ALIGNFLY: LANDMARK-ALIGNMENT REINFORCEMENT FINE-TUNING FOR UAV VISION-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

Abstract

Vision-language navigation for unmanned aerial vehicles (UAV-VLN) requires agents to follow natural language instructions using rapidly changing visual observations along three-dimensional trajectories. Such instructions usually describe routes via ordered visual landmarks, which provide stage-wise feedback for navigation. Existing UAV-VLN research rarely explores the correspondence between landmarks and navigation stages. In this paper, we propose AlignFly, a landmark-alignment reinforcement fine-tuning (RFT) method for optimizing navigation actions. Starting from a vision-language model (VLM) initialized through supervised fine-tuning (SFT) on expert demonstrations, AlignFly samples candidate action chunks from the same expert state and executes them independently in a simulator. A VLM-as-Judge is introduced to compare the visual evidence of the target landmark in post-action observations of candidates and construct rewards for group relative policy optimization (GRPO). To provide the required target-landmark information, we develop an automated annotation pipeline that extracts ordered landmark phrases from instructions, scores their visual evidence in expert trajectory observations, and monotonically aligns them with trajectory stages through dynamic programming. We apply this pipeline to a scene dataset of OpenFly and use the resulting annotations for RFT. Experiments on the dataset demonstrate consistent improvements on the seen split and a held-out scene, while ablations support the contribution of landmark-alignment feedback to UAV navigation. Code and annotated dataset will be released after review.

1 INTRODUCTION

Vision-language navigation (VLN) aims to enable embodied agents to navigate using natural language instructions and visual observations (Anderson et al., 2018). Extending this capability to unmanned aerial vehicles (UAVs) enables intuitive human-robot interaction in applications such as inspection, environmental monitoring, delivery, and search and rescue (Mohsan et al., 2022). Compared with ground-based VLN, UAV-VLN operates in large three-dimensional environments, involves rapid changes in viewpoint and landmark scale, and requires decisions based on visual information from a moving camera with a limited field of view. Recent platforms and benchmarks have continued to expand the scale, diversity, flight realism, and urban coverage of this task (Liu et al., 2023; Lee et al., 2025; Wang et al., 2025; Gao et al., 2026), providing an increasingly mature foundation for studying language-guided aerial agents.

Building on these resources, recent UAV-VLN methods have improved navigation decisions by combining vision-language models (VLMs) with hierarchical semantic planning, efficient history modeling, and spatial reasoning (Cai et al., 2025; Jiang et al., 2025; Zhang et al., 2025; Cai et al., 2026; Fan et al., 2026; Ning et al., 2026; Sun et al., 2026; Zhao et al., 2026). However, few studies have used landmark evidence encountered during navigation to optimize decisions. Natural language instructions typically interleave motion descriptions with a sequence of landmarks that provide visual references along the route (He et al., 2021; Liu et al., 2023). At each decision step, aligning visual observations with the corresponding instruction landmarks can help estimate navigation progress and identify the target for the current navigation stage. This correspondence grounds the current

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

navigation target in observable scene content, providing visual evidence for subsequent action decisions. Figure 1 illustrates how landmark alignment affects navigation decisions.

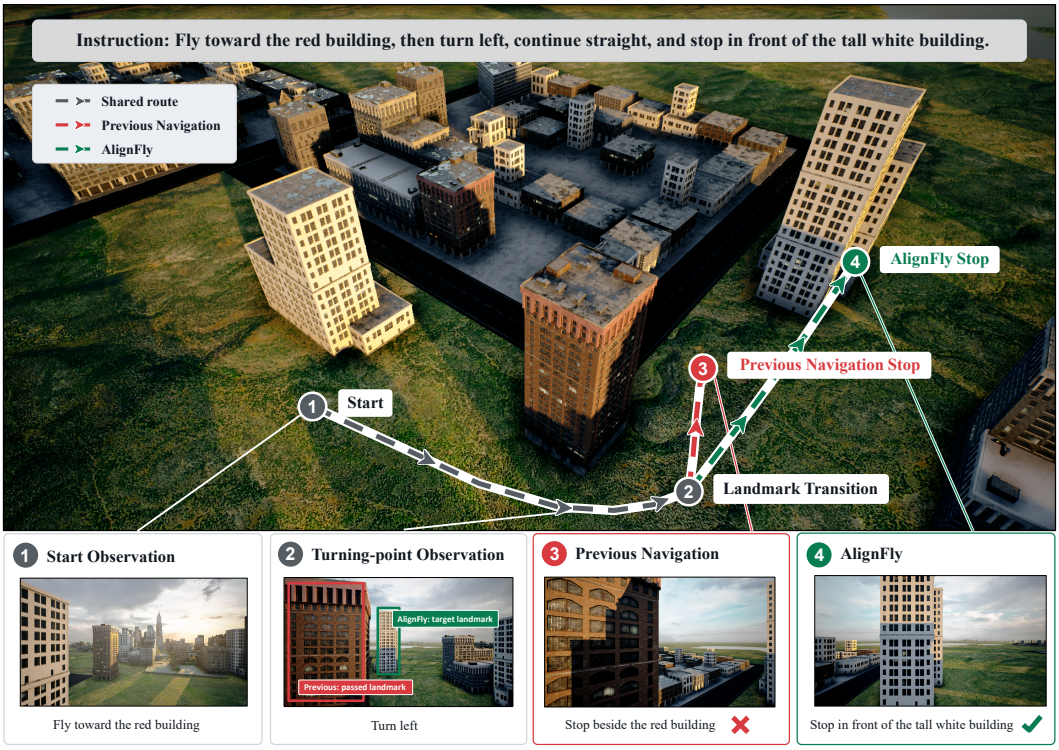


Figure 1: **Effect of landmark alignment on UAV navigation decisions.** Without dedicated landmark-alignment training, the previous navigation policy continues to base its action decisions on the red building after turning left, ultimately failing because of its persistent reliance on the same visual reference. In contrast, AlignFly, optimized through landmark alignment, aligns the observation after the left turn with the correct target landmark, namely the tall white building, and selects subsequent actions consistent with the intended route.

We propose **AlignFly**, a landmark-alignment RFT method for UAV-VLN. Its central idea is to convert the correspondence between instruction landmarks and expert trajectory stages into reward feedback on action outcomes. Specifically, starting from a VLM, we first use basic SFT to learn the correspondence among instructions, observations, and expert actions, obtaining an initial navigation policy, and then perform landmark-alignment RFT. We sample candidate action decisions from the same state and execute them independently in a simulated environment to obtain post-action visual observations. A VLM-as-Judge compares the visual evidence for the target landmark of the current navigation stage in these candidate observations, providing a landmark-alignment reward for group relative policy optimization (GRPO).

