RCPNAV: REVERSIBLE CRUISE–PRECISION CONTROL WITH KEYFRAME MEMORY AND LANDMARK GRAPHS FOR AERIAL VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Aerial vision-and-language navigation requires a UAV to use onboard visual observations to locate a destination described in natural language, with applications in urban inspection and search. Reaching the destination requires efficient long-distance flight as well as accurate localization near the target. A one-way coarse-to-fine controller may switch prematurely to local search and cannot return to cruise even if the target remains distant. Long flights also make it difficult to retain useful past observations and reason about spatial relations among named landmarks. We propose RCPNav, which combines three components: a reversible cruise–precision controller that can return to cruise when the target is estimated to be farther away; selective keyframe memory that retrieves earlier views conditioned on the instruction; and a language-grounded landmark graph that organizes named places to support goal prediction. On all three evaluation splits of the refined CityNav benchmark, RCPNav achieves the highest success rate and success weighted by path length among the compared methods. Ablation experiments with separately retrained variants show that all three components improve navigation performance.

Keywords: Aerial Vision-and-Language Navigation; Reversible Cruise–Precision Control; Historical Keyframe Memory; Language-Grounded Landmark Graph; Long-Horizon UAV Navigation; Geographic Semantic Map.

1 INTRODUCTION

Vision-and-language navigation (VLN) asks agents to combine natural-language instructions with visual observations to reach a specified destination. R2R and subsequent work in continuous environments have advanced visual-language alignment and sequential decision-making (Anderson et al., 2018; Krantz et al., 2020). Aerial VLN (AVLN) extends this task to UAVs, which must make decisions across larger three-dimensional environments under changing aerial viewpoints (Liu et al., 2023b). CityNav further studies navigation in reconstructed real cities, where an agent must combine visual observations with geographic semantic information to locate a destination described in language (Lee et al., 2025).

Urban-scale navigation requires both long-range travel toward a destination and precise localization near it. The former benefits from a global goal estimate, whereas the latter requires frequent action updates based on local observations. Staged approaches address these different decision scales (Ding et al., 2026; Fan et al., 2026). However, the goal estimate used to switch modes can change with subsequent observations. If an agent enters local navigation prematurely and cannot return to long-range control, its action scale cannot adapt to the revised estimate.

Sustained destination estimation also depends on visual evidence acquired earlier in the flight. Landmarks mentioned in an instruction may no longer appear in the current view, making past observations informative beyond the current image. Prior work has modeled navigation history with temporal encoding or spatial memory (Chen et al., 2021; Wang et al., 2023b; Ding et al., 2026; Jiang et al., 2025). With limited memory capacity, an agent must decide which views to retain and how to extract evidence relevant to the instruction and current goal prediction.

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.

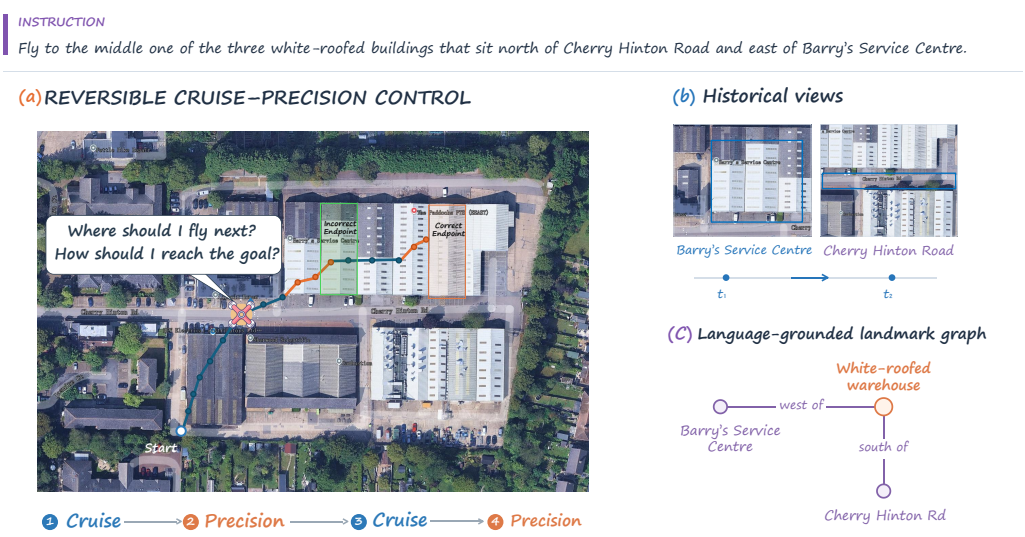

Figure 1: Illustration of RCPNav. Given a language goal described through named landmarks, (a) the agent switches from cruise to precision, returns to cruise after a premature local search, and refines its final approach; (b) historical views retain visual evidence of the referenced landmarks; and (c) a landmark-relation sketch shows how the landmarks locate the target.

