# Towards Fine-tuning a Small Vision-Language Model for Aerial Navigation

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Visual Language Navigation (VLN) for autonomous robots presents a significant challenge, requiring models to ground textual instructions in visual environments. This paper addresses the CityNav aerial navigation benchmark by fine-tuning a small, open-source Vision-Language Model, Qwen2.5-VL-3B. Our investigation reveals that model performance is critically affected by a severe action imbalance in the training data and is substantially improved by incorporating recent flight trajectory history as an input. By addressing these factors, we achieve an 8% success rate on the Test Unseen split of CityNav, establishing a new state-of-the-art. Despite this result, we observe pronounced overfitting due to data scarcity. To mitigate this limitation, we propose a synthetic data generation strategy focused on explicitly teaching critical navigational skills, such as map interpretation. This work demonstrates that targeted, skill-based data synthesis is a promising direction for building more capable VLN agents.

## 1 Introduction

Recent advances in Vision-Language Models (VLMs) have opened new paths for tackling complex real-world tasks, including embodied navigation. Aerial navigation, in particular, presents unique challenges due to its expansive 3D search space and the need for nuanced understanding of visual and linguistic cues. Datasets like CityNav [Lee et al., 2024] provide a valuable environment for developing aerial navigation agents, offering realistic 3D settings and human-generated flight paths.

However, training effective agents on such datasets faces significant difficulties. Baseline models often struggle with generalization and can be influenced by artifacts in the training data, such as a severe imbalance in the action distribution. Our approach is to directly fine-tune a compact, open-source VLM **Qwen2.5-VL-3B-Instruct** [Wang et al., 2024], to teach it how to read the map effectively. The model receives three kind of inputs: a natural language instruction, a top-down landmark map showing the agent's position with a landmark polygon, and a first-person RGB image from the drone's perspective. Moreover, because of pronounced overfitting, we then target the core failure mode, misreading the map, with a small, purpose-built synthetic dataset that explicitly teaches the discrete geometric cases the agent must master.

Aerial VLN systems such as OpenFly [Gao et al., 2025] introduce large-scale simulated data and engineering to scale training and also employ *multi-granularity forward actions* to mitigate action imbalance. Our action grouping follows the same intuition. In parallel, FlightGPT [Cai et al., 2025] tackles UAV VLN by feeding the model a global semantic map of the entire environment (annotated with landmarks and agent pose) and reasoning over that map before emitting actions; this setting gives the model a view of its entire environment not just the drone's first-person view, which makes their problem formulation meaningfully different from ours. We show that, even under local-observation regime, careful treatment of actions and history plus targeted synthetic supervision already yields competitive navigation. Our method outperforms the more complex baseline models from the original CityNav paper, highlighting the importance of data-centric strategies for improving VLM performance on specialized tasks.

## 2 Related Works

Recent efforts in aerial vision-and-language navigation have produced several complementary benchmarks. Aerial Vision-and-Dialog Navigation (AVDN) [Fan et al., 2022] introduced dialog-driven drone control with a continuous photorealistic simulator and a human-collected dataset (~3k trajectories), emphasizing interactive instruction-following and human attention modeling. AerialVLN [Liu et al., 2023] proposed a city-scale, outdoor VLN task using a 3D simulator and near-realistic renderings across multiple city scenarios, highlighting the difficulty of spatial reasoning and height-aware control in aerial settings. More recently, OpenFly [Gao et al., 2025] introduced an automated toolchain and a very large-scale benchmark (~100k trajectories) that fuses multiple rendering engines and real-to-sim techniques to scale diversity and visual fidelity for aerial VLN research.

CityNav [Lee et al., 2024] occupies an important niche between these efforts by providing human-piloted trajectories rendered from real aerial imagery and 3D point clouds (enabling AirSim-based photogrammetric simulation) at a scale (~32k trajectories) that is larger than earlier human-curated aerial datasets. Importantly, CityNav exposes limitations in how landmark maps are incorporated and used by baseline models; this motivates our focus on teaching VLMs to better interpret landmark maps.