The landmark-alignment reward requires a target landmark to be assigned to each trajectory state. However, existing UAV-VLN datasets typically pair an instruction with a complete expert trajectory without aligning its landmarks with the corresponding trajectory stages. To address this gap, we propose an automated pipeline for aligning landmarks with trajectory stages. Specifically, we extract an ordered landmark list from each instruction. For each trajectory state, the VLM-as-judge assesses the visual evidence for different landmarks in the observation obtained after the expert action, and dynamic programming determines a unique target landmark using this evidence and the order of the landmark list. We apply this pipeline to data from one OpenFly (Gao et al., 2026) environment to obtain fine-grained target-landmark annotations and use the annotated dataset for RFT.

Our main contributions are as follows.

- We propose AlignFly, a landmark-alignment RFT method that constructs rewards from target-landmark evidence in observations obtained after different candidate actions from the same expert state, providing stage-specific signals for optimizing navigation actions.
- We design an automated annotation pipeline for aligning landmarks with trajectory stages. By combining ordered landmark extraction, visual evidence scoring, and monotonic alignment, the pipeline assigns target-landmark annotations to expert trajectory states and provides landmark-alignment information for RFT.
- We annotate landmark alignment in data from an OpenFly scene and use these annotations for RFT. Experiments show that AlignFly outperforms existing UAV-VLN methods on both Test-Seen and the held-out Test-Unseen split.

2 Related Work

2.1 Vision-Language Navigation for UAVs

VLN was initially introduced to study natural language instruction in indoor environments (Anderson et al., 2018; Fried et al., 2018; Krantz et al., 2020). Subsequent research extended the task to UAVs, revealing distinctive challenges such as three-dimensional motion and long outdoor trajectories (Fan et al., 2023; Liu et al., 2023). As UAV simulators and navigation benchmarks have developed, improvements in flight realism, data diversity, and trajectory scale have enabled a broader investigation of UAV navigation methods (Gao et al., 2026; Lee et al., 2025; Wang et al., 2025).

One line of research formulates UAV navigation as a planning problem, using foundation models to interpret language and aerial observations, generate subgoals or waypoints, and coordinate memory and control modules (Zhang et al., 2025; Ning et al., 2026; Li et al., 2025; Fan et al., 2026). Another line builds end-to-end policies directly on VLMs, mapping language and visual observations to actions or short action sequences (Cai et al., 2025; Jiang et al., 2025; Cai et al., 2026; Sun et al., 2026; Zhao et al., 2026). By jointly learning perception, reasoning, and action, these methods reduce the reliance on manually designed interfaces and allow task feedback to optimize the entire policy. AlignFly follows this paradigm, focusing on reward feedback to optimize navigation decisions.

2.2 Reinforcement Fine-Tuning for Navigation

Reinforcement fine-tuning (RFT) uses task feedback to refine the decisions of large models beyond what can be learned from supervised demonstrations (Shao et al., 2024). In navigation, policy learning with task rewards can improve exploration, recovery behavior, spatial decisions, and generalization beyond expert trajectories (Wang et al., 2019; Zhu et al., 2020). Recent ground-based VLN studies have applied GRPO to RFT of VLMs and graph-based navigation policies (Qi et al., 2025; Ye et al., 2025; Li et al., 2026). UAV-VLN methods have also begun to adopt RFT, first establishing instruction-following and action-generation capabilities through SFT or imitation learning, and then using GRPO to optimize rewards associated with progress, reasoning, and output validity (Cai et al., 2025; 2026; Zhou et al., 2026; Zhao et al., 2026). AlignFly also uses GRPO as its optimization algorithm, but is distinguished by stage-specific semantic feedback: candidate actions are executed from the same expert state, and the landmark assigned to that trajectory stage enables group-level comparisons of landmark evidence across the resulting observations.

2.3 Alignment between Language and Vision for Navigation

The correspondence between visual observations and natural language instructions has long been a central problem in navigation research. The Self-Monitoring Agent incorporates visual–textual co-grounding into a progress estimation system to improve navigation decisions (Ma et al., 2019). Landmark-RxR constructs manually annotated sub-instruction and sub-trajectory pairs in a VLN dataset (He et al., 2021). GELA learns these correspondences through pretraining objectives for entity phrase prediction and landmark bounding-box prediction (Cui et al., 2023). In aerial vision-and-dialog navigation, FELA combines semantic grid representations with a visual grounding module to explicitly learn the correspondence between dialogue and overhead imagery (Su et al., 2025). In contrast to previous work, we construct an automated annotation pipeline for aligning landmarks with trajectory stages and convert the resulting alignment into rewards for RFT. This design directly

uses stage-specific language–vision correspondences to optimize actions and improve navigation decisions.

3 METHOD

3.1 TASK DEFINITION AND METHOD OVERVIEW

UAV-VLN requires a policy to navigate under partial observability to a destination described by a natural language instruction $\mathcal{I}$. At the decision state t, the policy receives the input

$$x_t = (\mathcal{I},\, p_t,\, \mathbf{a}_{<t},\, I_t,\, \mathcal{K}_t)\,, \tag{1}$$

where $p_t = (p_t^x, p_t^y, p_t^z, p_t^{yaw})$ is a four-dimensional coordinate vector containing the UAV's current position and yaw angle, $\mathbf{a}_{<t}$ is the sequence of executed actions, I_t is the current observation, and $\mathcal{K}_t$ contains selected historical observations; the selection rule is given in Appendix A. The pose provides spatial position and orientation information, while the action history records the motion that produced the current observation. The current observation describes the visible surroundings and landmarks, and historical images provide temporal context for understanding route progress.

The policy π_θ predicts an action chunk

$$\mathbf{a}_t = (a_t^1, \ldots, a_t^{L_t}), \qquad a_t^k \in \mathcal{A}, \quad 1 \le L_t \le L_{\max}, \tag{2}$$

where $L_{\max}$ is the maximum number of primitive actions in a chunk. The discrete primitive action space is

$$\begin{aligned}\mathcal{A} = \{&\texttt{A} : \text{Forward}(3\text{ units}), \quad \texttt{B} : \text{TurnLeft}(30^\circ), \quad \texttt{C} : \text{TurnRight}(30^\circ),\\ &\texttt{D} : \text{Up}(2\text{ units}), \quad \texttt{E} : \text{Down}(2\text{ units}), \quad \texttt{F} : \text{Forward}(6\text{ units}),\\ &\texttt{G} : \text{Forward}(9\text{ units}), \quad \texttt{H} : \text{Stop}\}.\end{aligned}$$

The UAV executes the predicted action chunk and returns the updated state x_{t+1}, after which the next decision begins. This interaction repeats until the policy predicts a stop action or reaches the maximum number of predictions.

3.2 ANNOTATING LANDMARK–TRAJECTORY STAGE ALIGNMENT

To provide the supervision required by the landmark-alignment reward in RFT, we design an automated annotation pipeline for aligning landmarks with trajectory stages, as shown in Figure 2(a). We construct this supervision from instructions and trajectory observations using only expert trajectories in an existing dataset.

Extracting instruction landmarks. We use a language model P and the prompt in Appendix B.3 to extract an ordered list of landmark phrases $\mathcal{L} = (\ell_1, \ldots, \ell_M)$ from instruction $\mathcal{I}$. The landmark phrases retain visually distinctive attributes and exclude motion descriptions without visual referents. The landmark order provides a temporal constraint for alignment.