Named landmarks also provide spatial cues beyond the current image. Locating a destination can require reasoning about the names and positions of several places rather than recognizing one landmark in isolation. GeoNav and LookasideVLN have explored structured spatial information and landmark relationships for aerial navigation (Xu et al., 2026; Ning et al., 2026). In CityNav, geometric information about named landmarks enables a spatial representation for predicting the final destination. A further challenge is to fuse this representation with current and historical visual evidence.

We propose RCPNav to address these challenges. **First**, its reversible cruise–precision controller updates the navigation mode according to estimated goal proximity, allowing a return to cruise after premature entry into precision. **Second**, it retains two selected historical keyframes and retrieves instruction-relevant visual features to complement the current view. **Third**, its language-grounded landmark graph combines the names, geometry, and spatial proximity of known landmarks to support final-destination prediction. A shared multimodal Transformer fuses these sources with the instruction, current observation, pose, and map, and predicts normalized global coordinates of the final destination, relative direction, and Progress as goal proximity.

Our method exhibits robust performance on both the original and refined CityNav datasets, and extensive experiments consistently demonstrate its superiority over all existing state-of-the-art baselines on key metrics.

2 Related Work

Vision-and-language navigation. R2R established the task of following natural-language route instructions in indoor environments, while REVERIE extended navigation to locating objects described in language (Anderson et al., 2018; Qi et al., 2020). Subsequent benchmarks broadened the task: RxR introduced multilingual instructions with dense alignment to navigation trajectories; TouchDown studied navigation and spatial reasoning in urban street views; and VLN-CE required agents to execute low-level actions in continuous environments (Ku et al., 2020; Chen et al., 2019; Krantz et al., 2020). Navigation models developed alongside these benchmarks. PREVALENT investigated vision-language pretraining, VLN-BERT introduced a recurrent Transformer, and ScaleVLN expanded training data (Hao et al., 2020; Hong et al., 2021; Wang et al., 2023c). HAMT and HOP modeled observation history, while Iterative VLN studied memory across repeated navigation tasks (Chen et al., 2021; Qiao et al., 2022; Krantz et al., 2023). DUET combined local perception with global planning, and DREAMWALKER explored planning in continuous environments (Chen et al., 2022;

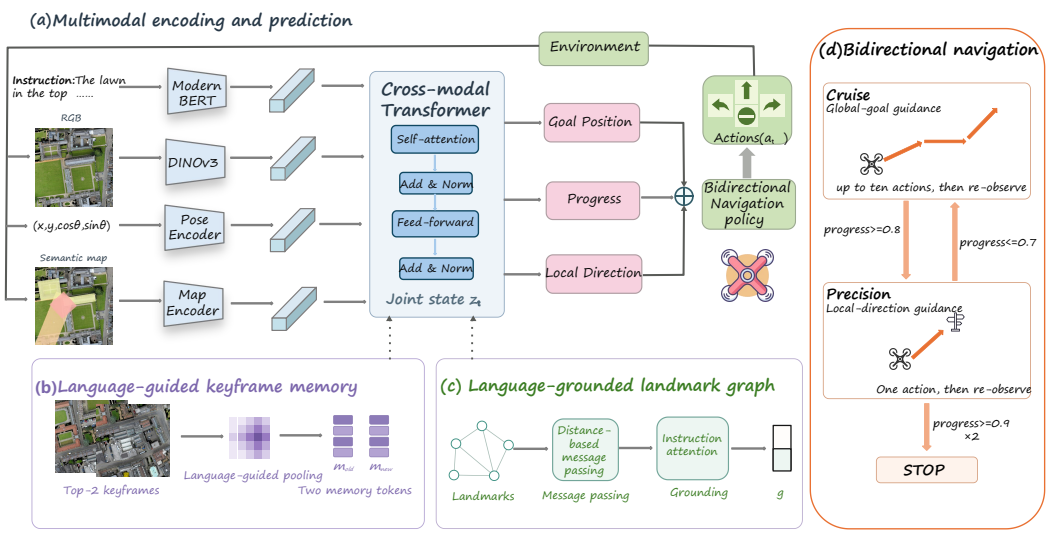

Figure 2: Multimodal prediction and bidirectional navigation in RCPNav. (a) The instruction, current RGB view, pose, and online semantic map, together with (b) two selected historical keyframes and (c) a language-grounded landmark graph, are encoded as tokens and fused by a shared multimodal Transformer. Its three heads predict the normalized global coordinates of the final destination to guide navigation, the relative direction $(\sin \Delta\text{yaw}, \cos \Delta\text{yaw})$ for local actions in precision, and Progress as goal proximity for mode switching and stopping. (d) Cruise follows the global goal, while precision follows the relative direction; Progress at or above 0.8 enters precision, one estimate at or below 0.7 returns to cruise, and two consecutive raw estimates at or above 0.9 trigger stopping.

