# SIMPLIFYING COMPLEXITY: STAGE ADAPTIVE EVIDENCE ROUTING FOR UAV VLN

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

UAV vision-language navigation requires long-range search, target identification, and precise terminal control. However, existing methods typically use the same visual-language inputs throughout navigation, failing to account for the change in evidence requirements before and after target discovery. Before the target appears, navigation primarily relies on scene context and directional hints for exploration; once the target is reliably identified, the decision process should shift toward target-centered local control. To address this, we propose a stage-adaptive evidence-routing framework that divides navigation into exploration before target discovery and target-centered navigation afterward. The latter is further divided into approach, regrounding, and reacquisition. The framework combines hint-conditioned target identification with lightweight target memory to determine the current stage from the relationship between current and previously accepted target evidence. It then routes the corresponding visual and textual information to a shared action policy. On 3,005 TravelUAV test episodes, our method achieves success rates of 58.41%, 66.29%, and 40.71% on the Seen, Unseen Object, and Unseen Map splits, improving over AerialVLA by 10.45, 9.69, and 3.13 percentage points, respectively. These results show that adapting decision inputs to target discovery and changes in target evidence improves successful termination across diverse navigation conditions.

## 1 INTRODUCTION

Unmanned aerial vehicle vision-and-language navigation (UAV VLN), at the intersection of embodied intelligence and three-dimensional spatial perception, requires an agent to continuously perceive and autonomously navigate through large-scale, complex 3D environments by following unstructured natural-language instructions. Compared with ground robots, UAVs offer greater mobility and broader fields of view, making them promising for applications such as emergency search and rescue, autonomous delivery, and intelligent inspection (Liu et al., 2023b; Wang et al., 2025).

To address the challenges of cross-modal instruction understanding and large-scale spatial planning, existing UAV VLN methods commonly rely on end-to-end deep learning frameworks, including cross-modal architectures based on Transformers or graph neural networks. These methods typically construct a large monolithic model that jointly encodes continuous visual observations, trajectory history, and language instructions into a high-dimensional feature space for control prediction. Although effective during global exploration, they generally employ a monolithic and uniform processing strategy throughout navigation: the same complex model continuously fuses and reasons over the complete multimodal input, regardless of whether the UAV is conducting broad regional search or has already entered the terminal stage near the destination (Wang et al., 2025; Jiang et al., 2025; Zhang et al., 2026; Xu et al.).

However, this one-size-fits-all paradigm overlooks the intrinsic differences among the temporal stages of navigation. By examining the underlying physical and geometric properties of UAV VLN, we observe that the nature of the task changes fundamentally once the destination and its spatial location have been discovered and visually grounded within the perceptual range of the UAV. After destination confirmation, the original vision-language reasoning problem can be naturally simplified into goal-directed direct-flight control. This formulation is supported by two key properties:



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.

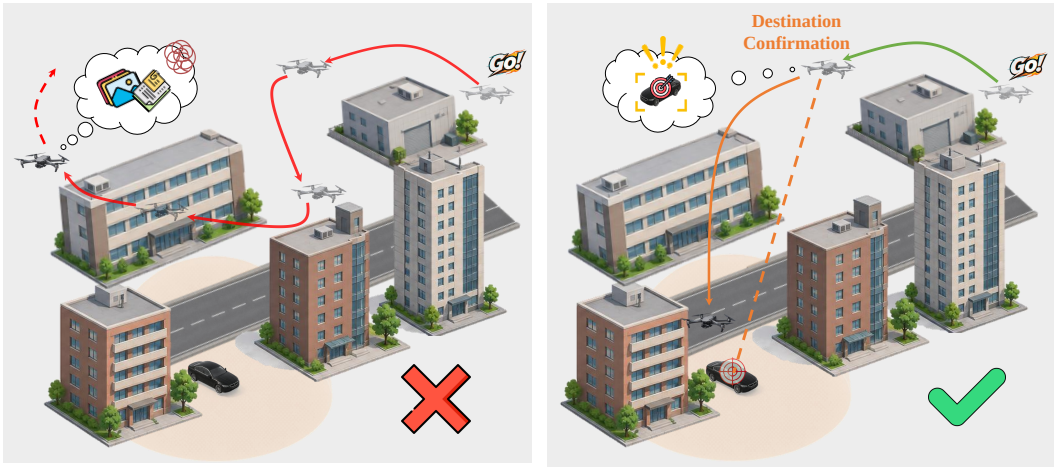



Figure 1: Existing methods continuously process complete multimodal information throughout the entire navigation process of locating the black car. Once the target is reliably localized, our method adaptively transitions from global exploration to goal-directed direct-flight control.

1. **Static destination.** Unlike targets in visual object tracking, the destination in a VLN episode remains stationary in physical space, and its absolute position does not drift over time.
2. **Unobstructed line of sight.** When the visual sensors of the UAV can clearly observe and localize the destination in three-dimensional space, the UAV and the target are connected by an unobstructed viewing ray. This direct sightline provides the geometric basis for locally approaching the destination through direct flight.

Based on these observations, we argue that once the destination has been confirmed, the primary requirement of the agent changes from instruction understanding and environmental search to precise target alignment and direct-flight control. Continuing to process the complete instruction, trajectory history, and background observations with a unified model at this stage provides limited additional guidance. Instead, as shown in Fig. 1, it introduces unnecessary computation and irrelevant information, complicating an otherwise straightforward control problem and potentially causing unstable decisions or trajectory deviation.

To address this issue, we propose a *Stage-Adaptive Evidence-Routing Framework*. The framework departs from the conventional paradigm of indiscriminately fusing all available information and explicitly divides UAV VLN into exploration before destination confirmation and target-centered navigation afterward. A hint-conditioned target identification module and a lightweight target memory jointly represent the currently available target evidence, integrating target discovery, evidence correction, and target recovery into an explicit stage-determination process. Before destination confirmation, the routing mechanism emphasizes global scene context and directional information from the instruction to guide efficient exploration in large-scale 3D environments. After destination confirmation, the navigation process switches to a goal-directed direct-flight mode. To handle dynamic disturbances and temporary loss of visibility during flight, target-centered navigation is further divided into three sub-stages: *approach*, *regrounding*, and *reacquisition*. Approach and regrounding suppress redundant background and linguistic information and retain only compact evidence associated with the target. Reacquisition combines current observations with target memory to rapidly recover a temporarily lost destination. The dynamically selected and denoised stage-specific evidence is then routed to a jointly trained Shared Action Policy. The proposed framework requires neither a complete observation history nor a high-dimensional environment map. Instead, lightweight evidence routing reduces irrelevant information and enables accurate, robust, and stage-adaptive continuous control.