Obtaining landmark evidence from expert trajectory observations. We index expert trajectory states by $t = 0, \ldots, T-1$. At each state, $\mathbf{a}_t^E$ denotes the expert action chunk, and I_{t+1}^E denotes the observation obtained after executing that action. For each observation–landmark pair, we use a VLM-as-Judge J with the prompt in Appendix B.2 to determine whether the image contains evidence of the landmark. We score this evidence using the log-probability difference between affirmative and negative responses:

$$q_J(I, \ell_m) = \log P_J(\texttt{yes} \mid I, \ell_m) - \log P_J(\texttt{no} \mid I, \ell_m). \tag{3}$$

Computing $q_J(I_{t+1}^E, \ell_m)$ for the observation obtained after the expert action at each state and the extracted landmark list yields the evidence matrix used for alignment.

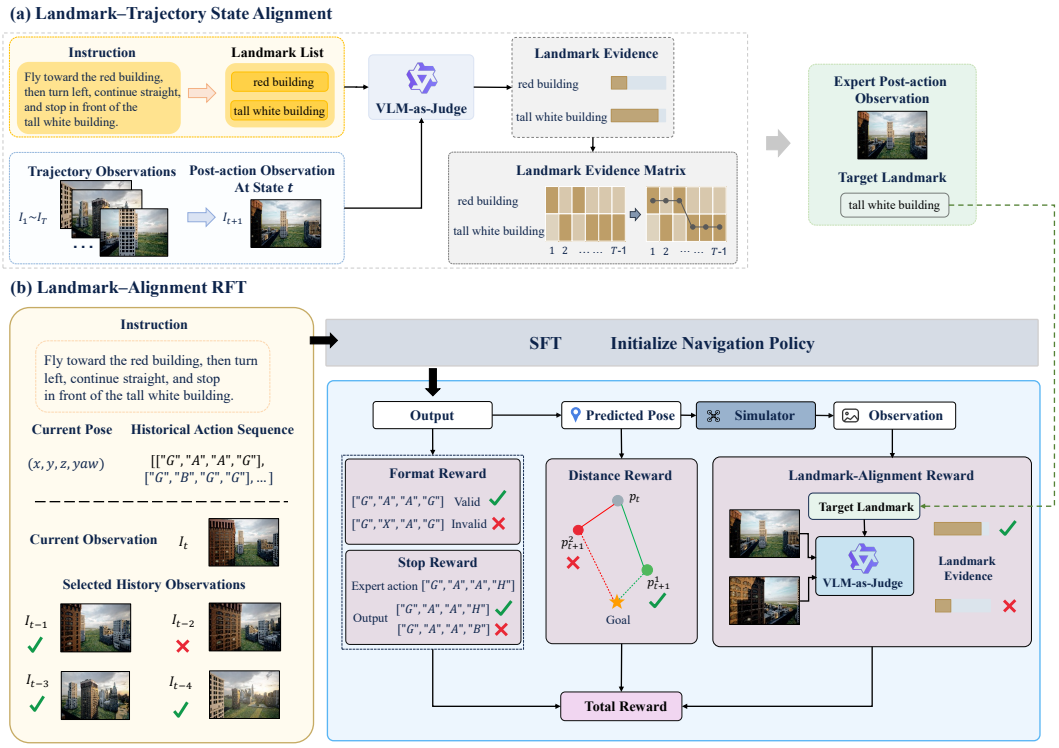


Figure 2: Method Overview. (a) Landmark–Trajectory Stage Alignment. Extract landmark descriptions from natural language instructions, obtain landmark evidence within expert trajectories, and perform monotonic alignment. (b) Landmark-Alignment RFT. First, initialize the navigation policy via SFT. Then, the target landmark evidence in observations after executing candidate actions under expert states is used to construct rewards. Combined with format reward, stop reward and distance reward, it provides action optimization signals relevant to the current stage for navigation decision-making.

Monotonic alignment of landmarks and trajectory stages. Let $z_t \in \{1, \ldots, M\}$ denote the target landmark assigned to expert state t. Subject to the order in which landmarks appear in the instruction, we select the assignment that maximizes cumulative evidence:

$$\mathbf{z}^* = \arg \max_{\mathbf{z} \in \mathcal{Z}_{\text{mono}}} \sum_{t=0}^{T-1} q_J(I_{t+1}^E, \ell_{z_t}), \qquad \mathcal{Z}_{\text{mono}} = \{\mathbf{z} \mid 1 \le z_0 \le \cdots \le z_{T-1} \le M\}. \tag{4}$$

This objective can be solved by dynamic programming, with the recurrence $D(t, m) = q_J(I_{t+1}^E, \ell_m) + \max_{j \le m} D(t-1, j)$, initialized by $D(0, m) = q_J(I_1^E, \ell_m)$. Backtracking from the optimal entry at the final step yields $\mathbf{z}^*$. Consecutive states assigned to the same target landmark form a trajectory stage, with $\ell_t^{\text{tar}} = \ell_{z_t^*}$ as its target landmark. The constraint permits a landmark to span multiple states and allows some landmarks to be skipped.

Human validation results and data distributions for this annotation pipeline are presented in Appendix D.

3.3 Landmark-Alignment RFT

AlignFly uses a VLM as its backbone. Following prior work (Cai et al., 2025; 2026), we first use SFT to learn a basic navigation policy, and then perform landmark-alignment RFT on this basis. The overall framework is shown in Figure 2(b).

Initializing the navigation policy through SFT. We initialize π_θ by learning from expert context–action pairs. SFT learns the basic correspondence among instructions, observations, and navigation

actions, together with the output format. It thus provides a policy with initial navigation competence from which RFT can sample meaningful candidate actions. The training prompt is provided in Appendix B.1.

Sampling candidate action chunks from the same expert state. At each RFT update, we select an expert trajectory state t and sample G candidate responses from the policy snapshot $\pi_{\theta_{\text{old}}}$:

$$\mathbf{a}_t^{(g)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x_t^E), \qquad g = 1, \ldots, G. \tag{5}$$

All candidates share the input context, expert trajectory state, and target landmark.

Executing candidate action chunks. For each valid candidate action chunk, we first restore expert trajectory state t in the simulator, then execute the chunk and return the resulting image $I_{t+1}^{(g)}$ and pose $p_{t+1}^{(g)}$. Resetting the state before each candidate is executed allows their outcomes to be compared from the same starting point. Each simulation rollout terminates after one action chunk, and its outcome is used to compute reward feedback.