Wang et al., 2023a). Progress estimation and backtracking have also supported navigation decisions (Ma et al., 2019a;b). Spatial grounding has been studied through cross-modal maps, bird's-eye-view scene graphs, visual-language maps, and entity–landmark alignment (Georgakis et al., 2022; Liu et al., 2023a; Huang et al., 2023; Cui et al., 2023).

Aerial vision-and-language navigation. Aerial VLN extends language-guided navigation to UAV flight, with larger outdoor environments and changing viewpoints. AerialVLN established a simulated benchmark for following flight instructions; OpenUAV and OpenFly subsequently expanded aerial environments and navigation data (Liu et al., 2023b; Wang et al., 2025; Gao et al., 2026). OpenFly-Agent also incorporated historical keyframes into its navigation policy (Gao et al., 2026). CityNav introduced language-described destinations grounded in named landmarks across reconstructed real cities, together with geographic semantic maps (Lee et al., 2025). CityNavAgent explored hierarchical semantic planning and trajectory memory for urban aerial navigation (Zhang et al., 2025). Methods developed for this setting have pursued different approaches: FlightGPT uses vision-language models for navigation, GeoNav emphasizes geospatial reasoning, and HETT combines coarse-to-fine navigation with historical context (Cai et al., 2025; Xu et al., 2026; Ding et al., 2026). HTNav explores tiered decision-making, while LookasideVLN uses directional information about landmarks (Fan et al., 2026; Ning et al., 2026).

3 Method

3.1 Problem Formulation

In CityNav, a UAV must navigate to a destination specified by a natural-language instruction I (Lee et al., 2025). At decision step t, RCPNav uses the current RGB observation V_t, planar pose $\mathbf{p}_t = (x_t, y_t, \theta_t)$, an online three-channel geographic semantic map M_t, and the named landmarks and footprints available for the current city block. The UAV flies at a fixed altitude in our implementation. Earlier observations may be retained in a causal memory. The ground-truth destination $\mathbf{g}^*$ is available for supervision during training but is unknown at inference time.

The model predicts a global estimate $\hat{\mathbf{g}}_t$ of the *final destination*, a relative heading $\hat{\mathbf{u}}_t$, and a progress estimate $\hat{r}_t$. A controller converts these predictions into turns, forward movements, or STOP. The central decision is whether the UAV should continue traveling toward its estimated destination in cruise mode or inspect the nearby area with shorter, more frequent decisions in precision mode.

3.2 RCPNAV OVERVIEW

RCPNav combines a shared destination predictor with a reversible cruise–precision controller. The predictor draws on the current view, two selected historical keyframes, and a graph of named landmarks. Keyframes retain previously observed visual evidence, while the graph describes the positions and proximity of places mentioned in the instruction.

A two-layer, 12-head Transformer with width 768 fuses six groups of tokens through self-attention: instruction tokens, one current-view token, up to two historical-keyframe tokens, one enhanced-pose token, one map token combining global and local map features, and one language-grounded landmark-graph token (Figure 2). The latter five groups are constructed by the model from observations and geographic priors; they are not separate inputs supplied by the environment. Three prediction heads read their respective fused contextual representations to predict the normalized global coordinates of the final destination $\hat{\mathbf{g}}_t \in [0,1]^2$, the relative direction $\hat{\mathbf{u}}_t = (\sin\widehat{\Delta\theta}_t, \cos\widehat{\Delta\theta}_t)$, and Progress $\hat{r}_t$ as goal proximity. Historical keyframes and the landmark graph thus influence all three predictions through fusion; their outputs are not separately concatenated at the prediction heads.

ModernBERT-base encodes up to 256 instruction tokens and provides their masked-mean representation $\boldsymbol{\ell}$ (Warner et al., 2025). DINOv3 ViT-B/16 encodes 224×224 RGB images (Siméoni et al., 2025); a learned mixture of its last four dense feature layers is adapted by a 1×1 convolution, group normalization, and GELU to a $512 \times 7 \times 7$ feature map. The current view is pooled into a visual token. The pose token augments $[\sin\theta_t, \cos\theta_t, \bar{x}_t, \bar{y}_t]$ with four Fourier position bands and higher-order yaw harmonics through a zero-initialized residual. The map token combines an encoded global three-channel map with a zero-initialized residual from a UAV-centered, yaw-aligned $80\,\mathrm{m} \times 80\,\mathrm{m}$ crop. The crop is sampled at 64×64 resolution and contains three semantic, two egocentric-coordinate, and one validity channel.