Our main contributions are summarized as follows:

- We propose a target-discovery-driven decision framework for UAV vision-and-language navigation. The navigation process is explicitly divided into exploration before target discovery and target-centered navigation afterward, with the latter simplified from complex vision-language reasoning to target-conditioned direct-approach control.
- The proposed framework is implemented through hint-conditioned target identification, lightweight target memory, and stage-adaptive evidence routing. The navigation state is determined as exploration, approach, regrounding, or reacquisition according to the current target evidence, and stage-matched evidence is routed to a jointly trained Shared Action Policy for stage-adaptive continuous control.
- We conduct a comprehensive evaluation on 3,005 TravelUAV test episodes. Compared with AerialVLA, our method improves the success rate by 10.45, 9.69, and 3.13 percentage points and reduces the navigation error by 27.92, 22.36, and 13.29 m on the Seen, Unseen Object, and Unseen Map splits, respectively.

## 2 Related Work

### 2.1 Vision-and-Language Navigation for UAVs

Vision-and-language navigation (VLN) maps natural-language instructions and visual observations to goal-directed navigation behaviors. Early indoor studies explored policy learning, data augmentation, and cross-modal pretraining, followed by continuous-environment and multilingual benchmarks (Anderson et al., 2018; Fried et al., 2018; Wang et al., 2019; Tan et al., 2019; Zhu et al., 2020; Majumdar et al., 2020; Krantz et al., 2020; Ku et al., 2020; Guhur et al., 2021; Hong et al., 2021; Pashevich et al., 2021; Chen et al., 2021). UAV VLN extends these challenges to large-scale search and continuous motion control (Savva et al., 2019; Chaplot et al., 2020; Ramakrishnan et al., 2022). AerialVLN established an outdoor aerial navigation setting, while TravelUAV introduced continuous UAV control (Liu et al., 2023b; Wang et al., 2025). LongFly models long-horizon trajectory context, and NavFoM learns transferable navigation representations (Jiang et al., 2025; Zhang et al., 2026). AerialVLA directly maps fixed front and downward views to 3-DoF actions through vision-language-action modeling (Xu et al.; Liu et al., 2023a; Li et al., 2023; Driess et al., 2023; Brohan et al., 2023; Kim et al., 2024). Although these studies improve instruction understanding and navigation action prediction, they typically retain uniform input and decision processes, limiting their ability to accommodate the different evidence requirements of long-range search before target discovery and local control afterward. Our method enables the UAV to use scene context and directional cues for exploration before target discovery, then focus on target evidence for local control once the destination is reliably identified, aligning decision inputs with the current navigation task.

### 2.2 Vision-Language Target Grounding

Vision-language grounding associates language queries with image regions, enabling target localization beyond fixed category vocabularies. CLIP learns transferable visual-semantic representations through image-text contrastive learning (Radford et al., 2021). TransVG uses Transformers to establish cross-modal correspondence and directly regress bounding boxes for referring expressions (Deng et al., 2021), while MDETR learns text-conditioned detection through joint image-text reasoning (Kamath et al., 2021). GLIP unifies object detection and phrase grounding during pretraining (Li et al., 2022), and GLIPv2 extends this paradigm to jointly support localization and vision-language understanding (Zhang et al., 2022). For open-vocabulary detection, OWL-ViT transfers pretrained image-text representations to object localization, while Detic expands detector vocabularies using image-level supervision (Minderer et al., 2023; Zhou et al., 2022). OWLv2 further scales detector training through self-training on image-text pairs with automatically generated bounding-box annotations (Minderer et al., 2023). Grounding DINO combines grounded pretraining with cross-modal feature interaction to localize objects specified by category names or referring expressions (Liu et al., 2024). Although existing methods improve language-guided localization, single-frame matching remains vulnerable to same-category distractors and viewpoint changes. To address this issue, we combine directional constraints, instruction-based verification, and historical evidence to achieve stable target identification.

# 3 METHOD

We divide navigation into four stages: exploration, approach, regrounding, and reacquisition. At each decision step, the system extracts target evidence from the current observation, compares it with target memory, determines the navigation stage, and predicts an action using a shared policy. An overview is shown in Fig. 2.

## 3.1 OVERALL NAVIGATION LOOP

Given a natural-language instruction $I$, the UAV receives five RGB views:

$$\mathcal{O}_t = \left\{ O_t^{\text{front}}, O_t^{\text{left}}, O_t^{\text{right}}, O_t^{\text{rear}}, O_t^{\text{down}} \right\},$$

where the superscripts denote the front, left, right, rear, and downward cameras. The UAV also receives its world-frame position $\mathbf{p}_t$, orientation $\mathbf{q}_t$, and a discrete directional hint. Let $\mathbf{p}^{\text{goal}}$ denote the target position used only by the hint generator. The hint is computed as

$$\mathbf{d}_t^{\text{body}} = \mathbf{R}(\mathbf{q}_t)^\top \left( \mathbf{p}^{\text{goal}} - \mathbf{p}_t \right), \qquad \varphi_t = \operatorname{atan2}\left( d_{t,y}^{\text{body}}, d_{t,x}^{\text{body}} \right), \qquad h_t = \mathcal{H}(\varphi_t). \tag{1}$$

Here, $\mathbf{R}(\mathbf{q}_t)$ is the body-to-world rotation matrix induced by $\mathbf{q}_t$, $\operatorname{atan2}$ computes the horizontal target bearing, and $\mathcal{H}$ quantizes the continuous bearing into a discrete directional hint $h_t$. The quantization intervals and view-routing rules are provided in Appendix B. The hint contains only a coarse bearing and does not provide target distance or an obstacle-avoidance path.

We define the current input and target memory as

$$\mathcal{X}_t = (\mathcal{O}_t, I, h_t, \mathbf{p}_t, \mathbf{q}_t), \qquad M_t = (C_t^{\text{mem}}, \bar{\mathbf{p}}_t^{\text{mem}}, \nu_t^{\text{mem}}, \tau_t^{\text{mem}}),$$

where $C_t^{\text{mem}}$, $\bar{\mathbf{p}}_t^{\text{mem}}$, $\nu_t^{\text{mem}}$, and $\tau_t^{\text{mem}}$ are the stored target context, the UAV position at which it was recorded, its source camera, and its recording time, respectively. We set $M_t = \varnothing$ when no accepted target exists. Let $\mathcal{F}_{\text{SD}}$, $\Phi_{\text{SAI}}$, and $\pi_\theta$ denote the Stage Determination function, Stage-Adaptive Input function, and Shared Action Policy, respectively:

$$\begin{aligned} (s_t, E_t, M_{t+1}) &= \mathcal{F}_{\text{SD}}(\mathcal{X}_t, M_t), \\ (V_t, L_t) &= \Phi_{\text{SAI}}(s_t, \mathcal{X}_t, E_t, M_{t+1}), \\ a_t &= \pi_\theta(V_t, L_t). \end{aligned} \tag{2}$$

Here, $s_t$ is the current navigation stage, $E_t$ is the current target evidence, $M_{t+1}$ is the updated target memory, $V_t$ and $L_t$ are the visual and language inputs, and $a_t$ is the predicted action.

