TEACHING VISUAL LANGUAGE MODELS TO NAVIGATE USING MAPS

Tigran Galstyan[1,2], **Hakob Tamazyan**[1,2], **Narek Nurijanyan**[1,2,3], **Hrant Khachatrian**[1,2]
[1] Yerevan State University
[2] YerevaNN
[3] American University of Armenia
{tigran,hakob,narek,hrant}@yerevann.com

ABSTRACT

Visual Language Models (VLMs) have shown impressive abilities in understanding and generating multimodal content by integrating visual and textual information. Recently, language-guided aerial navigation benchmarks have emerged, presenting a novel challenge for VLMs. In this work, we focus on the utilization of *navigation maps*, a critical component of the broader aerial navigation problem. We analyze the CityNav benchmark, a recently introduced dataset for language-goal aerial navigation that incorporates navigation maps and 3D point clouds of real cities to simulate environments for drones. We demonstrate that existing open-source VLMs perform poorly in understanding navigation maps in a zero-shot setting. To address this, we fine-tune one of the top-performing VLMs, Qwen2-VL, on map data, achieving near-perfect performance on a landmark-based navigation task. Notably, our fine-tuned Qwen2-VL model, using *only the landmark map*, achieves performance on par with the best baseline model in the CityNav benchmark. This highlights the potential of leveraging navigation maps for enhancing VLM capabilities in aerial navigation tasks.

1 INTRODUCTION

In context of huge advances in large language modeling research, more examples of successful knowledge transfer between various modalities (Kar et al., 2024; Shi et al., 2024) and knowledge sources, specifically using VLMs, appear in recent literature (Wang et al., 2023; Lin et al., 2024; Ma et al., 2024) providing promising grounds for advances in AI (particularly foundtaion models) usage for unmanned aerial vehicle (UAV) control tasks. Also, extensive research is being conducted trying to exploit VLM capabilities as an autonomus agent in various environments, e.g. self-driving vehicles (Tian et al., 2024), GUI understanding (You et al., 2024; Hong et al., 2024). However, applications of VLMs in aeral navigation tasks, particularly utilizing navigation maps, remain underexplored.

We also see growing number of various multimodal benchmarks covering various aspects of interest, e.g. multimodal reasoning (Yue et al., 2024), multilinguality (Liu et al., 2024c), safety (Liu et al., 2024b). In this work we concentrate on language-guided navigation, specifically aerial navigation benchmarks. We chose CityNav (Lee et al., 2024) for our experiments, we will discuss this decision in detail in Sec. 2. Before arriving at CityNav, we considered several other available aerial navigation benchmarks.

```
{self.prompt}
[ActionSpace] {self.action_space_description} [/ActionSpace]
[History]
{history}
[/History]
Current navigation map:  {self.predictor.IMAGE_TOKEN}
Pick one of the following options.
[Options]
0) STOP
1) MOVE_FORWARD
2) TURN_RIGHT
3) TURN_LEFT
[/Options]
```

Table 1: Context template used in zero-shot evaluations.

AerialVLN (Liu et al., 2023) is a language-goal based aerial navigation benchmark, using unreal engine for it's flight simulation, which is not ideal in our case as it will eventualy fall into the sim-to-real gap. AerialVLN also does not provide map information. We ultimately rejected AerialVLN because of the aforementioned reasons.

```
You are piloting a flying drone in a 3D space using verbal
instructions and a top-down navigation map.
The drone's current position and orientation, which is represented by
a green arrow.
The arrow points in the direction the drone is currently facing.
Landmarks, which are highlighted in red on the map.
Your task is to analyze both the verbal instructions and the map
carefully to determine the drone's next action.
Use the green arrow to understand the drone's position and
orientation, and use the red landmarks as reference points for
navigation.
Provide only the number of your choice.
```

Table 2: Initial prompt.