3.3 REVERSIBLE CRUISE–PRECISION CONTROL

Progress is trained as a truncated linear function of the true distance to the final target:

$$r_t^* = \mathrm{clip}\left(1 - \frac{\|\mathbf{p}_t - \mathbf{g}^*\|_2}{100\,\mathrm{m}}, 0, 1\right), \qquad \hat{d}_t = 100\,\mathrm{m}\,[1 - \mathrm{clip}(\hat{r}_t, 0, 1)]. \tag{1}$$

Here $\mathbf{g}^*$ is the final target. At test time, $\hat{d}_t$ is only a proxy inferred from progress; the target position is unknown. The target saturates beyond 100 m, so the proxy is used for the near-goal transition and stopping rather than for measuring the whole route.

Let $q_t \in \{\mathsf{Cruise}, \mathsf{Precision}\}$ be the current mode. Before updating it or moving, the controller checks $\hat{r}_t \geq 0.90$ for two consecutive decisions and stops if the check passes. Otherwise it updates the mode by

$$q_{t+1} = \begin{cases} \mathsf{Precision}, & q_t = \mathsf{Cruise} \text{ and } \hat{d}_t \leq 20\,\mathrm{m}, \\ \mathsf{Cruise}, & q_t = \mathsf{Precision} \text{ and } \hat{d}_t \geq 30\,\mathrm{m}, \\ q_t, & \text{otherwise.} \end{cases} \tag{2}$$

The 20–30-m interval holds the current mode. Entry requires one estimate at or below 20 m, so the agent can inspect a possible goal immediately; return to cruise requires one updated estimate at or above 30 m. The different thresholds provide hysteresis between the two modes. The agent starts in cruise; neither mode is terminal, and the two-confirmation STOP check always has priority.

Cruise. The destination head predicts the final normalized global goal, not an intermediate waypoint. We take the coordinate-wise median of up to the five most recent causal predictions and map it to world coordinates. At each primitive action the controller recomputes the bearing to this fixed goal. A heading error greater than 30° triggers a 30° rotation; otherwise the UAV advances approximately 5 m. A cruise macro-action contains at most ten primitives and at most three forward advances (about

15 m), then the model observes again. The macro-action also ends if another forward move can no longer reduce the distance to its fixed predicted goal, including after overshooting.

Precision. The local direction head outputs a normalized $(\sin \Delta\theta, \cos \Delta\theta)$ pair. If the magnitude of its decoded heading error exceeds $15°$, the controller turns by the predicted angle; otherwise it advances about 5 m. It then immediately obtains a new observation. A rollout permits at most 30 model decisions. Stopping takes precedence over both modes, with two consecutive progress readings at or above 0.90, corresponding to an estimated distance of at most 10 m; this is stricter than the 20-m evaluation radius.

3.4 Selective Language-Conditioned Historical Memory

The memory $\mathcal{K}_t$ contains at most $K = 2$ keyframes. We measure the wrapped difference between UAV yaw values at consecutive *decision starts*, so multiple primitive turns within one cruise macro-action do not each create a separate candidate. A frame becomes a candidate when $|\operatorname{wrap}(\theta_t - \theta_{t-1})| \geq 15°$; a turn in precision mode can therefore also make the next observed frame a candidate. If the instruction matches a landmark footprint, its score is the visible fraction of the referenced footprint in the current field of view. When there is no valid match, the normalized yaw change supplies a fallback score. We retain the two highest-scoring candidates encountered so far; ties favor newer frames, and readout follows chronological order. The first frame serves as a replaceable fallback with score $-\infty$. The current frame is written only after producing the current decision and is available at $t + 1$, so selection cannot see future observations.

DINOv3 features adapted to $512 \times 7 \times 7$ provide 49 visual positions $\mathbf{v}_{ij}$ for stored frame i. A masked-mean instruction representation ℓ queries these positions:

$$a_{ij} = \operatorname{softmax}_j \left(\frac{(W_q \ell)^\top W_k \mathbf{v}_{ij}}{\sqrt{d}} \right), \tag{3}$$

$$\mathbf{m}_i = W_m \frac{1}{49} \sum_{j=1}^{49} \mathbf{v}_{ij} + W_r \sum_{j=1}^{49} a_{ij} W_v \mathbf{v}_{ij}. \tag{4}$$

We initialize W_r at zero so the attention branch starts as a residual to mean pooling. A stored frame still contributes only one 768-dimensional token; empty slots are masked. This separates the cost of storing useful views from the length of the flown route. Figure 3 traces admission and readout.