For a non-terminating action, the environment transition function $\mathcal{T}$ produces the next observation:

$$(\mathcal{O}_{t+1}, \mathbf{p}_{t+1}, \mathbf{q}_{t+1}) = \mathcal{T}(\mathcal{O}_t, \mathbf{p}_t, \mathbf{q}_t, a_t), \qquad a_t \in \mathcal{A}_{\text{move}}. \tag{3}$$

The loop terminates when the termination condition is satisfied or the maximum number of decision steps is reached.

## 3.2 STAGE DETERMINATION

The Stage Determination module constructs current target evidence, compares it with target memory, and selects the navigation stage. Thus, the stage represents the observable target-evidence state rather than elapsed flight time or traveled distance.

**Candidate generation and filtering.** Let $y_{\text{tgt}}$ be the final target name extracted from $I$, $\mathcal{V}$ the hint-conditioned view-routing function, $\mathcal{D}_{\text{GDINO}}$ the open-vocabulary detector, and $\mathcal{G}$ the geometric filtering function. They produce routed views, raw proposals, and filtered proposals:

$$\mathcal{O}_t^{\text{route}} = \mathcal{V}(h_t, \mathcal{O}_t), \qquad \mathcal{P}_t^{\text{raw}} = \mathcal{D}_{\text{GDINO}}\left( \mathcal{O}_t^{\text{route}}, y_{\text{tgt}} \right), \qquad \widetilde{\mathcal{P}}_t = \mathcal{G}(\mathcal{P}_t^{\text{raw}}, h_t). \tag{4}$$

Here, $\mathcal{O}_t^{\text{route}}$ is the subset of views sent to the detector, $\mathcal{P}_t^{\text{raw}}$ is the raw proposal set, and $\widetilde{\mathcal{P}}_t$ is the direction-consistent proposal set. The detailed routing and filtering rules are given in Appendix B.

We write the filtered proposals as

$$\widetilde{\mathcal{P}}_t = \{P_{t,i}\}_{i=1}^{N_t}, \qquad P_{t,i} = (C_{t,i}, \nu_{t,i}, \mathbf{b}_{t,i}, \rho_{t,i}),$$

where $N_t$ is the number of proposals, $i$ is the proposal index, $C_{t,i}$ is a context image containing the proposal and its surrounding scene, $\nu_{t,i}$ is the source camera, $\mathbf{b}_{t,i}$ is the normalized bounding box, and $\rho_{t,i}$ is the detection confidence.

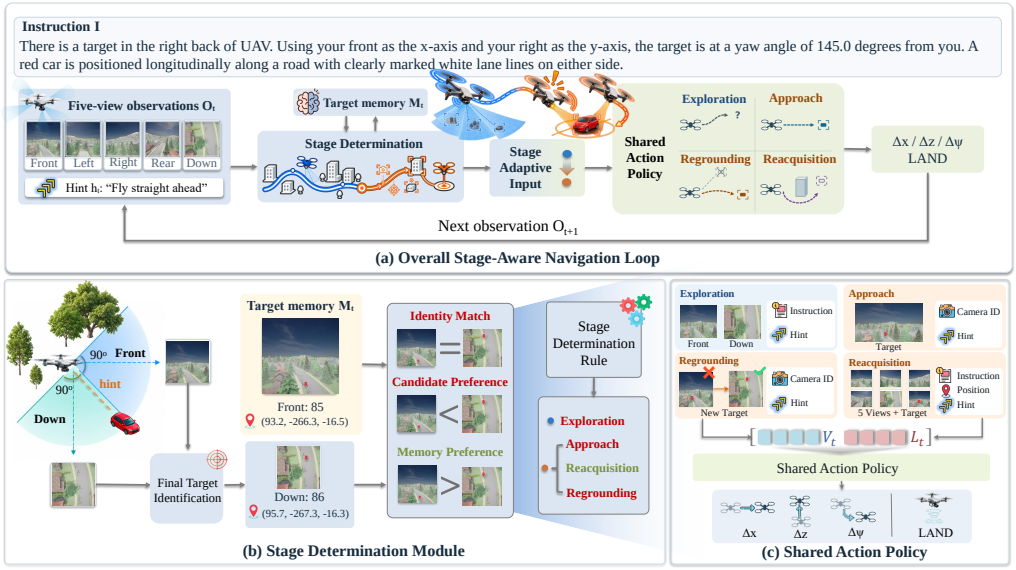



Figure 2: Overview of the proposed framework. (a) Overall Navigation Loop; (b) Stage Determination through target recognition, temporal association, and memory comparison; (c) Shared Action Policy with Stage-Adaptive Inputs.

**Target verification and current evidence.** Let $\mathcal{Q}_{\text{pres}}$ denote the target-presence verifier and $\mathcal{Q}_{\text{rank}}$ the candidate-ranking function. We first verify whether any proposal is consistent with the complete instruction:

$$z_t = \begin{cases} \mathcal{Q}_{\text{pres}}\left(I, y_{\text{tgt}}, h_t, \widetilde{\mathcal{P}}_t\right), & \widetilde{\mathcal{P}}_t \neq \varnothing, \\ \texttt{NONE}, & \widetilde{\mathcal{P}}_t = \varnothing, \end{cases} \qquad z_t \in \{\texttt{NONE}, \texttt{PRESENT}\}. \tag{5}$$

The value PRESENT means that at least one proposal is accepted under the instruction, whereas NONE means that no proposal is accepted. Only when $z_t = \texttt{PRESENT}$ do we rank the proposals:

$$i_t^\star = \mathcal{Q}_{\text{rank}}\left(I, \widetilde{\mathcal{P}}_t\right), \qquad i_t^\star \in \{1, \ldots, N_t\}. \tag{6}$$

The current evidence is

$$E_t = \begin{cases} P_{t,i_t^\star}, & z_t = \texttt{PRESENT}, \\ \varnothing, & z_t = \texttt{NONE}. \end{cases} \tag{7}$$

Separating presence verification from ranking prevents the ranker from selecting a distractor when no valid target is visible.