AVDN (Fan et al., 2022) is another aerial navigation benchmark, which employs satelite imagery as a base for simulations, but it is sub-par compared to CityNav as CityNav uses 3D scans of real cities in contrast to 2D satelite images of AVDN. AVDN also contains significantly less number of trajectories (3k vs 32k in CityNav). Other recently introduced benchmark is EmbodiedCity (Gao et al., 2024). Similar to AerialVLN it is based on a graphical simulated environment, just in this case environment is based on real world (i. e. some parts of Beijing were graphically modeled by human designers). Notable aspect of EmbodiedCity is that the overall benchmark is diveded into subproblems, mainly first-view scene understanding, embodied question answering, dialogue, task planning and embodied navigation. We think that this is a viable approach and will experiment on EmbodiedCity in our future work.

We see our main contributions in this work to be the following:

- We closely analyse the CityNav dataset, particularly highlighting the exploitablity of the way *landmark maps* are incorporated in the data. We also show that baseline models of Lee et al. (2024) are massively underperforming and are unable to adequately utilize avaliable information.
- We show that without additional post-training current visual language models struggle to comprehend navigation maps which can be a substantial hurdle in general navigation and planning tasks for VLMs.
- We successfully fine-tune Qwen2-VL on navigation map data, achieving almost perfect results in the task of reaching landmarks on a given map. Notably our fine-tuned Qwen2-VL performs on par with best baseline models from CityNav on benchmark evaluations using **only the landmark map**, disregarding all the remaining information.

```
There are the following 4 actions: 'STOP', 'MOVE_FORWARD', 'Turn
Right', and 'Turn Left'.
The 'STOP' action ends the agent's navigation when it arrived at its
destination.
The 'MOVE_FORWARD' action advances the agent by 5 meters in the
direction it is facing.
'TURN_LEFT' and 'TURN_RIGHT' cause the agent to rotate 30 degrees
counterclockwise and clockwise, respectively.
```

Table 3: Action space description.

2 CITYNAV DATASET

In this work, we chose to focus on the CityNav (Lee et al., 2024) dataset for several reasons. First, it offers trajectories operated by human pilots in an environment based on real aerial imagery, in contrast to simulation-based datasets (Liu et al., 2023; Gao et al., 2024). Second aspect in favor of CityNav is the environment in which the drone is operated. In contrast to data sets such as AVDN (Fan et al., 2022), which uses already projected 2D satellite imagery, CityNav uses 3D point clouds (Hu et al., 2022) (which can be loaded into AirSim (Shah et al., 2018)) to render more realistic drone flight simulations. We believe one of the main disadvantages of the CityNav dataset is the way *Landmarks* are used in direction instructions and generated maps. We will discuss this further in

Section 2.1. Another argument in favor of CityNav is the raw number of trajectories, more than 32K here, which we believe is the only publicly available human-curated instruction-based aerial navigation dataset of this scale that has aformentioned qualities.

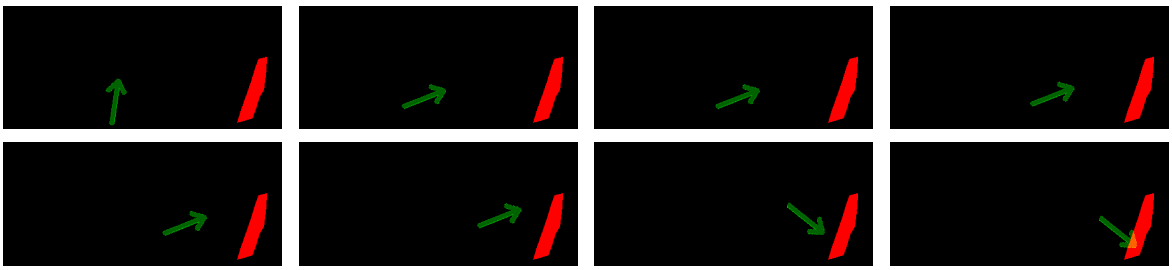

Figure 1: These landmark maps show intermediate steps of a successfull trajectory from validation set generated by following the actions chosen by Qwen-SP. The images are ordered from left to right and top to bottom. The arrow size has been artificially increased for better visualization.