Computing the landmark-alignment reward. The annotated target landmark ℓ_t^{tar} specifies the semantic evidence to attend to at the current state. We use the same VLM-as-Judge and evidence computation as in data annotation. To facilitate evidence comparison for candidate g, we subtract the evidence score of the expert post-action observation I_{t+1}^E from that of the candidate observation $I_{t+1}^{(g)}$, then apply a sigmoid function to obtain the landmark-alignment reward:

$$r_{\text{lmk}}^{(g)} = \sigma\left(\frac{q_J(I_{t+1}^{(g)}, \ell_t^{\text{tar}}) - q_J(I_{t+1}^E, \ell_t^{\text{tar}})}{\tau_\ell}\right), \qquad \sigma(u) = \frac{1}{1+\exp(-u)}, \tag{6}$$

Here $\tau_\ell > 0$ controls the reward's sensitivity to differences in evidence. This reward evaluates the semantic consequences of actions: candidate action chunks receive stronger positive feedback when their outcomes strengthen evidence for the target landmark in the observation.

Other rewards. To improve output validity and navigation accuracy, we introduce three additional rewards that complement the landmark-alignment reward:

- **Format reward.** This reward encourages outputs to comply with the discrete action interface. Let $\mathcal{V}$ denote the set of outputs that (i) can be parsed as sequences containing only actions in $\mathcal{A}$, (ii) have length $1 \le L_t^{(g)} \le L_{\max}$, and (iii) end with a stop action whenever $L_t^{(g)} < L_{\max}$, with the stop action appearing only in the final position. We define

$$v_{\text{fmt}}^{(g)} = \mathbf{1}\left[\mathbf{a}_t^{(g)} \in \mathcal{V}\right], \qquad r_{\text{fmt}}^{(g)} = c_{\text{fmt}} v_{\text{fmt}}^{(g)}. \tag{7}$$

  If $v_{\text{fmt}}^{(g)} = 0$, the output does not satisfy the format requirements.
- **Distance reward.** This reward encourages actions that reduce the distance between the UAV and the goal. Let $\bar{p}_t = (p_t^x, p_t^y, p_t^z)$ and $\bar{p}_{t+1}^{(g)}$ denote the pre- and post-action positions, respectively, and let $\bar{p}_{\text{goal}}$ denote the goal position. We compute $d_t = \|\bar{p}_t - \bar{p}_{\text{goal}}\|_2$ and $d_{t+1}^{(g)} = \|\bar{p}_{t+1}^{(g)} - \bar{p}_{\text{goal}}\|_2$, and define the distance reward as

$$r_{\text{dist}}^{(g)} = \text{clip}\left(\frac{d_t - d_{t+1}^{(g)}}{D_{\text{ref}}}, 0, 1\right). \tag{8}$$

  where D_{ref} is a fixed distance scale.
- **Stop reward.** This reward encourages stop-or-continue decisions to agree with the expert. Let $h_t^{(g)} = \mathbf{1}[\texttt{H} \in \mathbf{a}_t^{(g)}]$ and $h_t^E = \mathbf{1}[\texttt{H} \in \mathbf{a}_t^E]$ indicate whether the predicted and expert action chunks contain a stop action, respectively. We define

$$r_{\text{stop}}^{(g)} = \begin{cases} c_{\text{stop}}^+, & h_t^{(g)} = 1 \ \wedge \ h_t^E = 1, \\ c_{\text{stop}}^-, & h_t^{(g)} = 0 \ \wedge \ h_t^E = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

  where $c_{\text{stop}}^+ > c_{\text{stop}}^- \ge 0$.

Combining the landmark-alignment reward with the format, distance, and stop rewards gives the composite reward for candidate g:

$$r^{(g)} = r^{(g)}_{\text{fmt}} + v^{(g)}_{\text{fmt}} \left(\lambda_\ell r^{(g)}_{\text{lmk}} + \lambda_d r^{(g)}_{\text{dist}} + \lambda_s r^{(g)}_{\text{stop}} \right). \tag{10}$$

The validity gate $v^{(g)}_{\text{fmt}}$ ensures that malformed outputs receive zero reward and bypass the remaining reward computations. Appendix C describes the hyperparameter choices used in the method.

4 Experiments

4.1 Experimental Setup

Dataset. OpenFly is a large-scale UAV-VLN benchmark containing diverse trajectories from multiple scenes (Gao et al., 2026). We annotate landmark–trajectory stage alignment in one scene containing 9,731 trajectories and train AlignFly on this basis. Of these trajectories, 9,500 provide approximately 44,267 training examples at individual decision steps, and the remaining 231 form the *seen test split (Test-Seen)*. We also sample 231 trajectories from another scene in the benchmark to form the *unseen test split (Test-Unseen)*. These splits evaluate navigation in seen and unseen scenes, respectively.

Evaluation metrics. Following the standard evaluation protocol, we report four basic metrics: Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OSR), and Success weighted by Path Length (SPL). In addition, using the landmark evidence computation, we measure the mean landmark evidence (LE) in the visual observations obtained after executing model-predicted actions from each expert state. We treat LE as a diagnostic measure of reward-aligned visual grounding rather than an independent evaluation metric. All evaluation metrics are averaged over three runs with different random seeds.

Implementation details. We use Qwen2.5-VL-7B as the navigation backbone (Bai et al., 2025). During SFT, we apply LoRA with rank 16 to the language backbone (Hu et al., 2022), using a batch size of 4, an initial learning rate of 1.0×10^{-4}, and two training epochs. During RFT, GRPO samples five candidate action chunks per expert state, with a batch size of 16, an initial learning rate of 1.0×10^{-6}, and one training epoch. For the auxiliary models, we query `gpt-4o-mini` through the OpenAI API to extract landmarks from instructions and use Qwen2.5-VL-32B as the VLM-as-Judge to compute landmark evidence. Training uses four NVIDIA H100 GPUs, each with 80 GB of memory. We implement SFT with LLaMA-Factory (Zheng et al., 2024) and RFT with veRL (Sheng et al., 2025). Candidate actions are executed in the OpenFly simulator, which we deploy as FastAPI services on a separate server equipped with four NVIDIA RTX 4090 GPUs. During RFT, the training server sends candidate action chunks to these services and receives the observations and position coordinates required to compute rewards.

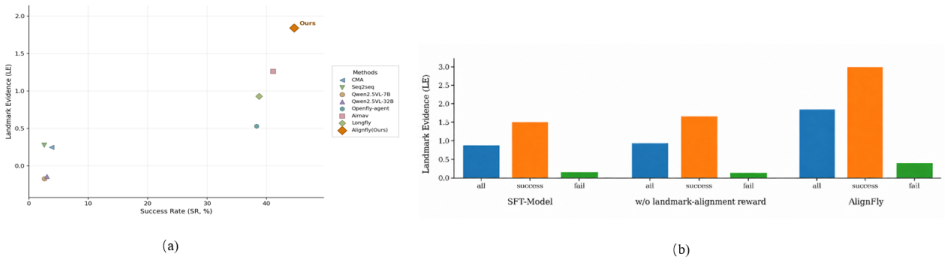
4.2 Comparative Results

Table 1 reports the navigation results of different models on Test-Seen and Test-Unseen. Detailed settings of the baselines are provided in Appendix E. The heuristic method, conventional methods, and pretrained VLMs without adaptation all achieve low success rates. Increasing Qwen2.5-VL from 7B to 32B yields only limited gains, indicating that increasing model size alone is not the most effective approach to improving navigation performance. In contrast, end-to-end UAV-VLN agents trained for navigation perform substantially better. AlignFly compares semantic evidence from candidate actions through its landmark-alignment reward to improve local action decisions, increasing navigation success over the strongest baseline by 3.51 and 4.59 percentage points on Test-Seen and Test-Unseen, respectively.