**Temporal comparison.** Let $\eta_t \in \{0, 1\}$ be the semantic-audit indicator. We set $\eta_t = 1$ when the current evidence requires full semantic verification and $\eta_t = 0$ when reliable temporal association allows the previously verified identity to be transferred. Let $\mathcal{Q}_{\text{cmp}}$ be the semantic comparison function. The relation between current evidence and memory is

$$r_t = \begin{cases} \mathcal{Q}_{\text{cmp}}\left(I, C_t^{\text{mem}}, C_t^{\text{cand}}\right), & E_t \neq \varnothing,\ M_t \neq \varnothing,\ \eta_t = 1, \\ \textsc{Identity Match}, & E_t \neq \varnothing,\ M_t \neq \varnothing,\ \eta_t = 0, \\ \perp, & \text{otherwise}. \end{cases} \tag{8}$$

Here, $C_t^{\text{cand}}$ is the context image of the current evidence. The relation IDENTITY MATCH denotes the same target instance, CANDIDATE PREFERENCE means that the current candidate is more consistent with the complete instruction, and MEMORY PREFERENCE means that the stored target should be kept. The symbol $\perp$ indicates that no comparison is performed. The association and audit rules for $\eta_t$ are specified in Appendix B.

Table 1: Stage-selection and memory-update rules.

| $E_t$ | $M_t$ | $r_t$ | $s_t$ | Memory operation |
|---|---|---|---|---|
| × | × | – | Exploration | None |
| ✓ | × | – | Approach | Store current evidence |
| ✓ | ✓ | IDENTITY MATCH | Approach | Refresh with current evidence |
| ✓ | ✓ | CANDIDATE PREFERENCE | Regrounding | Replace memory |
| ✓ | ✓ | MEMORY PREFERENCE | Reacquisition | Retain memory |
| × | ✓ | – | Reacquisition | Retain memory |

**Stage selection and memory update.** Let $\mathcal{S}$ denote the discrete stage-selection function:

$$s_t = \mathcal{S}(E_t, M_t, r_t). \tag{9}$$

The relations that permit the current evidence to update memory are

$$\mathcal{R}_{\text{upd}} = \{\text{IDENTITY MATCH}, \text{CANDIDATE PREFERENCE}\}.$$

The memory update is

$$M_{t+1} = \begin{cases} \left(C_t^{\text{cand}}, \mathbf{p}_t, \nu_t^{\text{cand}}, t\right), & E_t \neq \varnothing \text{ and } (M_t = \varnothing \text{ or } r_t \in \mathcal{R}_{\text{upd}}), \\ M_t, & \text{otherwise.} \end{cases} \tag{10}$$

Thus, the system stores the current evidence when a target is first accepted, identity is confirmed, or a better instruction-consistent candidate is found. It retains the previous memory when the current candidate is absent or rejected. The updated memory is used by the current action input and by the next stage decision.

### 3.3 SHARED ACTION POLICY

The Stage-Adaptive Input function $\Phi_{\text{SAI}}$ routes the visual and language evidence required by the current stage:

$$\begin{aligned} (V_t, L_t) &= \Phi_{\text{SAI}}\left(s_t, \mathcal{X}_t, E_t, M_{t+1}\right) \\ &= \begin{cases} \left((O_t^{\text{front}}, O_t^{\text{down}}), (I, h_t)\right), & s_t = \text{exploration}, \\ \left(C_t^{\text{cand}}, (\nu_t^{\text{cand}}, h_t)\right), & s_t \in \{\text{approach}, \text{regrounding}\}, \\ \left((\mathcal{O}_t, C_{t+1}^{\text{mem}}), (I, \mathbf{p}_t, \bar{\mathbf{p}}_{t+1}^{\text{mem}}, h_t)\right), & s_t = \text{reacquisition}. \end{cases} \end{aligned} \tag{11}$$

Here, $V_t$ and $L_t$ are the visual and language streams. Exploration uses the front and downward views with the full instruction; approach and regrounding use the accepted current target context, source camera, and hint; reacquisition uses all five views together with the memory context, current position, memory position, instruction, and hint.

All four stages share one action policy:

$$a_t = \pi_\theta(V_t, L_t) \in \mathcal{A}_{\text{move}} \cup \{\texttt{LAND}\}, \tag{12}$$

where $\pi_\theta$ has shared parameters $\theta$ and $\mathcal{A}_{\text{move}}$ is the motion-action set. A motion action is

$$a_t = \langle \Delta x_t, \Delta z_t, \Delta\psi_t \rangle,$$

where the three components denote forward displacement, vertical displacement, and yaw change. Their encoding and decoding are given in Appendix B.

Let $\mathbf{1}[\cdot]$ denote the indicator function and $\varepsilon_a$ the near-zero motion threshold. The termination decision is

$$d_t = \mathbf{1}\left[s_t \in \{\text{approach}, \text{regrounding}\}\right] \mathbf{1}\left[a_t = \texttt{LAND} \ \vee \ (a_t \in \mathcal{A}_{\text{move}} \ \wedge \ \|a_t\|_{\text{move}} \leq \varepsilon_a)\right]. \tag{13}$$

Landing is executed only when $d_t = 1$. The same output is ignored in exploration and target reacquisition, where the target is not currently reliably localized.

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETUP

**Dataset and difficulty division.** We evaluate our method on all 3,005 test episodes of the UAV-Need-Help task in TravelUAV (Wang et al., 2025), covering the Seen, Unseen Object, and Unseen Map splits. For each split, we report results under the Full, Easy, and Hard settings. Episodes with an expert trajectory no longer than 250 m are categorized as Easy, whereas those exceeding 250 m are categorized as Hard.

**Evaluation metrics.** We compare different methods in terms of final navigation error, successful termination, target reaching, and path efficiency using NE, SR, OSR, and SPL. NE measures the Euclidean distance between the UAV and the target at termination. SR and OSR respectively measure the percentage of episodes that terminate successfully within 20 m of the target and that enter this region at any point during navigation. SPL further evaluates the path efficiency.

**Implementation details.** We use Grounding DINO (Liu et al., 2024) to generate target candidates and Qwen3-VL-8B (Bai et al., 2025) to perform target-presence verification, candidate ranking, and temporal evidence comparison. The Shared Action Policy is jointly trained with four stage-specific input configurations corresponding to exploration, approach, regrounding, and reacquisition.

Grounding DINO uses a detection confidence threshold of $0.7$ and retains at most five candidates from each routed camera view. Geometric filtering uses an angular tolerance of $10^{\circ}$. Event-triggered semantic auditing uses a periodic validation interval of eight decision steps, an intersection-over-union (IoU) threshold of $0.2$, and a normalized center-distance threshold of $0.15$. Details of the training-data construction and model-training procedure are provided in Appendix A.

**Baselines.** We compare our method with TravelUAV (Wang et al., 2025), NavFoM (Zhang et al., 2026), LongFly (Jiang et al., 2025), and AerialVLA (Xu et al.), which respectively represent modular planning, general navigation representation learning, long-horizon contextual modeling, and end-to-end continuous control. For a fair comparison, all baseline methods receive the same directional hint $h_t$.