2.1 Greedy Solution

A specific aspect of the CityNav dataset we think is worth looking into is the *Landmarks*. In Lee et al. (2024) we see that the baseline model has a *Navigation Map* as an input, which mainly consists of information from off the shelf models (e.g. GroundingDino (Gao et al., 2024), MobileSam (Zhang et al., 2023), LLaVa (Liu et al., 2024a)) and a landmark map, which essentially shows where the landmarks described in text instruction (e. g. street or building names) are located on the map in relation to the agent. We think this can be easily exploited which will undermine all other parts of their model (off the shelf models, RGB and Deapth image processors, etc.). To support this claim we introduce a **Dummy Landmark based Goal Predictor**, which will ignore all the other inputs and take into account only the landmark map. For every task as it's prediction **DLGP** will calculate the *center of mass* of the landmark map, i. e. the average contour points of all the landmarks mentioned in text instruction. In Table 4 you can see the results achieved by **DLGP** compared to the best model from Lee et al. (2024). Numbers from Lee et al. (2024) are cited, not reproduced. In Table 4 we can see that **DLGP** significantly outperforms the best baseline model from Lee et al. (2024) across all data splits, indicating the significant underperformance of the baseline models. In the following sections we will talk about our work on CityNav.

Table 4: Results of Zero-Shot experiments (except for MGP, which is the best result from Lee et al. (2024)). Success is measured with respect to actual CityNav target.

Model	Validation Seen			Validation Unseen			Test Unseen		
	SR	OSR	NE	SR	OSR	NE	SR	OSR	NE
MGP Lee et al. (2024)	8.69	35.51	59.7	5.84	22.19	75.1	6.38	26.04	93.8
DLGP	25.17	25.17	48.00	21.70	21.70	51.13	33.46	33.46	53.23
Qwen2-VL	0.82	5.26	195.86	0.86	4.16	205.93	1.32	6.63	185.98
LLaVA-NeXT-Interleave	0.0	12.29	466.50	0.0	10.94	455.30	0.0	14.31	466.55

Table 5: Results of Zero-Shot experiments. Success is measured with respect to landmarks mentioned in directions (not actual interest point).

Model	Validation Seen			Validation Unseen			Test Unseen		
	SR	OSR	NE	SR	OSR	NE	SR	OSR	NE
DLGP	92.87	92.87	9.42	90.99	90.99	9.98	87.82	87.82	13.14
Qwen2-VL	10.93	23.63	146.93	7.83	18.48	156.15	9.93	21.36	148.72
LLaVA-NeXT-Interleave	0.0	48.85	412.03	0.0	42.02	405.11	0.0	46.11	422.06

3 Zero-Shot Experiments

We specifically selected Qwen2-VL and LLaVA-NeXT-Interleave from the available open-source models for our experiments due to their training on multiple image-text inputs. This choice aligns with the nature of our task,

where the model must process a new input image at each step, reflecting its current navigation state. All experiments were conducted using the 7B versions of these models. Tables 1, 2 and 3 present the template and initial prompt used for zero-shot next-action prediction. We followed the standard evaluation protocols from Lee et al. (2024), assessing navigation performance using Navigation Error (NE), Success Rate (SR), and Oracle Success Rate (OSR). NE quantifies the final Euclidean distance in meters between the agent's stopping position and the goal. SR measures the proportion of episodes where the agent successfully halts within 20 meters of the destination. OSR is the proportion of episodes in which the agent's trajectory, at any point, comes within 20 meters of the target location. We conducted our experiments with a context length of 10, providing the model with the last 10 landmark maps, including the current view. The maximum number of steps was set to 120, which was sufficient to reach any point in all episodes.