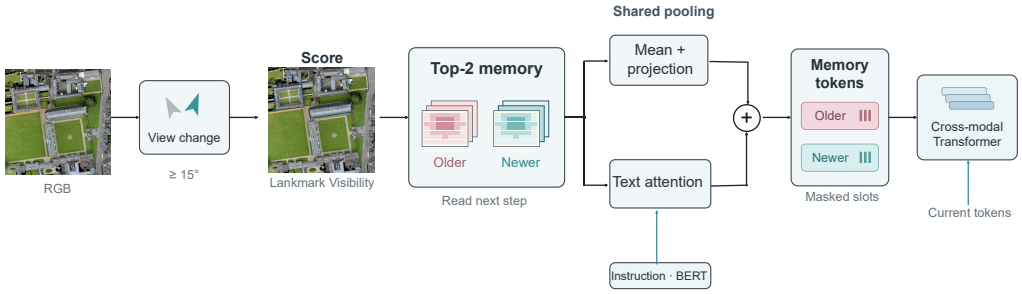

Figure 3: Historical keyframe selection and readout. At each decision start, an observation becomes a candidate when the view changes by at least $15°$. Visibility of instruction-referenced landmarks ranks candidates, with yaw change providing a fallback score when no landmark matches. The two highest-scoring frames are retained and read in chronological order at the next decision. Mean pooling and instruction-guided attention produce one memory token per frame, which enters the cross-modal Transformer alongside current tokens.

3.5 Language-Grounded Landmark Graph

For each city block, we construct nodes from named CityRefer landmark annotations (Miyanishi et al., 2023). If multiple annotations share a name within a block, we retain the one with the largest footprint. This yields 1,110 block–landmark-name nodes in refined CityNav. Each node combines a

cached name embedding from frozen ModernBERT with its normalized map-center coordinates and log footprint area:

$$\mathbf{h}_i = \mathrm{LayerNorm}(\mathrm{ReLU}(W_n[\mathrm{TextEnc}(name_i);\ \bar{x}_i;\ \bar{y}_i;\ \log(1+area_i)])) \,. \tag{5}$$

Each node has width 768; a block contains at most approximately 190 nodes. We form one row-normalized distance-weighted message layer using landmark-center distances in meters (rather than the normalized coordinates used in the node features):

$$A_{ij} = \frac{\exp(-\|\mathbf{p}_i - \mathbf{p}_j\|^2/(2\sigma^2))}{\sum_k \exp(-\|\mathbf{p}_i - \mathbf{p}_k\|^2/(2\sigma^2))}, \quad \mathbf{h}'_i = \mathbf{h}_i + \tanh\left(W_o \sum_j A_{ij} W_s \mathbf{h}_j\right), \tag{6}$$

where σ is learned and initialized at 25 m. The weights encode proximity, without explicit directional labels. Attention from the current instruction selects the graph readout,

$$\alpha_i = \mathrm{softmax}_i\left(\frac{(W_q^g \boldsymbol{\ell})^\top W_k^g \mathbf{h}_i}{\sqrt{768}}\right), \qquad \mathbf{z}_t^G = W_g \sum_i \alpha_i \mathbf{h}'_i. \tag{7}$$

The attention weights use the pre-message features $\mathbf{h}_i$; the pooled values use the updated features $\mathbf{h}'_i$. The 768-dimensional $\mathbf{z}_t^G$ is appended to the multimodal sequence, with both the weight and bias of W_g initialized to zero. An auxiliary head predicts s_i from $[\mathbf{h}'_i; \alpha_i \mathbf{h}'_i]$. Its target $y_i = \exp[-\|\mathbf{p}_i - \mathbf{g}^*\|^2/(2(25\,\mathrm{m})^2)]$ encodes proximity to the final destination, and $\mathcal{L}_{graph} = |\mathcal{V}|^{-1} \sum_{i\in\mathcal{V}} (\mathrm{sigmoid}(s_i) - y_i)^2$ averages over valid nodes $\mathcal{V}$. Instruction-conditioned attention supplies the language grounding; this auxiliary target supervises goal proximity, not name matching. The relevance scores do not participate in the STOP rule.

3.6 Training Objective

We supervise the final-destination coordinates with mean squared error $\mathcal{L}_{goal}$, the relative heading with $\mathcal{L}_{dir}$, and progress with $\mathcal{L}_{prog}$. A class-weighted logistic term $\mathcal{L}_{thr}$ emphasizes the stopping threshold. The complete objective at training epoch e is

$$\mathcal{L} = 0.75\mathcal{L}_{dir} + 5.00\mathcal{L}_{goal} + 0.50\mathcal{L}_{prog} + 0.10\mathcal{L}_{thr} + \mathbb{1}[e \geq 21]\, 0.10\mathcal{L}_{graph}. \tag{8}$$

The goal loss is coordinate MSE. Progress examples within 30 m receive weight 3, and those in the 10-m stopping region weight 6; the threshold term is a class-weighted logistic objective at $r = 0.90$. The direction target is emphasized within the 10–30-m precision range. In epochs 1–20, the graph-token position remains in the sequence but its content is zero and graph parameters are frozen; in epochs 21–30, the graph is active and its auxiliary loss carries weight 0.10. Batches alternate between expert and student rollouts, with one rollout type per batch and a factor of two on the imitation objective to match the expectation of a paired update. Validation uses student rollouts. This is DAgger-style training on states reached by the learner (Ross et al., 2011).