## 4.2 MAIN RESULTS

We evaluate the proposed method on the Seen, Unseen Object, and Unseen Map splits under the Full, Easy, and Hard settings. In the Full setting, our method achieves the lowest NE and the highest SR, OSR, and SPL on all three splits. Compared with AerialVLA, it improves SR by $10.45$, $9.69$, and $3.13$ percentage points and reduces NE by $27.92$, $22.36$, and $13.29$ m on Seen, Unseen Object, and Unseen Map, respectively. The simultaneous improvements in OSR and SR on Seen and Unseen Object indicate that our method not only reaches the target region more frequently, but also converts target encounters into successful termination more reliably. The corresponding SPL gains further suggest that this improvement is achieved with more efficient navigation trajectories. On Unseen Map, our method also outperforms AerialVLA in OSR ($53.32\%$ vs. $52.92\%$), SR ($40.71\%$ vs. $37.58\%$), NE ($54.13$ m vs. $67.42$ m), and SPL ($31.68\%$ vs. $28.22\%$), demonstrating its ability to generalize to unseen environments. Moreover, its higher SR and lower NE across all three Hard subsets show that stage-adaptive evidence routing remains effective in more challenging long-horizon navigation scenarios.

## 4.3 ABLATION STUDY

We further evaluate the contributions of the four stages: exploration, approach, regrounding, and reacquisition. Removing a stage only disables its stage-specific evidence routing; when that stage is triggered, the policy instead uses a fixed multimodal input comprising the complete five-view observation, full instruction, and directional hint. As shown in Table 5, the full model achieves the best overall performance across all three splits. The performance degradation of *w/o Exploration* and *Exploration only* demonstrates the complementary roles of environmental exploration before target discovery and local control afterward. The remaining ablations further confirm the importance of target approach, evidence correction, and memory-guided target recovery for reliable navigation.

Table 2: Results on the Seen split. NE is reported in meters; SR, OSR, and SPL are reported as percentages.

| Method | Full ($n = 1418$) | | | | Easy ($n = 783$) | | | | Hard ($n = 635$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) |
| TravelUAV | 98.66 | 17.45 | 48.87 | 15.76 | 66.40 | 20.26 | 51.23 | 18.10 | 138.04 | 14.02 | 45.98 | 12.90 |
| NavFoM | 93.05 | 29.17 | 49.24 | 25.03 | 58.98 | 32.91 | 53.16 | 27.87 | 143.83 | 23.58 | 43.40 | 20.80 |
| LongFly | 60.02 | 36.39 | 65.87 | 31.07 | 38.10 | 38.52 | 71.90 | 31.24 | 85.20 | 33.94 | 58.94 | 30.88 |
| AerialVLA | 65.88 | 47.96 | 57.69 | 38.54 | 43.76 | 49.30 | 61.30 | 37.14 | 93.16 | 46.30 | 53.23 | **40.26** |
| **Ours** | **37.96** | **58.41** | **79.06** | **40.62** | **19.87** | **64.10** | **85.73** | **42.36** | **60.26** | **51.39** | **70.83** | 38.48 |

Table 3: Results on the Unseen Object split. NE is reported in meters; SR, OSR, and SPL are reported as percentages.

| Method | Full ($n = 629$) | | | | Easy ($n = 209$) | | | | Hard ($n = 420$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) |
| TravelUAV | 118.11 | 22.42 | 46.90 | 20.51 | 86.12 | 24.40 | 49.28 | 22.03 | 134.03 | 21.43 | 45.71 | 19.75 |
| NavFoM | 108.04 | 29.83 | 47.99 | 27.20 | 70.51 | 32.54 | 50.72 | 29.54 | 133.01 | 28.03 | 46.18 | 25.64 |
| LongFly | 66.74 | 43.87 | 64.56 | 38.39 | 54.84 | 38.01 | 56.84 | 31.36 | 57.07 | 50.25 | 74.16 | 45.27 |
| AerialVLA | 61.45 | 56.60 | 64.86 | 46.61 | **45.72** | 56.94 | 64.11 | 43.76 | 69.27 | 56.43 | 65.24 | 48.03 |
| **Ours** | **39.09** | **66.29** | **81.83** | **53.56** | 46.04 | **62.00** | **72.90** | **45.51** | **35.63** | **68.43** | **86.27** | **57.57** |

Table 4: Results on the Unseen Map split. NE is reported in meters; SR, OSR, and SPL are reported as percentages.

| Method | Full ($n = 958$) | | | | Easy ($n = 561$) | | | | Hard ($n = 397$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) | NE(↓) | SR(↑) | OSR(↑) | SPL(↑) |
| TravelUAV | 138.80 | 4.18 | 20.77 | 3.84 | 102.94 | 4.63 | 22.82 | 4.24 | 189.46 | 3.53 | 17.88 | 3.28 |
| NavFoM | 125.10 | 6.30 | 18.95 | 5.68 | 102.41 | 6.77 | 20.07 | 6.04 | 170.58 | 5.36 | 15.71 | 4.97 |
| LongFly | 108.32 | 11.27 | 30.27 | 9.32 | 78.56 | 12.96 | 34.31 | 10.32 | 148.10 | 9.02 | 24.88 | 7.98 |
| AerialVLA | 67.42 | 37.58 | 52.92 | 28.22 | **44.99** | 41.89 | **58.47** | 29.72 | 99.11 | 31.49 | 45.09 | 26.11 |
| **Ours** | **54.13** | **40.71** | **53.32** | **31.68** | 53.19 | **45.81** | 57.14 | **33.92** | **55.47** | **33.50** | **47.92** | **28.52** |

### 4.4 Controlled Evaluation of Stage-Adaptive Routing

To isolate the effect of evidence routing, we compare the full model with a controlled variant that uses the complete five-view observation and the full instruction at every navigation stage, without stage-specific input selection. Both variants are evaluated under the Full setting on the Seen, Unseen Object, and Unseen Map splits. The results show that stage-adaptive routing achieves lower NE and higher SR, OSR, and SPL across all three splits, demonstrating that matching the policy input to the current navigation stage effectively improves navigation performance.

### 4.5 Navigation Visualization

As shown in Figure 3, compared with existing methods, our method enables the UAV to adaptively transition from exploration to target-centered navigation after destination confirmation and fly toward the target along an approximately straight terminal trajectory, thereby reducing unnecessary trajectory deviations near the destination.