In addition, Figure 3(a) presents the success rate and mean landmark evidence (LE) of each model on Test-Seen. LE is computed by pooling states across all trajectories and averaging the landmark evidence score q_J in equation 3. The figure shows an overall trend in which stronger landmark evidence accompanies higher success rates, with AlignFly achieving the highest values on both metrics. Compared with other navigation-trained models, AlignFly improves navigation success

Table 1: Navigation performance on the OpenFly test splits. NE is measured in meters; SR, OSR, and SPL are percentages. **Bold** and underlined values denote the best and second-best results, respectively.

Method	Test-Seen				Test-Unseen			
	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑
Random Action	106.23	4.65	43.38	3.21	137.68	3.89	31.47	2.84
CMA (Krantz et al., 2020)	108.24	3.83	19.17	2.62	142.93	2.14	16.55	1.31
Seq2Seq (Krantz et al., 2020)	102.33	2.58	26.51	1.94	146.75	2.73	15.39	1.64
Qwen2.5-VL-7B (Bai et al., 2025)	132.95	2.63	21.42	1.26	162.30	2.51	18.72	1.48
Qwen2.5-VL-32B (Bai et al., 2025)	127.94	3.02	23.18	1.47	158.86	3.26	20.41	1.76
OpenFly-Agent (Gao et al., 2026)	113.88	38.31	69.32	35.78	124.59	23.17	53.24	16.43
AirVLN-R1 (Cai et al., 2026)	89.57	41.13	68.62	**37.85**	110.51	28.26	56.42	18.75
LongFly (Jiang et al., 2025)	96.53	38.76	65.29	34.86	116.92	23.94	55.18	20.62
AlignFly (Ours)	**84.74**	**44.64**	**71.97**	37.81	**104.06**	**32.85**	**61.27**	**25.40**



(a) (b)

Figure 3: **Navigation success rate and landmark evidence on Test-Seen.** (a) Success rate (SR) and mean landmark evidence (LE) for different navigation models. (b) Mean LE for SFT only, w/o landmark-alignment reward, and AlignFly over all test tasks (all), tasks with three successful runs (success), and tasks with three failed runs (fail).

while obtaining visual observations along its executed trajectories that provide stronger evidence for instruction landmarks. This observation is consistent with the motivation for using landmark alignment to optimize action outcomes.

4.3 Ablation Study

We conduct ablations on Test-Seen to examine the role of the landmark-alignment reward in RFT and the importance of the initial navigation policy. The unadapted Qwen2.5-VL-7B model uses neither SFT nor RFT. *SFT only* omits RFT, whereas *RFT only* applies GRPO directly to the pretrained backbone. Starting from the complete AlignFly objective, we then remove the distance, format, stop, or landmark-alignment reward while keeping the remaining configuration unchanged.

Table 2: Ablation results on the OpenFly Test-Seen and Test-Unseen splits. Metrics follow Table 1.

Variant	Test-Seen				Test-Unseen			
	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑
Qwen2.5-VL-7B (w/o SFT and RFT)	132.95	2.63	21.42	1.26	162.30	2.51	18.72	1.48
SFT only	92.15	38.69	66.49	31.22	122.37	22.86	54.25	16.29
RFT only	116.53	9.75	43.51	5.64	141.62	7.19	33.86	3.47
w/o distance reward	88.74	40.23	68.84	34.05	114.94	26.52	58.61	21.95
w/o format reward	85.21	42.04	70.63	36.48	108.56	30.43	59.18	24.31
w/o stop reward	89.29	41.46	69.85	33.43	110.47	29.71	59.77	23.52
w/o landmark-alignment reward	89.62	39.94	68.36	32.27	116.81	24.94	57.53	18.68
AlignFly (Ours)	**84.74**	**44.64**	**71.97**	**37.81**	**104.06**	**32.85**	**61.27**	**25.40**

As shown in Table 2, initializing the navigation policy through SFT before RFT substantially outperforms direct RFT in both seen and unseen environments, and landmark-alignment RFT further improves navigation performance on this basis. Without the landmark-alignment reward, RFT yields only limited gains over SFT in both environments. Among the reward-removal experiments, omitting the landmark-alignment reward has the largest effect on success rate and final navigation error, with a larger absolute performance loss in the unseen environment. Removing any auxiliary reward also worsens all four metrics: the distance reward complements semantic feedback by encouraging progress toward the goal, while the stop and format rewards constrain termination decisions and output executability, respectively. The overall pattern is consistent with these signals addressing different requirements of navigation, although their effects may interact during optimization and should not be interpreted as independent, additive contributions.

Figure 3(b) compares mean LE for SFT only, the RFT model without the landmark-alignment reward, and AlignFly over all Test-Seen tasks, tasks with three successful runs, and tasks with three failed runs. Without the landmark-alignment reward, RFT produces only a small increase in mean LE over SFT across all states, whereas AlignFly achieves a more pronounced improvement. For each model, mean LE is substantially higher on tasks that succeed in all three runs than on tasks that fail in all three runs, indicating that stronger landmark evidence accompanies stronger navigation performance. AlignFly also achieves higher LE than the other two variants in both outcome categories, although its evidence on tasks with three failed runs remains considerably weaker than on tasks with three successful runs. Together with the success-rate ablations, the overall increase in LE supports the role of landmark alignment in promoting visually grounded navigation decisions.

4.4 Additional Experiment

To examine whether the benefits of our RFT procedure depend on our own SFT initialization, we additionally use OpenFly-Agent as an alternative navigation model initialized through SFT. We apply our landmark-alignment RFT procedure while retaining its original model architecture, input formulation, and history-processing mechanism, and evaluate the resulting model in seen and unseen scenes.

Table 3: Additional results of applying landmark-alignment RFT to OpenFly-Agent. Metrics follow Table 1.

	Test-Seen				Test-Unseen			
Model	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑
OpenFly-Agent	113.88	38.31	69.32	35.78	124.59	23.17	53.24	16.43
OpenFly-Agent + RFT	**92.16**	**44.27**	**72.51**	**36.94**	**111.82**	**30.49**	**60.35**	**22.62**

Table 3 shows that landmark-alignment RFT improves all four metrics on both test splits, increasing SR by 5.96 and 7.32 percentage points in seen and unseen scenes, respectively. The reduction in NE and concurrent improvements in OSR and SPL indicate that the training procedure improves proximity to the goal and successful-path efficiency alongside task completion. These results support the conclusion that the benefits of our RFT procedure are not restricted to our own SFT policy. They extend to another navigation policy initialized through supervised learning, providing additional evidence for the applicability of the training procedure.