## 3 CityNav Dataset

In this study, we selected the *CityNav* dataset [Lee et al., 2024] due to several key advantages. First, it provides trajectories generated by human pilots in environments constructed from real aerial imagery, offering a more authentic setting compared to simulation-focused datasets [Liu et al., 2023, Gao et al., 2024]. Another strength of *CityNav* lies in the realism of its environment. Unlike datasets such as AVDN [Fan et al., 2022], which rely on pre-projected 2D satellite images, *CityNav* incorporates 3D point cloud data [Hu et al., 2022] that can be rendered in AirSim [Shah et al., 2018], enabling immersive and realistic drone flight simulations. Additionally, the dataset stands out for its scale, comprising over 32,000 human-curated trajectories, making it, to our knowledge, the largest publicly available instruction-based aerial navigation dataset with these characteristics.

## 4 Method

We use Qwen2.5-VL-3B-Instruct as the base VLM. At each step, the model takes three inputs: (i) the natural language instruction, (ii) the current landmark map image with the agent's pose, and (iii) the first-person RGB image from the drone's camera. The model autoregressively predicts one of the following actions: STOP, MOVE FORWARD, TURN RIGHT, TURN LEFT, GO UP, GO DOWN, and their grouped variants. We fine-tune the model with next-token prediction using a cross-entropy loss.

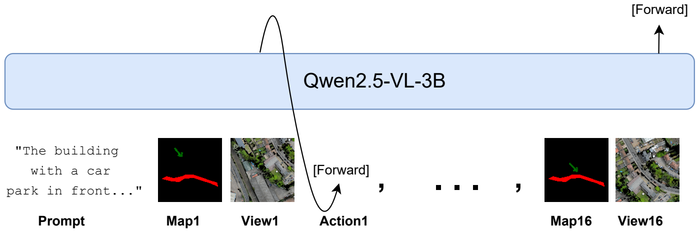


Figure 1: Overview of the method pipeline. The chart illustrates the input (map + state) and the autoregressive prediction of actions.

As shown in Figure 1, the model processes both visual and textual states to predict navigation actions.

### 4.1 Training Details

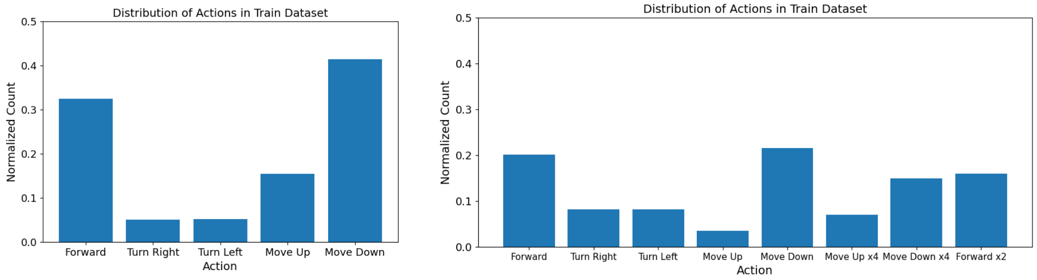
The training was carried out with a peak learning rate of $2 \times 10^{-5}$, employing a cosine learning rate scheduler with warmup. The model was trained with a batch size of 4 per GPU across 8 NVIDIA H100 GPUs. We trained with a landmark map size of $112 \times 112$ and a local view of size $224 \times 224$ to ensure both efficiency and sufficient spatial information for navigation. In all experiments, the vision encoder was frozen.

### 4.2 Action Imbalance

Initially, our action space consisted of six basic navigation commands: **stop**, **move forward**, **turn right**, **turn left**, **go up**, and **go down**. However, as shown in Figure 2(a), after analyzing the distribution of actions in the dataset, we observed a significant imbalance actions, particularly **move forward** and **move down**, occurred much more frequently than others. Since our model predicts actions token by token, this imbalance introduced a bias toward more frequently repeated actions.

To address this, we introduced *action grouping*, where consecutive repetitions of the same action are combined into a single higher-level action token. This technique has also been effectively used in other aerial navigation works, such as OpenFly [Gao et al., 2025], to balance action distributions. Specifically, we added three new composite actions: **go up ×4**, **go down ×4**, and **move forward ×2**. These tokens represent scaled versions of the original actions, for example, four consecutive 2-meter upward steps are replaced by a single 8-meter upward action, and two consecutive 5-meter forward steps are represented as a single 10-meter forward action.

This grouping strategy, as illustrated in Figure 2(b), reduced the skew in the action distribution, yielding a more balanced action space and allowing the model to generalize more effectively without overpredicting the most frequent actions.