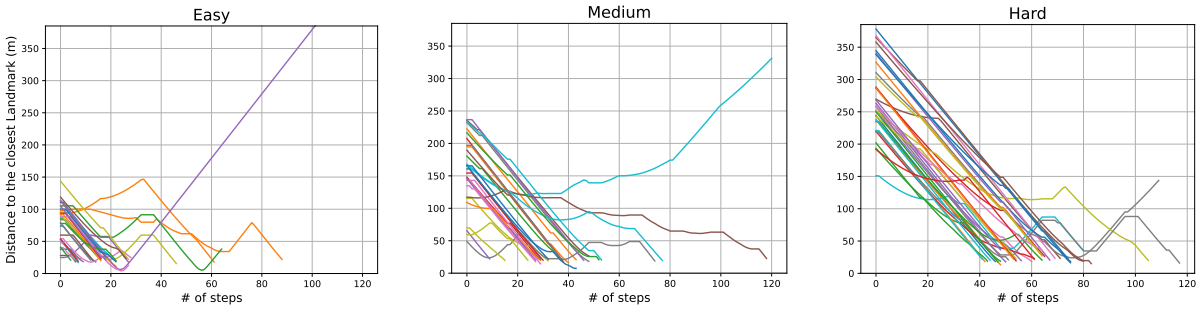

Figure 2: Distance to closest landmark during the flight trajectories accross difficulty splits of CityNav unseen test set. Each line on a plot is a simulated drone path controlled by Qwen-SP. Episodes are sampled randomly.

Table 4 summerizes main results of Zero-Shot experiments. You can see both Qwen2-VL and LLaVA-NeXT-Interleave perform poorly across all difficulty levels. We also evaluated both models using *Landmark metrics*, i. e. we interpret the landmark as our point of interest. For landmark metrics we use 30 meters to measure success as the arrow we use (see Fig. 1) overlaps with the landmark from 30 meters (agent is considered to be at the center of the arrow, not at the point). Results are shown in Table 5

We observe that modern visual language models struggle with understanding absolute and relative directions. Even simple tasks, such as determining the direction in which an arrow is pointing, were challenging for the models to comprehend and answer correctly. As a result, our focus shifted toward teaching models to follow instructions and analyze interleaved map and text input to determine the next move using the landmark map.

Table 6: Results of Finetuning experiments. Success is measured with respect to actual interest point. In this case models were prompted only to reach the landmark. 'SP' stands for 'shortest path', 'HD' - 'human driven'

Model	Validation Seen			Validation Unseen			Test Unseen		
	SR	OSR	NE	SR	OSR	NE	SR	OSR	NE
MGP (Lee et al., 2024)	8.69	35.51	59.7	5.84	22.19	75.1	6.38	26.04	93.8
Qwen-SP	6.62	9.33	78.52	7.14	9.33	80.68	6.26	9.64	96.40
Qwen-HD	4.65	7.61	109.53	5.77	8.39	110.96	5.48	9.43	140.42
Qwen-SP w/ Frozen ViT	5.06	8.22	103.36	5.60	8.54	114.32	5.46	9.23	142.36
Qwen-HD w/ Frozen ViT	4.48	7.36	144.69	4.45	7.03	159.61	4.53	8.26	185.18

Table 7: Results of Finetuning experiments. Success is measured with respect to landmarks mentioned in directions (not actual interest point). 'SP' stands for 'shortest path', 'HD' - 'human driven'

Model	Validation Seen			Validation Unseen			Test Unseen		
	SR	OSR	NE	SR	OSR	NE	SR	OSR	NE
DLGP	92.87	92.87	9.42	90.99	90.99	9.98	87.82	87.82	13.14
Qwen-SP	95.64	96.40	20.45	95.68	96.71	21.82	93.83	95.82	25.43
Qwen-HD	84.83	86.01	46.65	86.42	87.66	48.66	81.34	83.94	57.00
Qwen-SP w/ Frozen ViT	78.05	81.36	42.19	76.21	80.67	52.14	74.55	79.15	56.38
Qwen-HD w/ Frozen ViT	71.51	75.49	84.85	66.29	71.01	101.46	64.20	69.84	111.44

4 FINETUNING EXPERIMENTS

We finetuned Qwen2-VL 7B for this task using the HuggingFace TRL library. We assumed that the drone has reached a landmark if it is within 30 meters of it. To generate the dataset for reaching landmarks, we applied the following steps. First, we filtered the training trajectories, retaining only those that contained a position within 20 meters of a landmark, which resulted in keeping 80% of the trajectories. Next, all positions following this identified position were removed, and the corresponding action was modified to 'STOP'.

