. URL: https://www.sciencedirect.com/science/article/pii/S0031320324002620.

[17] Yicong Hong et al. "Vln bert: A recurrent vision-and-language bert for navigation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2021, pp. 1643–1653.

[18] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *CoRR* abs/2106.09685 (2021). arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[19] Alex Irpan et al. "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances". In: 2022. URL: https://arxiv.org/abs/2204.01691.

[20] Jacob Krantz et al. *Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments*. 2020. arXiv: 2004.02857 [cs.CV]. URL: https://arxiv.org/abs/2004.02857.

[21] Alexander Ku et al. "Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding". In: *Conference on Empirical Methods for Natural Language Processing (EMNLP)*. 2020.

[22] Junnan Li et al. *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 2023. arXiv: 2301.12597 [cs.CV]. URL: https://arxiv.org/abs/2301.12597.

[23] Junnan Li et al. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. arXiv: 2201.12086 [cs.CV]. URL: https://arxiv.org/abs/2201.12086.

[24] Xiujun Li et al. *Robust Navigation with Language Pretraining and Stochastic Sampling*. 2019. arXiv: 1909.02244 [cs.CL]. URL: https://arxiv.org/abs/1909.02244.

[25] Bingqian Lin et al. *NavCoT: Boosting LLM-Based Vision-and-Language Navigation via Learning Disentangled Reasoning*. 2024. arXiv: 2403.07376 [cs.CV]. URL: https://arxiv.org/abs/2403.07376.

[26] Ji Lin et al. "AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration". In: *Proceedings of Machine Learning and Systems*. Ed. by P. Gibbons, G. Pekhimenko, and C. De Sa. Vol. 6. 2024, pp. 87–100. URL: https://proceedings.mlsys.org/paper_files/paper/2024/file/42a452cbafa9dd64e9ba4aa95cc1ef21-Paper-Conference.pdf.

[27] Haotian Liu et al. *Visual Instruction Tuning*. 2023. arXiv: 2304.08485 [cs.CV]. URL: https://arxiv.org/abs/2304.08485.

[28] Yuxing Long et al. "Discuss before moving: Visual language navigation via multi-expert discussions". In: *arXiv preprint arXiv:2309.11382* (2023).

[29] Chih-Yao Ma et al. "The Regretful Agent: Heuristic-Aided Navigation Through Progress Estimation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6725–6733. DOI: 10.1109/CVPR.2019.00689.

[30] Trong-Hieu Nguyen-Mau et al. "Enhancing Visual Question Answering with Pre-trained Vision-Language Models: An Ensemble Approach at the LAVA Challenge 2024". In: *Proceedings of the Asian Conference on Computer Vision (ACCV) Workshops*. Dec. 2024, pp. 275–286.

[31] OpenAI. *Learning to reason with LLMs*. https://openai.com/index/learning-to-reason-with-llms/. [Online]. Accessed: 2025-01-10. 2024.

[32] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[33] Bowen Pan et al. *LangNav: Language as a Perceptual Representation for Navigation*. 2024. arXiv: 2310.07889 [cs.CV]. URL: https://arxiv.org/abs/2310.07889.

[34] Fabio Petroni et al. "Language Models as Knowledge Bases?" In: *CoRR* abs/1909.01066 (2019). arXiv: 1909.01066. URL: http://arxiv.org/abs/1909.01066.

[35] Xavi Puig et al. *Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots*. 2023.

[36] Yanyuan Qiao et al. *Open-Nav: Exploring Zero-Shot Vision-and-Language Navigation in Continuous Environment with Open-Source LLMs*. 2024. arXiv: 2409.18794 [cs.RO]. URL: https://arxiv.org/abs/2409.18794.

[37] Alec Radford. "Improving language understanding by generative pre-training". In: (2018).

[38] Manolis Savva et al. "Habitat: A Platform for Embodied AI Research". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[39] Andrew Szot et al. "Habitat 2.0: Training Home Assistants to Rearrange their Habitat". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[40] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[41] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[42] Ashish Vaswani et al. *Attention is All you Need*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[43] Peng Wang et al. "Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution". In: *arXiv preprint arXiv:2409.12191* (2024).

[44] Xin Wang et al. "Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6622–6631. DOI: 10.1109/CVPR.2019.00679.

[45] Wansen Wu, Tao Chang, and Xinmeng Li. *Vision-Language Navigation: A Survey and Taxonomy*. 2022. arXiv: 2108.11544 [cs.CV]. URL: https://arxiv.org/abs/2108.11544.

[46] Yaowei Zheng et al. "LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Bangkok, Thailand: Association for Computational Linguistics, 2024. URL: http://arxiv.org/abs/2403.13372.

[47] Gengze Zhou, Yicong Hong, and Qi Wu. "NavGPT: Explicit Reasoning in Vision-and-Language Navigation with Large Language Models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 7. 2024, pp. 7641–7649. DOI: 10.1609/aaai.v38i7.28597.

[48] Gengze Zhou et al. "NavGPT-2: Unleashing Navigational Reasoning Capability for Large Vision-Language Models". In: *Computer Vision – ECCV 2024*. Ed. by Aleš Leonardis et al. Cham: Springer Nature Switzerland, 2025, pp. 260–278. ISBN: 978-3-031-72667-5.

[49] Xizhou Zhu et al. *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. 2021. arXiv: 2010.04159 [cs.CV]. URL: https://arxiv.org/abs/2010.04159.