4 Experiments

4.1 Experimental Setup

Datasets and protocol. Original CityNav contains 32,637 episodes: 22,002 training, 2,498 val-seen, 2,826 val-unseen, and 5,311 test-unseen episodes (Lee et al., 2025). We train RCPNav on the CityNav version refined by HETT (Ding et al., 2026), which corrects landmark annotations and excludes 311 episodes. Its corresponding split sizes are 21,878, 2,470, 2,697, and 5,281. For retained episodes, the instruction text and demonstration trajectories are unchanged. CityNav divides the episodes into training, val-seen, val-unseen, and test-unseen. The val-seen split shares scenes with training and measures performance in seen environments, whereas val-unseen and test-unseen contain scenes not encountered during training and measure cross-scene generalization. The training and val-seen splits cover the same 24 scenes; val-unseen and test-unseen cover 4 and 6 unseen scenes, respectively.

Following CityNav's standard protocol, an episode is successful if the UAV stops within $20\,\mathrm{m}$ of the destination. For episode i, let $\mathbf{g}_i$ be the target position, $\mathbf{p}_i^T$ the final position, P_i the traveled path

length, and L_i the reference shortest-path length. We define the terminal-success and oracle-success indicators as

$$S_i = \mathbb{1}\left[\|\mathbf{p}_i^T - \mathbf{g}_i\|_2 \le 20\,\mathrm{m}\right], \qquad O_i = \mathbb{1}\left[\min_t \|\mathbf{p}_i^t - \mathbf{g}_i\|_2 \le 20\,\mathrm{m}\right]. \tag{9}$$

Navigation error (NE) measures the average Euclidean distance between the final position and the target,

$$\mathrm{NE} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i^T - \mathbf{g}_i\|_2. \tag{10}$$

Success rate (SR) measures the percentage of episodes that terminate within the success radius, while oracle success rate (OSR) measures the percentage whose trajectory enters that radius at any time:

$$\mathrm{SR} = \frac{100}{N}\sum_{i=1}^{N} S_i, \qquad \mathrm{OSR} = \frac{100}{N}\sum_{i=1}^{N} O_i. \tag{11}$$

Success weighted by path length (SPL) combines terminal success with path efficiency:

$$\mathrm{SPL} = \frac{100}{N}\sum_{i=1}^{N} S_i \frac{L_i}{\max(P_i, L_i)}. \tag{12}$$

NE evaluates final localization accuracy; SR evaluates whether the agent stops successfully; OSR evaluates whether it reaches the target neighborhood at any point; and SPL evaluates successful navigation together with path efficiency. Because OSR does not require the UAV to stop within the success radius, it can exceed SR when the UAV passes through the target neighborhood but does not terminate there. NE is reported in meters, while SR, OSR, and SPL are reported as percentages.

Implementation. We train for 30 epochs on four RTX 4090 GPUs with a global batch size of 16, AdamW, weight decay 0.01, and FP16 mixed precision. Navigation components use a learning rate of 10^{-4}. ModernBERT remains frozen. The last two DINOv3 Transformer blocks are unfrozen from epoch 11 with a learning rate of 10^{-5}.

4.2 Comparison with Prior Methods

Table 1 reports navigation results on the original and refined CityNav evaluation splits.

Table 1: Navigation results on original and refined CityNav. A star marks evaluation on refined CityNav. NE is in meters; SR, OSR, and SPL are percentages. Each numeric column has one bold model result; Human is excluded from this ranking.

	val-seen				val-unseen				test-unseen			
Method	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑
Random	222.3	0.0	1.2	0.0	223.0	0.0	0.9	0.0	208.8	0.0	1.4	0.0
Seq2Seq	257.1	1.8	7.9	1.6	317.4	0.8	8.8	0.6	245.3	1.5	8.3	1.3
CMA	240.8	1.0	9.4	0.9	268.8	0.7	7.9	0.6	252.6	0.8	9.7	0.8
AerialVLN	65.6	9.8	23.8	8.6	81.8	6.8	17.9	5.7	64.1	8.1	19.1	5.9
MGP	53.0	16.9	29.9	14.4	73.8	8.4	17.9	7.1	86.1	10.9	20.2	9.9
HETT	45.2	25.2	48.4	23.0	62.1	17.5	25.1	14.5	72.9	23.0	39.3	17.0
HETT*	37.2	31.1	51.9	25.8	51.3	19.1	34.8	16.7	40.4	28.9	**49.6**	23.8
HTNav	47.20	28.30	47.92	25.07	59.80	15.85	34.43	13.55	68.50	22.23	41.02	19.63
HTNav*	37.50	31.05	51.82	27.48	49.80	17.69	**37.82**	15.44	40.30	25.49	48.82	21.56
RCPNav	36.1	46.0	56.9	44.1	60.6	19.2	31.3	17.8	70.6	25.0	37.1	23.4
RCPNav*	**27.44**	**49.76**	**60.73**	**47.62**	**48.54**	**21.32**	34.04	**19.87**	**38.88**	**29.98**	43.31	**28.04**
Human	9.1	89.3	96.4	60.2	9.4	88.4	95.5	62.7	9.8	87.9	95.3	57.0