For imitation learning, we utilized both shortest path and human demonstration trajectories and refer to the resulting models as Qwen-SP and Qwen-HD, correspondingly. We conducted both full fine-tuning and fine-tuning with a frozen ViT.

Training was carried out with a peak learning rate of 5×10^{-5}, employing a cosine learning rate scheduler with warmup. The model was trained with a batch size of 2 per GPU across 8 NVIDIA H100 GPUs. We trained with a context length of 32 and a map size of 112×112 to ensure the model captures sufficient spatial information for navigation.

Result are shown on Tables 6 and 7. We see two important points here. First in Table 7 we see that Qwen-SP outperforms DLGP, which indicates nearly perfect performance. Second, in Table 6 we see that Qwen-SP performes on par with MGP from Lee et al. (2024), indicating strong underperformance of current baseline models on CityNav. In Table 6 we can also see two minor trends. First, shortest path guidance always outperformes human guidance. It is worth noting that in Lee et al. (2024) they observed an opposite trend. Secondly, in all our experiments freezing visiual encoding, ViT to be specific, seems to harm overall final performance.

Figure 2 illustrates the distance to the closest landmark during flight trajectories, providing insight into how Qwen-SP navigates across different difficulty splits of the CityNav unseen test set.

5 CONCLUSION AND FUTURE WORK

We demonstrated that modern vision language models struggle with seemingly trivial understanding of the simplest navigation map. To tackle this problem we fine-tuned Qwen2-VL 7B (Wang et al., 2024) on navigation map data and were able to successfully demonstrate that using simple supervised fintuning procedure Qwen2-VL is able to not only comprehend the navigation maps, but also make accurately informed action predictions based on provided maps. We also performed rigorous analysis of CityNav dataset, showing some of its exploitabilities. In future we plan to continue training VLMs as aerial navigation agents and explore other post-training recepies to enhance not only map-based action prediction, but other neccessary skills for general aerial text instruction-based navigation task.

ACKNOWLEDGEMENTS

This work was supported by the Higher Education and Science Committee of RA (Research project No 24RL-1B049). Hakob Tamazyan's work was supported by Yandex Armenia fellowship.

REFERENCES

Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Eric Wang. Aerial vision-and-dialog navigation. *arXiv preprint arXiv:2205.12219*, 2022.

Chen Gao, Baining Zhao, Weichen Zhang, Jinzhu Mao, Jun Zhang, Zhiheng Zheng, Fanhang Man, Jianjie Fang, Zile Zhou, Jinqiang Cui, et al. Embodiedcity: A benchmark platform for embodied agent in real-world city environment. *arXiv preprint arXiv:2410.09604*, 2024.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14281–14290, 2024.

Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Sensaturban: Learning semantics from urban-scale photogrammetric point clouds. *International Journal of Computer Vision*, 130(2): 316–343, 2022.

Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and Federico Tombari. Brave: Broadening the visual encoding of vision-language models. In *European Conference on Computer Vision*, pp. 113–132. Springer, 2024.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. Citynav: Language-goal aerial navigation dataset with geographic information. *arXiv preprint arXiv:2406.14240*, 2024.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26689–26699, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024a.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15384–15394, 2023.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, pp. 386–403. Springer, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024c.

Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *European Conference on Computer Vision*, pp. 417–435. Springer, 2024.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics: Results of the 11th International Conference*, pp. 621–635. Springer, 2018.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*, 2024.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In *European Conference on Computer Vision*, pp. 240–255. Springer, 2024.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.