(a) Before Action Grouping

(b) After Action Grouping

Figure 2: Action distribution in the training set (excluding the 'Stop' action). (a) shows the original imbalanced distribution where 'move forward' is dominant. (b) shows the more balanced distribution after applying action grouping.

### 4.3 History Size

The agent's ability to make informed decisions is dependent on its awareness of past actions and observations. To identify the optimal amount of historical context, we fine-tuned our model with varying history sizes: 0, 8, and 16 past landmark maps provided as input. A larger history can offer more trajectory context, helping the model understand its movement and avoid repetitive cycles. We evaluated each configuration on the Val Unseen dataset to assess generalization. The results, presented in Table 1, show a non-linear relationship; performance improved significantly from 0 to 8 history frames, but the model with 16 history frames showed a little improvement in performance, suggesting potential difficulty in processing longer sequences. Based on these findings, we selected a history size of 16 for all subsequent experiments.

Table 1: Performance on the Val Unseen split with different history sizes.

| History Size | NE ↓ | SR ↑ | OSR ↑ |
|---|---|---|---|
| 0 | 190.0 | 0.82 | 0.91 |
| 8 | **120.0** | 6.05 | 10.90 |
| 16 | 160.0 | **6.11** | **14.46** |

## 5 Synthetic Dataset

A key challenge we observed was significant overfitting when training on the CityNav dataset alone, likely due to its limited size and diversity for fine-tuning a large VLM. As shown in Figure 3, the training loss consistently decreases while the validation loss stagnates, indicating that the model is memorizing the training data rather than learning generalizable navigation skills.

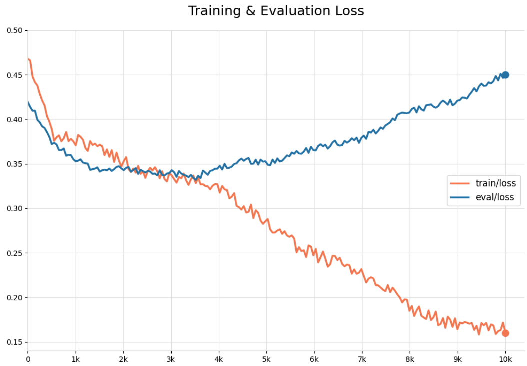


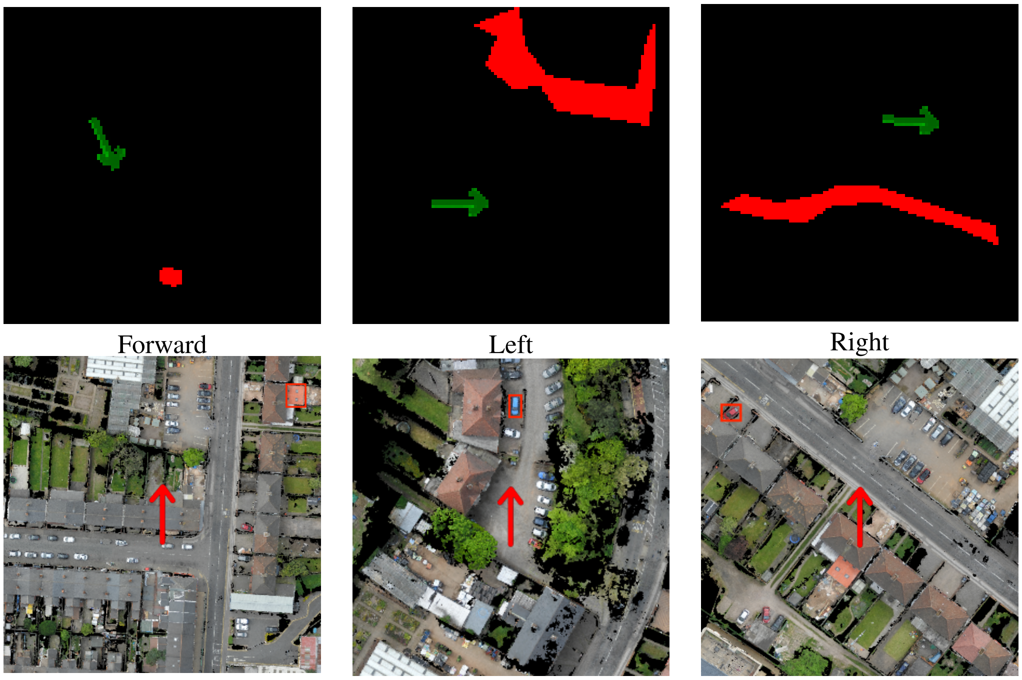
Figure 3: Training and validation loss curves. The divergence between the two curves indicates significant overfitting on the CityNav training data after several epochs.