### 4.6 Limitations

Although the explicit stage-determination mechanism improves navigation effectiveness, its repeated analysis of visual evidence also increases decision latency. In future work, we will investigate lightweight stage predictors, visual feature caching, and asynchronous updates to reduce redundant inference while preserving the benefits of stage-aware navigation. Furthermore, we will explore

Table 5: Ablation of exploration and target-centered navigation stages.

| Variant | Before Discovery | Target-Centered Navigation | | | Seen | | | Unseen Object | | | Unseen Map | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Exploration | Approach | Regrounding | Reacquisition | SR↑ | OSR↑ | SPL↑ | SR↑ | OSR↑ | SPL↑ | SR↑ | OSR↑ | SPL↑ |
| w/o Exploration | × | ✓ | ✓ | ✓ | 21.20 | 31.37 | 10.50 | 26.12 | 35.20 | 15.43 | 8.60 | 18.72 | 3.96 |
| Exploration only | ✓ | × | × | × | 47.87 | 57.52 | 34.33 | 56.43 | 64.66 | 41.83 | 37.43 | 51.80 | 19.64 |
| w/o Approach | ✓ | × | ✓ | ✓ | 51.61 | 63.20 | 35.78 | 58.80 | 69.79 | 44.11 | 38.11 | 44.50 | 22.23 |
| w/o Regrounding | ✓ | ✓ | × | ✓ | 56.77 | 72.82 | 39.20 | 63.43 | 78.10 | 50.90 | 39.81 | 50.21 | 28.13 |
| w/o Reacquisition | ✓ | ✓ | ✓ | × | 54.52 | 68.15 | 37.59 | 61.32 | 74.20 | 47.84 | 38.90 | 47.40 | 25.15 |
| Full model | ✓ | ✓ | ✓ | ✓ | **58.41** | **79.06** | **40.62** | **66.29** | **81.83** | **53.56** | **40.71** | **53.32** | **31.68** |

Table 6: Controlled comparison between fixed multimodal inputs and stage-adaptive evidence routing under the Full setting. NE is reported in meters, while SR, OSR, and SPL are reported as percentages.

| Variant | Seen | | | | Unseen Object | | | | Unseen Map | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NE↓ | SR↑ | OSR↑ | SPL↑ | NE↓ | SR↑ | OSR↑ | SPL↑ | NE↓ | SR↑ | OSR↑ | SPL↑ |
| Fixed multimodal input | 48.83 | 52.40 | 60.82 | 33.97 | 47.20 | 57.24 | 66.54 | 46.21 | 61.72 | 30.20 | 44.15 | 20.72 |
| Stage-adaptive routing | **37.96** | **58.41** | **79.06** | **40.62** | **39.09** | **66.29** | **81.83** | **53.56** | **54.13** | **40.71** | **53.32** | **31.68** |

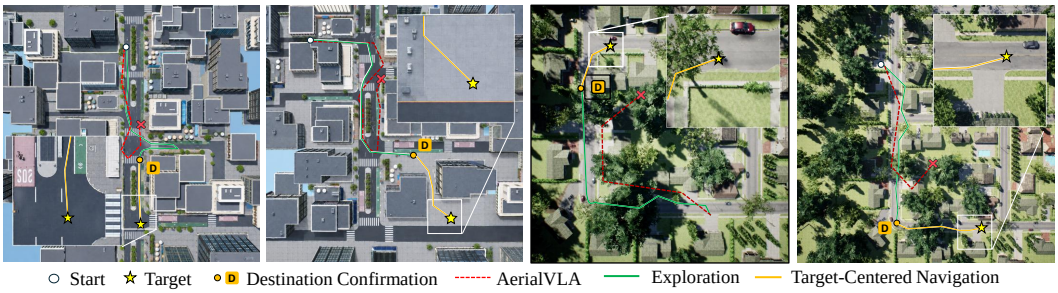



Figure 3: Visualization results of four navigation examples.

uncertainty-aware adaptive audit strategies and cross-platform deployment methods to further improve the real-time performance and robustness of the system in complex environments and under resource-constrained conditions.

## 5 Conclusion

We propose a stage-adaptive evidence-routing framework for UAV vision-and-language navigation. The framework is motivated by the observation that once a static destination is reliably grounded in the visual observations, the navigation task shifts from language-conditioned environmental exploration to target-centered local control. It combines hint-conditioned target identification with lightweight target memory to divide navigation into four stages: exploration, approach, regrounding, and reacquisition. Evidence matched to the current stage is then routed to a unified Shared Action Policy. Across 3,005 TravelUAV test episodes, our method achieves success rates of 58.41%, 66.29%, and 40.71% on the Seen, Unseen Object, and Unseen Map splits, respectively. Compared with AerialVLA, it improves the success rate by 10.45, 9.69, and 3.13 percentage points, while reducing the navigation error by 27.92, 22.36, and 13.29 m, respectively. These results validate our central premise that adaptively adjusting decision evidence according to target discovery and subsequent changes in target evidence can improve both successful task termination and final target proximity within a unified navigation policy. To further improve the real-time applicability of the framework, future work will investigate lightweight stage predictors, visual feature caching, and asynchronous stage updates to reduce stage-determination latency and enhance system robustness, while exploring adaptive evidence fusion and safety-aware control under dynamic occlusion, complex weather, cross-platform deployment, and real-world UAV environments.

## REFERENCES


Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 3674–3683. IEEE, 2018.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

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

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1749–1759. IEEE, 2021.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 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. *Advances in neural information processing systems*, 31, 2018.

Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1614–1623. IEEE, 2021.

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

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. *arXiv preprint arXiv:2512.22010*, 2025.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 1760–1770. IEEE, 2021.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model, 2024. *URL https://arxiv. org/abs/2406.09246*, 1(2):4, 2024.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, 2020.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

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

Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision*, pp. 259–274. Springer, 2020.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15922–15932. IEEE, 2021.

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

Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18868–18878. IEEE, 2022.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. IEEE, 2019.

Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2610–2621, 2019.

Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Hongsheng Li, Yue Liao, Si Liu, et al. Towards realistic uav vision-language navigation: Platform, benchmark, and methodology. In *International Conference on Learning Representations*, volume 2025, pp. 7292–7310, 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6622–6631. IEEE, 2019.

P Xu, Z Deng, J Deng, Z Gu, and S Wan. Aerialvla: A vision-language-action model for uav navigation via minimalist end-to-end control. arxiv 2026. *arXiv preprint arXiv:2603.14363*.

Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Advances in neural information processing systems*, 35:36067–36080, 2022.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *International Conference on Learning Representations*, volume 2026, pp. 127293–127322, 2026.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European conference on computer vision*, pp. 350–368. Springer, 2022.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10009–10019. IEEE, 2020.

# A Training Data Construction and Model Training

## A.1 Dataset Construction

**Collection of raw observations.** The data sources comprise simulation-based recollection of official TravelUAV training tasks, existing navigation records, and targeted collection for target ambiguity and recovery scenarios. For each official task, we read the navigation instruction, trajectory, and target-instance metadata. We restore the designated target and the UAV's initial pose in AirSim, then follow an expert trajectory toward the target while recording RGB images, depth maps, target-instance masks, and the UAV pose. Images are captured from five cameras: front, left, right, rear, and downward-facing.