On refined CityNav, RCPNav* obtains the highest SR and SPL and the lowest NE among the compared methods on all three evaluation splits. On test-unseen, it reaches 29.98% SR and 28.04% SPL. HETT* and HTNav* obtain higher OSR on some unseen splits. OSR only indicates whether the trajectory enters the target neighborhood; it does not require the UAV to stop there and does not measure path efficiency. Considering final success, terminal localization, and path efficiency together, RCPNav* provides the stronger overall navigation performance.

On the original CityNav dataset, RCPNav obtains higher SR and SPL than the compared methods on all three evaluation splits: 46.0% and 44.1% on val-seen, 19.2% and 17.8% on val-unseen, and 24.99% and 23.41% on test-unseen, respectively.

4.3 Ablation Studies

To assess the roles of reversible control, historical keyframes, and the landmark graph, we perform single-component ablations on the full refined CityNav val-unseen split. We remove one component at a time, retrain the corresponding model, and compare it with the full model under the same evaluation setting.

Table 2: Ablation results on refined CityNav val-unseen. All variants are separately retrained. NE is in meters; SR, OSR, and SPL are percentages.

Method	NE↓	SR↑	OSR↑	SPL↑
RCPNav (full)	**48.54**	**21.32**	**34.04**	**19.87**
w/o Reversible Cruise–Precision Control	58.08	16.73	28.92	14.60
w/o History Keyframe	56.71	17.49	29.46	16.62
w/o Landmark Graphs	52.65	20.14	31.67	18.62

Removing reversible control produces the largest performance decline: SR and SPL fall by 4.59 and 5.27 percentage points, NE rises by 9.54 m, and OSR falls by 5.12 percentage points. Allowing the UAV to return from precision to cruise helps correct premature local search and improves final navigation success.

Without historical keyframes, SR and SPL fall by 3.83 and 3.25 percentage points, NE rises by 8.17 m, and OSR falls by 4.58 percentage points. Historical visual information can complement the current observation with evidence that helps localize the goal and adjust the route.

Without the landmark graph, SR and SPL fall by 1.18 and 1.25 percentage points, NE rises by 4.11 m, and OSR falls by 2.37 percentage points. Although its effect is smaller than those of the other two components, the language-grounded landmark relations improve spatial reasoning near the target and navigation performance.

All three ablations increase NE and decrease SR, OSR, and SPL, indicating that reversible control, historical keyframes, and the landmark graph each contribute to the full model's navigation performance.

4.4 Analysis of Returns to Cruise

Reversible cruise–precision control lets the UAV adjust its navigation scale after entering precision mode based on subsequent observations. When an updated goal estimate no longer supports continued local search, the controller can return to cruise and resume navigation using the global goal prediction. To examine how this mechanism operates during evaluation, we count precision-to-cruise transitions and compare their frequency across dataset versions and evaluation splits.

Table 3: Precision-to-cruise transition events across evaluation splits.

	Refined CityNav			Original CityNav		
Split	Episodes	Returns/100	Events	Episodes	Returns/100	Events
val-seen	2,470	2.9	72	2,498	3.8	94
val-unseen	2,697	8.5	228	2,826	9.6	270
test-unseen	5,281	5.8	308	5,311	8.5	450