5 Conclusion

We presented AlignFly, a landmark-alignment RFT method. This method leverages landmark evidence to evaluate the performance of candidate actions and optimize navigation policies for UAVs. To provide this supervision, we developed an automated annotation pipeline that extracts ordered instruction landmarks, scores their visual evidence, and monotonically aligns them with expert trajectory stages. Experimental results demonstrate improved navigation performance on both test splits, including the held-out scene, supporting the effectiveness of landmark-alignment feedback for UAV navigation.

AI USE STATEMENT

Generative AI tools were utilized in this work for manuscript writing, language editing, and dataset annotation. As components of the research method, as described in Sections 3.2–3.3 and Section 4.1, `gpt-4o-mini` is adopted to extract instruction landmarks, and Qwen2.5-VL-32B is used to score landmark evidence for annotation and reward computation. The authors bear full responsibility for the research design, experimental results, and final manuscript content, including the verification of AI-assisted text.

REPRODUCIBILITY STATEMENT

To ensure research reproducibility, implementation details of AlignFly are provided in Section 4.1. To facilitate academic exchange and technical transparency, we will publicly release our source code, annotated dataset, and fine-tuned model weights in accordance with relevant licenses. This allows other researchers to replicate our findings and further build upon our work.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 3674–3683. Computer Vision Foundation / IEEE Computer Society, 2018.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL `https://arxiv.org/abs/2502.13923`.

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. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6659–6676, 2025.

Hengxing Cai, Yijie Rao, Ligang Huang, Zanyang Zhong, Jinhan Dong, Jingjun Tan, Changhao Nai, Jue Hou, Wenhao Lu, and Renxin Zhong. Airnav: A large-scale uav vision-and-language navigation dataset with natural and diverse instructions, 2026. URL `https://arxiv.org/abs/2601.03707`.

Yibo Cui, Liang Xie, Yakun Zhang, Meishan Zhang, Ye Yan, and Erwei Yin. Grounded entity-landmark adaptive pre-training for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12043–12053, 2023.

Chengjie Fan, Cong Pan, Zijian Liu, Ningzhong Liu, and Jie Qin. Htnav: A hybrid navigation framework with tiered structure for urban aerial vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10976–10985, 2026.

Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Wang. Aerial vision-and-dialog navigation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 3043–3061, 2023.

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. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 3318–3329, 2018.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Li Zhen, Pengan CHEN, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Yangpenghui, Yiwen Tang, Yuhang Tang, Shuai Liang, Songyi Zhu, Ziqin Xiong, Yifei Su, Xinyi Ye, Jianan Li, Yan Ding, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Openfly: A COMPREHENSIVE PLATFORM FOR AERIAL VISION-LANGUAGE NAVIGATION. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OKm3w71ymP.

Keji He, Yan Huang, Qi Wu, Jianhua Yang, Dong An, Shuanglin Sima, and Liang Wang. Landmark-rxr: Solving vision-and-language navigation with fine-grained alignment supervision. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 652–663, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Wen Jiang, Li Wang, Kangyao Huang, Wei Fan, Jinyuan Liu, Shaoyu Liu, Hongwei Duan, Bin Xu, and Xiangyang Ji. LongFly: Long-horizon UAV vision-and-language navigation with spatiotemporal context integration, 2025. URL https://arxiv.org/abs/2512.22010.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII*, volume 12373 of *Lecture Notes in Computer Science*, pp. 104–120. Springer, 2020.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. Citynav: A large-scale dataset for real-world aerial navigation. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 5912–5922. IEEE, 2025.

Jiangyang Li, Cong Wan, SongLin Dong, Chenhao Ding, Qiang Wang, Zhiheng Ma, and Yihong Gong. Trajectory-diversity-driven robust vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 9143–9154, 2026.

Tianshun Li, Tianyi Huai, Zhen Li, Yichun Gao, Haoang Li, and Xinhu Zheng. SkyVLN: Vision-and-language navigation and NMPC control for UAVs in urban environments. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 17199–17206. IEEE, 2025.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 15338–15348. IEEE, 2023.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Syed Agha Hassnain Mohsan, Muhammad Asghar Khan, Fazal Noor, Insaf Ullah, and Mohammed H. Alsharif. Towards the unmanned aerial vehicles (UAVs): A comprehensive review. *Drones*, 6(6):147, 2022.

Yuwei Ning, Ganlong Zhao, Yipeng Qin, Si Liu, Yang Liu, Liang Lin, and Guanbin Li. Lookaside-VLN: Direction-aware aerial vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32441–32450, 2026.

Zhangyang Qi, Zhixiong Zhang, Yizhou Yu, Jiaqi Wang, and Hengshuang Zhao. VLN-R1: Vision-language navigation via reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2506.17221.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pp. 1279–1297. ACM, 2025.

Yifei Su, Dong An, Kehan Chen, Weichen Yu, Baiyang Ning, Yonggen Ling, Yan Huang, and Liang Wang. Learning fine-grained alignment for aerial vision-dialog navigation. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 7060–7068. AAAI Press, 2025.

Xiaolou Sun, Wufei Si, Wenhui Ni, Yuntian Li, Dongming Wu, Fei Xie, Runwei Guan, He-Yang Xu, Henghui Ding, Yuan Wu, Yutao Yue, Yongming Huang, and Hui Xiong. Autofly: Vision-language-action model for UAV autonomous navigation in the wild. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=88RKxlFUNY.

Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Wenjun Wu, Hongsheng Li, Yue Liao, and Si Liu. Towards realistic UAV vision-language navigation: Platform, benchmark, and methodology. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6629–6638. Computer Vision Foundation / IEEE, 2019.

Shuhao Ye, Sitong Mao, Yuxiang Cui, Xuan Yu, Shichao Zhai, Wen Chen, Shunbo Zhou, Rong Xiong, and Yue Wang. Etp-r1: Evolving topological planning with reinforcement fine-tuning for vision-language navigation in continuous environments, 2025. URL https://arxiv.org/abs/2512.20940.

Weichen Zhang, Chen Gao, Shiquan Yu, Ruiying Peng, Baining Zhao, Qian Zhang, Jinqiang Cui, Xinlei Chen, and Yong Li. CityNavAgent: Aerial vision-and-language navigation with hierarchical semantic planning and global memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 31292–31309, 2025.

Baining Zhao, Jiacheng Xu, Weicheng Feng, Xin Zhang, Zhaolu Wang, Haoyang Wang, Shilong Ji, Ziyou Wang, Jianjie Fang, Zhiheng Zheng, Weichen Zhang, Yu Shang, Wei Wu, Chen Gao, Xinlei Chen, and Yong Li. Worldvln: Autoregressive world action model for aerial vision-language navigation, 2026. URL https://arxiv.org/abs/2605.15964.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 400–410. Association for Computational Linguistics, 2024.

Jiaxu Zhou, Shaobo Wang, Zhiyuan Yang, Zhenjun Yu, and Tao Li. Freefly-thinking : Aligning chain-of-thought reasoning with continuous uav navigation, 2026. URL https://arxiv.org/abs/2603.07181.

Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. Babywalk: Going farther in vision-and-language navigation by taking baby steps. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 2539–2556. Association for Computational Linguistics, 2020.

A HISTORICAL IMAGE SELECTION

The complete image history $\mathcal{H}_t^I = (I_0, \ldots, I_{t-1})$ contains substantial redundancy. We therefore select at most B_{hist} historical images using cumulative Fibonacci lags. Let $F_1 = F_2 = 1$ and $F_j = F_{j-1} + F_{j-2}$ for $j \geq 3$, and define the cumulative lag $d_j = \sum_{n=1}^{j} F_n$. The selected history is

$$\mathcal{K}_t = \left\{ I_{t-d_j} \mid 1 \leq j \leq B_{\text{hist}},\ d_j \leq t \right\}. \tag{11}$$

We insert the selected frames into the prompt in chronological order and use the same deterministic selection procedure during SFT, RFT, and inference.

In practice, we set $B_{\text{hist}} = 3$ during training, with accumulation intervals of 1, 2, and 4 decision states. Historical frames that do not exist at the beginning of a trajectory are omitted directly.

B PROMPT TEMPLATES

This section collects the prompt templates used by the navigation policy, the landmark evidence judge, and the instruction landmark extractor.

B.1 NAVIGATION POLICY PROMPT TEMPLATE

During SFT, RFT, and inference, the navigation policy uses a two-message multimodal prompt to map the navigation context to the next action chunk. The system message specifies the task, action space, output constraints, and serialization format. In the user message, $<$ historical_image_blocks $>$ is replaced by zero to three historical image–caption pairs in chronological order. The remaining placeholders are populated with the current front-view image, navigation instruction, UAV pose, and previously executed actions. During training, these inputs come from the expert anchor; during inference, they come from actual policy execution. Stage-specific landmarks and expert future images are not included in this prompt.

```
SYSTEM MESSAGE

## Role
You are a UAV navigation assistant in a 3D urban environment.

## Task
Given multimodal inputs, predict the next UAV actions to follow a natural language
    instruction.

## Inputs
1. Navigation instruction
2. Current UAV state: position (x, y, z) and heading
3. Up to 3 historical front-view images (optional)
4. Current front-view image
5. Previously executed actions (A-H)

## Objective
Generate a valid action sequence that moves the UAV toward the goal described in the
    instruction.

## Action Space
A: move forward 3m
B: rotate left 30°
C: rotate right 30°
D: move up 2m
E: move down 2m
F: move forward 6m
G: move forward 9m
H: stop

## Constraints
- Output at most 4 actions
- If the goal is reached early, end with 'H'
- Output only valid actions
- 'H' must only appear at the end

## Output Format
Return only a JSON list of actions, for example:
["G", "A", "H"]
```

```
USER MESSAGE

<historical_image_blocks>
<image>
(Above is the current front-view image)

## Navigation Instruction
<instruction>

## UAV State
position (x,y,z,heading): [<x>, <y>, <z>, <heading>]

## Previous Actions
actions: <previous_actions_json>

## Task Instance
Predict the next UAV action sequence based on the instruction, state, and images.

## Output
Strictly follow the system constraints. Return a JSON list only.

ASSISTANT MESSAGE (SFT TARGET)

<expert_action_sequence_json>
```

Each historical image block in $<$ historical_image_blocks $>$ has the following form, where k denotes its chronological index:

```
<image>
(Above is historical front-view image <k>)
```

B.2 VLM-AS-JUDGE PROMPT FOR LANDMARK EVIDENCE

For each image–landmark pair, we provide the image as visual input and instantiate the following text prompt by replacing <target_landmark>. The prompt restricts the response to one of two words so that the conditional probabilities of `yes` and `no` can be used in Eq. equation 3. The same template is used for landmark–trajectory alignment and RFT reward computation.

```
Determine whether the UAV first-person image contains the target landmark.
Answer with exactly one word: yes or no.

Target landmark: <target_landmark>

Question: Is this target landmark clearly present in the image or highly relevant to it?
```

B.3 PROMPT FOR EXTRACTING LANDMARKS FROM INSTRUCTIONS

We use the following two-message text prompt to obtain the ordered landmark list required for alignment. The placeholder $<$ instruction $>$ is replaced by the complete route instruction. Requiring JSON-only output enables direct parsing, while the rules preserve visually distinctive attributes and remove motion descriptions.

```
SYSTEM MESSAGE

You are an information extraction model.
Your task is to extract visual landmarks from UAV navigation instructions.
Do not explain, reason step by step, or output analysis.
Output only a JSON list of strings.

USER MESSAGE

Extract visually observable landmarks from the following instruction and return them as an
     ordered list.

Rules:
1. Keep only landmark noun phrases that can be identified in images.
2. Preserve useful visual attributes, such as color, size, material, style, facade, window
     patterns, and shape.
3. Remove action phrases, such as proceed, continue, turn left, turn right, and move forward
     .
4. Preserve the original order.
5. Output only a JSON list of strings.
```

```
6. Do not include explanations, prefixes, suffixes, Markdown, or example outputs.
7. Output a JSON list even if there is only one landmark.
8. If two mentions describe the same landmark at different times, retain the more complete
    phrase.

Instruction:
<instruction>
```

C Hyperparameter Selection

Table 4 summarizes the hyperparameter settings of our method. We explain the rationale behind these choices from the perspectives of action interface, computational budget and reward design.

Table 4: Hyperparameter settings in the method.

Symbol	Description	Value
$L_{\max}$	Maximum actions per sequence	4
G	Number of candidates in each GRPO group	5
c_{fmt}	Reward for a valid output format	0.1
$c^{+}_{\text{stop}}, c^{-}_{\text{stop}}$	Rewards for correct stop and continue decisions	1, 0.1
$\lambda_\ell, \lambda_d, \lambda_s$	Weights for the landmark-alignment, distance, and stop rewards	1, 1, 1
τ_ℓ	Sigmoid temperature for the landmark-alignment reward	1

Action Interface Selection The maximum action chunk length $L_{\max} = 4$ strikes a balance between policy invocation overhead and observation feedback frequency: predicting multiple actions at once reduces model calls, while limiting the action chunk length allows the policy to re-decide frequently based on new observations and constrains the scope of single candidate execution in RFT.

Candidate group size. Sampling $G = 5$ candidate action chunks per expert state provides multiple comparable outcomes for computing group-relative advantages while controlling the cost of simulation and judge calls. Larger groups can increase sampling coverage but require more executions and evaluations; this setting therefore balances candidate diversity against the training budget.

Reward scales and weights. The format reward $c_{\text{fmt}} = 0.1$ gives valid outputs a common small reward, while their relative ranking remains determined jointly by the landmark, distance, and stop rewards. Correct stopping and continuing receive $c^{+}_{\text{stop}} = 1$ and $c^{-}_{\text{stop}} = 0.1$, respectively, placing greater emphasis on correct termination while retaining feedback for appropriate continuation. The reward weights $(\lambda_\ell, \lambda_d, \lambda_s) = (1, 1, 1)$ use uniform explicit coefficients without introducing additional asymmetric weighting; they do not imply equal contributions during optimization.