To mitigate this and enhance our model's ability to perform the navigation task, we generated a synthetic dataset to support fine-tuning. This dataset was specifically designed to teach fundamental navigation concepts and enable the model to better generalize to different scenarios.

We categorized the drone's position and corresponding action into six distinct classes:

1. **Far from the landmark, random arrow direction**
   The drone is positioned far from the landmark, and the direction arrow is random. In this case, the model should predict a corrective action—either turning *left* or *right*—based on the landmark's relative position.
2. **Far from the landmark, arrow towards the landmark**
   The drone is far from the landmark, but the direction arrow is correctly pointing towards it. The expected action in this scenario is to move *forward*.
3. **Near the landmark, target not visible**
   The drone is close to the landmark, but the target is no longer visible. This situation introduces ambiguity, as there is no definitive correct next step. Since it cannot be supervised, no synthetic data was generated for this class.
4. **Near the landmark, target visible, random arrow**
   The drone is near the landmark, the target is visible, but the arrow direction is random. Here, the correct action is to adjust orientation by turning *left* or *right*.
5. **Near the landmark, target visible, arrow towards the target**
   The drone is close to the landmark, the target is visible, and the arrow points towards it. The drone should proceed *forward* or *downwards*, depending on proximity and altitude.
6. **Very close to the landmark and the target**
   When the drone is within 20 meters of both the landmark and the target, the only correct action is to *stop*.

We generated a large number of synthetic data points covering five of these six cases. The third scenario, due to its inherent ambiguity and lack of a definitive action label, was intentionally excluded. We generated two distinct sets of this synthetic data: one for training, created from the original CityNav training samples, and another for evaluation, generated from the remaining validation and test samples. Furthermore, each of these sets was divided into three subsets, each dedicated to a specific corrective action: *forward*, *turn left*, and *turn right*. This structured approach allowed us to systematically teach and evaluate the model's response to different navigational scenarios. The synthetic samples were then incorporated into the model's fine-tuning process, improving its decision-making capabilities in real-world navigation tasks.



Forward Left Right

Right Forward/Down Left

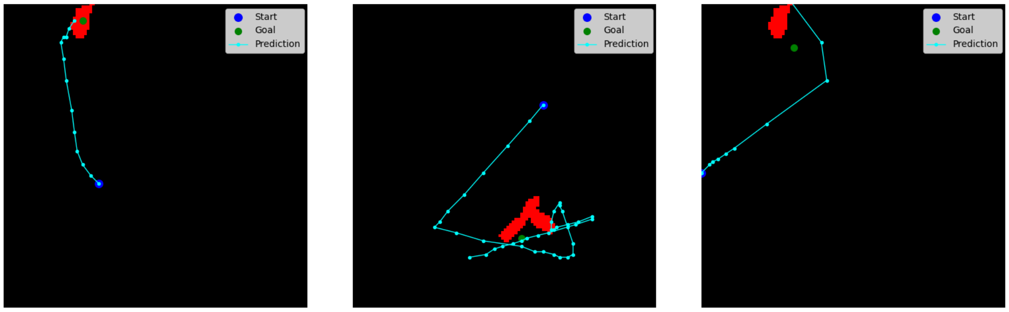
Figure 4: Six representative cases from the synthetic dataset. The top row displays **landmark-based** navigation samples, while the bottom row shows **target-based** navigation samples. The correct actions for those states are written at the bottom of each picture. Red rectangles on the images show the ground truth target locations. Red polygons on the maps are the landmarks and the arrows show the current locations and orientations of the drone.

## 6 Experiment Results

We evaluated our model with a history size of 16 and compared its performance against the baseline from the original CityNav paper. The comprehensive results are presented in Table 2. Our model demonstrates competitive performance across all splits. Notably, on the challenging Test Unseen split, our model achieves a Success Rate (SR) of 8.05, surpassing the baseline's 6.37. While the baseline model achieves a lower Navigation Error (NE), our model's higher success rate indicates it is more effective at reaching the target destination. Illustrative examples of various trajectory outcomes can be seen in Figure 5.