Let $d$ denote the distance from the UAV to the target. Observations are recorded every four action steps when $d > 150\,\mathrm{m}$, every two steps when $75\,\mathrm{m} < d \leq 150\,\mathrm{m}$, and at every step when $d \leq 75\,\mathrm{m}$. Training and validation sets are partitioned by official navigation task; recollected observations and derived samples from the same task are assigned to the same split.

A target is annotated as visible in a view when its instance mask covers at least 16 pixels and has valid depth values. The target bounding box is obtained from the mask boundary, and target depth is defined as the median valid depth within the mask. Targeted collection additionally records same-category distractors, observations of the target from different poses, and observations following target disappearance.

**Construction of task-specific samples.** The collected observations and existing navigation records are used to construct supervised samples for object detection, target-presence judgment, candidate ranking, temporal comparison, and action prediction. Table 7 summarizes their inputs, supervision, and the sizes of the listed training manifests.

Temporal comparison uses the three relation names defined in the main text: Identity Match, Memory Preference, and Candidate Preference. When constructing a comparison sample, we first use the instance identities recorded in simulation to determine whether the current and remembered candidates refer to the same target. For candidates corresponding to different instances, the target specified by the task instruction determines whether the remembered candidate is retained or the current candidate is selected.

Table 7: Construction and sizes of the listed task-training manifests. For object detection, each pair gives the number of records before sampling → the manifest length after sampling; all other values are manifest lengths.

| Module | Sample construction and supervision | Size (entries) |
|---|---|---|
| Object detection | Derive target boxes from instance masks and incorporate recovery observations and multiple camera views. Each record contains an image, target name, and target box. | Initial localization: $3{,}840 \rightarrow 4{,}632$<br>Recovery weighting: $4{,}800 \rightarrow 8{,}472$<br>Camera balancing: $5{,}038 \rightarrow 13{,}275$ |
| Target presence | Construct initial binary samples from candidate panels and condition subsequent samples on directional hints. Each record contains candidate inputs and a presence label. | Initial binary set: 2,638<br>Directional-hint set: 8,734 |
| Candidate ranking | Arrange detection proposals, same-category distractors, and false detections into numbered panels. Target-present records label the correct candidate; target-absent records provide auxiliary rejection supervision. | Positive candidates: 3,000<br>Balanced set: 2,638 |
| Temporal comparison | Pair remembered and current candidate evidence and label their identity or preference relation. | Pairwise preference: 4,192<br>Three-way relations: 2,816<br>Targeted extension: 792 |
| Action policy | Organize observations, instructions, and expert actions by navigation stage. Label the discrete action and, when applicable, the landing marker. | Exploration, approach, regrounding, and reacquisition: 9,049 each |

### A.2 Model Training

**Model initialization and supervision.** The object detector is initialized from pretrained Grounding DINO Tiny. Target-presence judgment, candidate ranking, and temporal comparison each initialize a separate LoRA adapter from Qwen3-VL-8B-Instruct. At inference time, these three tasks share the base model and switch to their corresponding task adapters. The action policy starts from OpenVLA-7B with an existing navigation adapter and is subsequently trained by navigation stage.

Language and action responses are trained with a causal language-modeling objective. Labels for the input prompt and padding positions are set to $-100$; only the target response and its end marker contribute to the loss. Let $x$ be the input, $y$ the target response, and $\mathcal{R}$ the supervised response-token positions. The objective is

$$\mathcal{L} = -\sum_{j \in \mathcal{R}} \log p_\theta(y_j \mid x, y_{<j}). \tag{14}$$

**Training stages.** The stages inherited by the evaluated checkpoints are listed in Table 8. Manifest names and sizes shared with Table 7 are kept consistent.

The initial-localization and recovery-weighting stages train the detector decoder for three and two epochs, respectively. Recovery weighting increases the sampling frequency of recovery observations; specifically, 120 recovery records are each sampled eight times. The camera-balancing stage continues from the preceding checkpoint, expands the trainable visual components, and enables the auxiliary decoder loss.

The directional-hint stage of target-presence training comprises $2 + 20 + 30 + 20 = 72$ consecutive updates. All four segments use the same 8,734-entry manifest. Adding 67 target-visible records produces the subsequent 8,801-entry manifest. The final 268-entry replay manifest contains 134 target-present and 134 target-absent records.

Candidate ranking first learns to select the correct candidate from 3,000 target-present records, then trains on a balanced manifest containing 1,319 target-present and 1,319 target-absent records. The target-absent records provide auxiliary rejection supervision during training. At inference time,

Table 8: Task-training stages inherited by the evaluated checkpoints. For object detection, each pair gives the number of records before sampling → the manifest length after sampling; all other sizes are manifest lengths. Update counts are the completed optimizer steps recorded for each stage.

| Module | Training stage | Size (entries) | Learning rate | Updates |
|---|---|---|---|---|
| Object detection | Initial localization | $3{,}840 \rightarrow 4{,}632$ | $10^{-5}$ | 348 |
| | Recovery weighting | $4{,}800 \rightarrow 8{,}472$ | $5 \times 10^{-6}$ | 424 |
| | Camera balancing | $5{,}038 \rightarrow 13{,}275$ | $10^{-5}$ | 200 |
| Target presence | Initial binary set | 2,638 | $10^{-4}$ | 50 |
| | Directional-hint set | 8,734 | $10^{-4}$ | 72 |
| | Full target-presence set | 8,801 | $10^{-4}$ | 20 |
| | Class-balanced replay | 268 | $5 \times 10^{-5}$ | 20 |
| Candidate ranking | Positive candidates | 3,000 | $10^{-4}$ | 50 |
| | Balanced set | 2,638 | $5 \times 10^{-5}$ | 50 |
| Temporal comparison | Pairwise preference | 4,192 | $10^{-4}$ | 210 |
| | Three-way relations | 2,816 | $5 \times 10^{-5}$ | 141 |
| | Targeted extension | 792 | $10^{-5}$ | 40 |
| Action policy | Exploration | 9,049 | $10^{-5}$ | 452 |
| | Approach | 9,049 | $5 \times 10^{-5}$ | 452 |
| | Regrounding | 9,049 | $5 \times 10^{-5}$ | 452 |
| | Reacquisition | 9,049 | $5 \times 10^{-5}$ | 452 |
| | Four-stage joint adaptation | 9,109 | $5 \times 10^{-7}$ | 455 |

candidate ranking is still invoked only after target-presence judgment returns `PRESENT`, as defined in the main text. Temporal comparison progresses from pairwise preference to the three relations defined in Section A.1. The targeted extension contains 264 records for each relation.

The action policy first undergoes stage-wise adaptation for exploration, approach, regrounding, and reacquisition. The final stage initializes from the reacquisition checkpoint and jointly adapts the shared policy using a mixed manifest whose sampling proportions are adjusted across navigation stages. Its 9,109-entry manifest contains 5,010 exploration, 1,822 approach, 911 regrounding, and 1,366 reacquisition records. Action inputs use the target evidence stored in the training records.