D Human Validation and Data Overview of Landmark–Trajectory Stage Alignment

D.1 Human Validation

We randomly sample 100 of the 9,731 automatically annotated trajectories and manually annotate their trajectory stages. We compare the manual and automatic annotations in terms of the number of instruction landmarks and the target landmark assigned at each decision state. Landmark counts agree exactly for all 100 trajectories. For target-landmark assignments, 85 trajectories agree with the manual annotations at every decision state, while another 9 differ at only one state. Thus, the trajectory-level exact agreement rate is 85%, and 94% of trajectories in the random sample contain at most one target-landmark assignment discrepancy. These results indicate high agreement with human judgments on the inspected sample.

D.2 Data Distributions and Landmark Vocabulary

Figure 4 summarizes landmark counts, target-landmark assignment frequencies, and vocabulary distributions in the annotated data. In Figure 4(a), landmark counts are concentrated between 2 and

4 per trajectory, with 2 being the most common. The number of trajectories generally decreases as landmark counts increase, forming a long-tailed distribution. This indicates that the data include navigation tasks with varying numbers of landmarks. Figure 4(b) counts the decision states at which a landmark is selected as the target within its trajectory. Zero assignments form the most frequent category; nonzero assignments are concentrated at small state counts, with a tail extending to longer spans. This pattern is consistent with monotonic alignment allowing some landmarks to be skipped and others to cover multiple consecutive states.

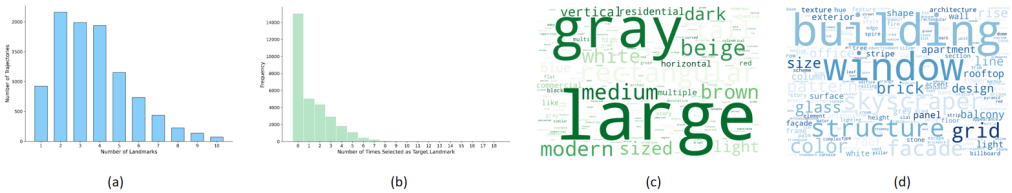


Figure 4: **Statistics and vocabulary of landmark–trajectory stage annotations.** (a) Distribution of the number of landmarks per trajectory. (b) Distribution of how often a landmark is selected as the target within its trajectory, counted over decision states; zero denotes a landmark that is never assigned as a target. (c) Word cloud of attribute terms in landmark phrases. (d) Word cloud of nouns in landmark phrases. Larger words indicate higher frequencies.

In Figure 4(c), prominent terms include colors such as *gray*, *white*, *beige*, and *brown*, as well as scale or style descriptors such as *large*, *medium*, and *modern*. Landmark descriptions therefore retain visually discriminative attributes in addition to object categories. Figure 4(d) is dominated by words referring to buildings and their components, such as *building*, *window*, *structure*, and *facade*, consistent with the urban character of the data. Together, these two types of vocabulary constitute semantic landmark descriptions that help associate visual references in instructions with trajectory observations.

E Baselines

To evaluate AlignFly's navigation performance, we compare it with four categories of baselines. (1) *Heuristic method*: Random Action samples actions without learning and indicates the difficulty of the evaluation trajectories. (2) *Conventional VLN methods*: Seq2Seq encodes instructions and observations for action prediction, while CMA combines them through recurrent networks and cross-modal attention (Krantz et al., 2020). (3) *Pretrained VLMs*: we directly evaluate Qwen2.5-VL-7B and Qwen2.5-VL-32B (Bai et al., 2025) to examine the effect of model size without navigation training. (4) *End-to-end UAV-VLN methods*: OpenFly-Agent is the supervised agent introduced with OpenFly (Gao et al., 2026); LongFly selects historical frames through spatiotemporal context encoding and is trained with SFT (Jiang et al., 2025); and AirVLN-R1 combines SFT and RFT (Cai et al., 2026). All baselines requiring task-specific training are trained on the same training trajectories as AlignFly and evaluated on the identical Test-Seen and Test-Unseen splits.

F Qualitative Analysis

Figure 5 presents a representative navigation trajectory. The instruction sequentially describes a gray building with rectangular windows, a beige-and-glass office building with rectangular and grid-like windows, and a medium-sized beige commercial building with a triangular roof. Observations 1–3 show the gray building and changes in the surrounding viewpoint. In observations 4–5, the beige-and-glass office building becomes the main visual reference ahead, while observations 6–8 reveal the subsequent commercial building and its triangular roof as the UAV approaches it. This sequence illustrates how visual references change as navigation progresses through the instruction stages.

This example illustrates that navigation actions must be related to the current stage's target landmark in addition to following motion descriptions such as moving forward and turning. After completing one stage, subsequent decisions should organize viewpoint adjustments and local progress around

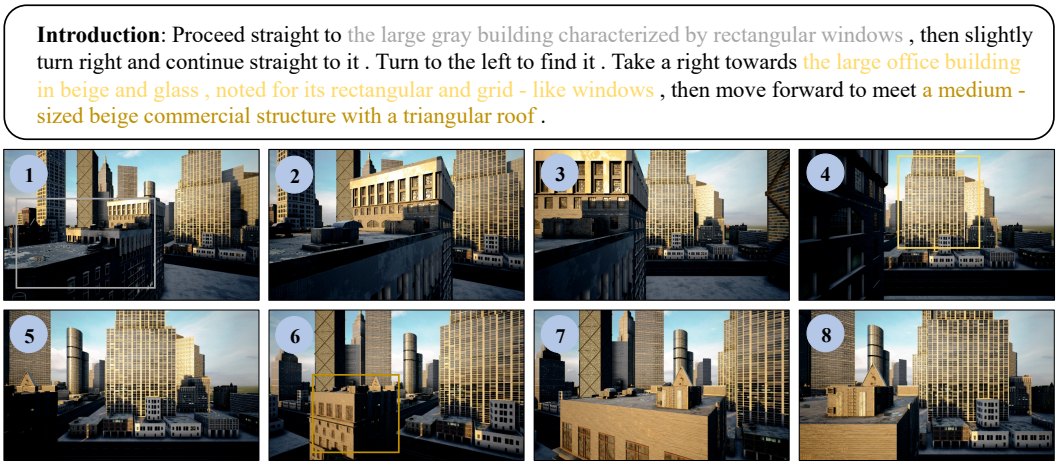


Figure 5: **Landmarks and stage transitions in a representative navigation trajectory.** The route instruction appears above eight observations arranged in temporal order. Colored landmark descriptions in the instruction correspond to the colored boxes, successively highlighting the gray building, the beige-and-glass office building, and the beige commercial building with a triangular roof as visual references for different navigation stages.

the next landmark. During training, landmark-alignment RFT compares evidence for the stage-specific target in post-action observations produced by different candidates from the same expert state, encouraging outcomes that better match the current semantic objective. The transition from an earlier building to subsequent buildings as visual references is consistent with the design of this stage-specific feedback.