We further analyzed the impact of our synthetic data augmentation. Table 3 shows the action prediction accuracy of the 16 history model before and after augmentation with synthetic data. Each score is calculated on correspoing

The effect of synthetic data augmentation is further highlighted in Table 4. The augmented model shows a significant improvement in NE, reducing it from 137.4 to 108, while having a similar SR and OSR scores. This suggests that the synthetic data helps the model to learn a little more precise navigation paths.



(a) Success Trajectory (b) Oracle Success (c) Non-Success

Figure 5: Examples of predicted trajectories on the Test Unseen split. (a) A successful trajectory where the agent reaches the target. (b) An unsuccessful trajectory where the agent fails but the goal was reachable within the oracle path. (c) An unsuccessful trajectory where the goal was not reachable even by the oracle.

Table 2: Comparison with CityNav Baseline Model

| | **Easy** | | | **Medium** | | | **Hard** | | | **All** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **NE** | **SR** | **OSR** | **NE** | **SR** | **OSR** | **NE** | **SR** | **OSR** | **NE** | **SR** | **OSR** |
| **Val Seen** | | | | | | | | | | | | |
| CityNav Baseline | 64.7 | 8.73 | 49.40 | 55.7 | 9.67 | 40.15 | 58.7 | 7.72 | 17.54 | 59.75 | 8.69 | 35.59 |
| Our Model (16 Hist) | 96.4 | 14.25 | 27.65 | 137.92 | 7.0 | 16.68 | 136.73 | 6.42 | 13.56 | 123.5 | **9.25** | 19.3 |
| **Val Unseen** | | | | | | | | | | | | |
| CityNav Baseline | 80.0 | 5.95 | 35.96 | 73.1 | 5.14 | 23.25 | 73.3 | 6.38 | 11.38 | 75 | 5.83 | 22.27 |
| Our Model (16 Hist) | 113 | 10.64 | 25.0 | 166.8 | 5.19 | 10.8 | 188.16 | 3.6 | 10.15 | 160 | **6.11** | 14.46 |
| **Test Unseen** | | | | | | | | | | | | |
| CityNav Baseline | 98.9 | 6.15 | 39.89 | 90.9 | 6.29 | 21.47 | 90.0 | 6.80 | 12.10 | 93.83 | 6.37 | 26.16 |
| Our Model (16 Hist) | 111.83 | 12.06 | 25.8 | 147.45 | 5.83 | 13.94 | 162.15 | 4.86 | 10.06 | 137.4 | **8.05** | 17.57 |

Table 3: Action Prediction Accuracy on Synthetic Data

| | **16 History Model** | | **16 History Model + Augmentation** | |
|---|---|---|---|---|
| **Action** | **Landmark Based** | **Target Based** | **Landmark Based** | **Target Based** |
| **Forward** | 0.86 | 0.35 | 0.57 | 0.92 |
| **Turn Right** | 0.20 | 0.24 | 0.79 | 0.53 |
| **Turn Left** | 0.24 | 0.41 | 0.58 | 0.4 |

Table 4: Test Unseen Scores

| | **NE** ↓ | **SR** ↑ | **OSR** ↑ |
|---|---|---|---|
| **16 History Model** | 137.4 | 8.05 | 17.57 |
| **16 History + Augmentation** | 108 | 8.07 | 16.36 |

# 7 Conclusion

In this work, we fine-tuned the Qwen2.5-VL-3B model for aerial navigation on the CityNav benchmark, achieving a new state-of-the-art success rate on the Test Unseen split by addressing action imbalance and incorporating trajectory history. We observed significant overfitting, which we mitigated by generating a targeted synthetic dataset to explicitly teach map interpretation skills. Our results highlight the effectiveness of using small, open-source VLMs for embodied AI when paired with data-centric strategies, and suggest that skill-specific synthetic data is a promising direction for creating more robust navigation agents.

# References

Hengxing Cai, Jinhan Dong, Jingjun Tan, Jingcheng Deng, Sihang Li, Zhifeng Gao, Haidong Wang, Zicheng Su, Agachai Sumalee, and Renxin Zhong. Flightgpt: Towards generalizable and interpretable uav vision-and-language navigation with vision-language models. *arXiv preprint arXiv:2505.12835*, 2025.

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.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, et al. Openfly: A comprehensive platform for aerial vision-language navigation. *arXiv preprint arXiv:2502.18041*, 2025.

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.

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.

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*, pages 15384–15394, 2023.

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*, pages 621–635. Springer, 2018.

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.