# B Routing, Evidence Association, and Action Details

## B.1 Directional Hint and View Routing

Let $\varphi_t \in [-\pi, \pi)$ be the horizontal target bearing from Eq. equation 1. The quantizer $\mathcal{H}$ uses the following intervals:

| Directional hint | Angular interval |
|---|---|
| front | $\lvert\varphi_t\rvert \le 15^\circ$ |
| front-left/right | $15^\circ < \lvert\varphi_t\rvert \le 60^\circ$ |
| left/right | $60^\circ < \lvert\varphi_t\rvert \le 120^\circ$ |
| rear-left/right | $\lvert\varphi_t\rvert > 120^\circ$. |

The sign of $\varphi_t$ determines the left or right branch. The hint is used only to reduce the candidate search views; it does not provide target distance or a complete motion path.

The routing function $\mathcal{V}$ selects views according to $h_t$. A front hint routes the front and downward views. Front-side, side, and rear-side hints route the corresponding horizontal camera together with the downward view. Left and right directions are handled symmetrically.

### B.2 Candidate Context and Geometric Filtering

The open-vocabulary detector $\mathcal{D}_{\mathrm{GDINO}}$ produces proposals from the routed views. Let $u_{t,i} \in [0,1]$ be the horizontal center of the normalized box $\mathbf{b}_{t,i}$, $\alpha_{\nu_{t,i}}$ the yaw offset of the source camera relative to the UAV front direction, and $f_{\nu_{t,i}}$ its horizontal field of view. The estimated bearing is

$$\widehat{\varphi}_{t,i} = \mathrm{wrap}\left(\alpha_{\nu_{t,i}} + \left(u_{t,i} - \frac{1}{2}\right) f_{\nu_{t,i}}\right), \tag{15}$$

where $\mathrm{wrap}$ maps angles to $[-\pi, \pi)$. Let $\Theta(h_t)$ be the angular interval associated with $h_t$ and $d_\angle$ the minimum angular distance to that interval. A horizontal-view proposal is retained when

$$d_\angle\left(\widehat{\varphi}_{t,i}, \Theta(h_t)\right) \le \varepsilon_\varphi, \tag{16}$$

where $\varepsilon_\varphi$ is the angular tolerance. Downward-view proposals are filtered according to their direction relative to the image center, while central proposals are retained directly. The context image is constructed by $\mathrm{CropPad}$, which crops the detected region and pads it with a fixed surrounding margin so that scene layout and spatial relations remain available to the semantic verifier.

### B.3 Target Verification and Temporal Association

Target-presence verification and candidate ranking are performed by a vision- language model. Proposals with detector confidence below $0.7$ are removed, and at most five proposals are retained from each routed camera. The ranking function $\mathcal{Q}_{\mathrm{rank}}$ is called only after $\mathcal{Q}_{\mathrm{pres}}$ returns `PRESENT`.

To avoid repeated full semantic verification, candidates in adjacent frames are temporally associated. Let $\mathrm{IoU}$ denote intersection over union and let $\mathbf{c}_{t,i}$ be the normalized center of a candidate box. A current candidate $P_{t,i}$ is associated with a previous candidate $P_{t-1,j}$ when they come from the same camera and satisfy

$$\mathrm{IoU}\left(\mathbf{b}_{t,i}, \mathbf{b}_{t-1,j}\right) \ge \tau_{\mathrm{iou}} \quad \text{or} \quad \left\|\mathbf{c}_{t,i} - \mathbf{c}_{t-1,j}\right\|_2 \le \tau_{\mathrm{ctr}}.$$

We use

$$\tau_{\mathrm{iou}} = 0.2, \qquad \tau_{\mathrm{ctr}} = 0.15.$$

A full semantic audit is triggered when a candidate first appears or reappears, when the number or source cameras change, when association is missing or ambiguous, or when the periodic interval is reached. In these cases $\eta_t = 1$; otherwise $\eta_t = 0$ and the previously verified identity is transferred to the associated current candidate. The periodic audit interval is

$$K_{\mathrm{audit}} = 8$$

decision steps. Only target identity is transferred; the context image and bounding box delivered to the action policy always come from the current observation.

### B.4 Semantic Evidence Comparison

When $E_t \neq \varnothing$, $M_t \neq \varnothing$, and $\eta_t = 1$, the comparison function $\mathcal{Q}_{\mathrm{cmp}}$ evaluates the current and memory contexts under the complete instruction and outputs

$$r_t \in \{\textsc{Identity Match}, \textsc{Candidate Preference}, \textsc{Memory Preference}\}.$$

The comparison considers target category, attributes, spatial relations, and instruction consistency rather than detector confidence. An IDENTITY MATCH refreshes memory, a CANDIDATE PREFERENCE replaces the stored target, and a MEMORY PREFERENCE rejects the current candidate and preserves the stored target.

### B.5 Action Encoding and Termination

The Shared Action Policy predicts three discrete motion codes:

$$\widehat{\mathbf{c}}_t = \left(\widehat{c}_{x,t}, \widehat{c}_{z,t}, \widehat{c}_{\psi,t}\right), \qquad \widehat{c}_{x,t}, \widehat{c}_{z,t}, \widehat{c}_{\psi,t} \in \{0, \ldots, 98\}.$$

The codes are linearly decoded as

$$\Delta x_t = 5\frac{\widehat{c}_{x,t}}{98}, \qquad \Delta z_t = -5 + 10\frac{\widehat{c}_{z,t}}{98}, \qquad \Delta\psi_t = -1.1 + 2.2\frac{\widehat{c}_{\psi,t}}{98}, \tag{17}$$

with physical ranges $\Delta x_t \in [0, 5]$ m, $\Delta z_t \in [-5, 5]$ m, and $\Delta\psi_t \in [-1.1, 1.1]$ rad. The token `LAND` is an additional discrete termination action.

Let $\varepsilon_a$ be the near-zero motion threshold. A termination request is produced when the policy outputs `LAND` or a motion action satisfying

$$a_t \in \mathcal{A}_{\text{move}}, \qquad \|a_t\|_{\text{move}} \leq \varepsilon_a.$$

The request is executed only in the approach and regrounding stages. It is disabled during exploration, where no target has been accepted, and during target reacquisition, where the accepted target lacks reliable current localization.

### B.6 Implementation Configuration

Grounding DINO is used for proposal generation, while Qwen3-VL-8B performs target-presence verification, candidate ranking, and semantic evidence comparison. The Shared Action Policy is jointly trained on the four stage-specific input configurations. The detector confidence threshold is $0.7$, with at most five proposals retained per routed camera. The geometric filter uses a $10^\circ$ angular tolerance. Event-triggered auditing uses an eight-step periodic interval, an IoU threshold of $0.2$, and a normalized center-distance threshold of $0.15$.