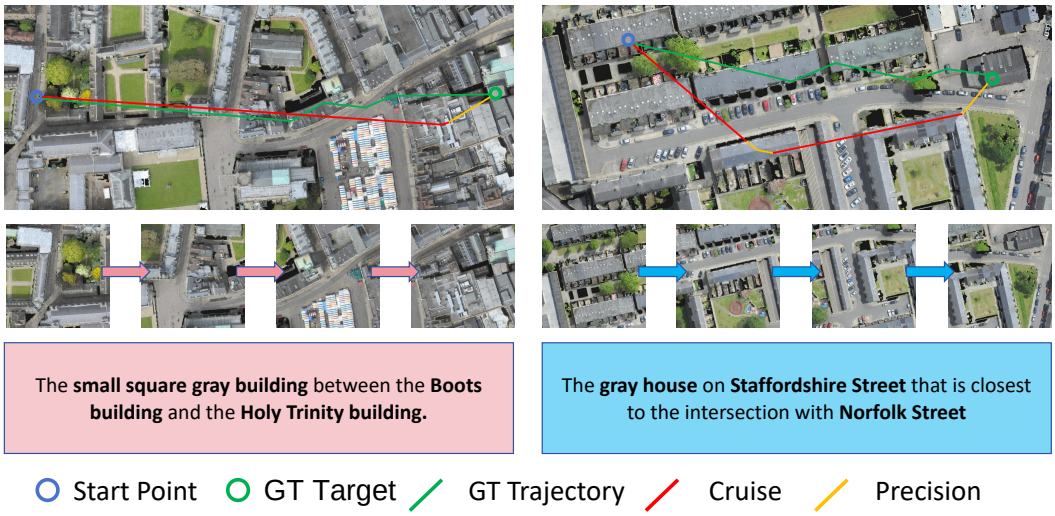

Figure 4: Two language-goal navigation examples. Each column shows the goal instruction, a top-down trajectory, and selected observations along the route. Blue and green circles mark the start and ground-truth target; green, red, and yellow paths denote the ground-truth trajectory, cruise, and precision, respectively. The left example enters precision near the target. In the right example, the agent returns to cruise after an early switch to precision, then re-enters precision near the target.

Returns to cruise occur in every evaluation split of both dataset versions, showing that the mode switch is used during navigation rather than only defined in the control rules. On both versions, val-unseen and test-unseen have more returns per 100 episodes than val-seen. Thus, precision-to-cruise transitions occur more frequently in evaluation trajectories from unseen scenes. Figure 4 provides two trajectory examples of mode switching to complement the return-event counts in Table 3.

5 Conclusion

We presented RCPNav for aerial vision-and-language navigation in urban environments. Its reversible cruise–precision controller enters precision mode when the goal appears close and returns to cruise if an updated prediction places it beyond the local search range. Selective historical keyframes retain visual evidence from earlier viewpoints, while a language-grounded landmark graph represents spatial connections between named places. Together, they provide information to guide navigation. Our model achieves state-of-the-art performance on both the original and refined CityNav benchmarks, demonstrating the effectiveness of RCPNav on challenging navigation tasks. Ablation studies confirm the contribution of each component.

Limitations and Future Work. RCPNav relies on existing annotations of named landmarks and estimates goal distance from predicted progress. Missing landmark information or inaccurate progress predictions near the goal may affect mode switching and stopping. Future work will explore updating landmark information during flight and improving the reliability of near-goal distance estimates and stopping decisions.

AI use statement

We used generative AI tools to aid or polish the writing, retrieve and discover related work, and draft sections of the paper. The authors reviewed and revised the AI-assisted text and take responsibility for the final manuscript.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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. doi: 10.18653/v1/2025.emnlp-main.338.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. TOUCHDOWN: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16537–16547, 2022.

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.

Xichen Ding, Jianzhe Gao, Cong Pan, Wenguan Wang, and Jie Qin. History-enhanced two-stage transformer for aerial vision-and-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18225–18233, 2026. doi: 10.1609/aaai.v40i22.38885.

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*, pp. 10976–10985, 2026.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, et al. OpenFly: A comprehensive platform for aerial vision-language navigation. In *International Conference on Learning Representations*, 2026.

Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15460–15470, 2022.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

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.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2023.

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. arXiv: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 *Proceedings of the European Conference on Computer Vision*, 2020.

Jacob Krantz, Shurjo Banerjee, Wang Zhu, Jason Corso, Peter Anderson, Stefan Lee, and Jesse Thomason. Iterative vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14921–14930, 2023.

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*, pp. 4392–4412, 2020. doi: 10.18653/v1/2020.emnlp-main.356.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Rui Liu, Xiaohan Wang, Wenguan Wang, and Yi Yang. Bird's-eye-view scene graph for vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10968–10980, 2023a.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. AerialVLN: Vision-and-language navigation for UAVs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023b.

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 *International Conference on Learning Representations*, 2019a.

Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019b.

Taiki Miyanishi, Fumiya Kitamori, Shuhei Kurita, Jungdae Lee, Motoaki Kawanabe, and Nakamasa Inoue. CityRefer: Geography-aware 3D visual grounding dataset on city-scale point cloud data. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. REVERIE: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. HOP: History-and-order aware pre-training for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15418–15427, 2022.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635, 2011.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, et al. DINOv3, 2025. arXiv:2508.10104.

Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. DREAMWALKER: Mental planning for continuous vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10873–10883, 2023a.

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 *International Conference on Learning Representations*, 2025.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. GridMM: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15625–15636, 2023b.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12009–12020, 2023c.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. doi: 10.18653/v1/2025.acl-long.127.

Haotian Xu, Yue Hu, Chen Gao, Zhengqiu Zhu, Yong Zhao, Yong Li, and Quanjun Yin. GeoNav: Empowering MLLMs with dual-scale geospatial reasoning for language-goal aerial navigation. *Pattern Recognition*, 177:113365, 2026. doi: 10.1016/j.patcog.2026.113365.

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 (Volume 1: Long Papers)*, pp. 31292–31309, 2025. doi: 10.18653/v1/2025.acl-long.1511.