GEOMETRY-ALIGNED REWARDS FOR AERIAL VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Aerial vision-and-language navigation (AVLN) requires an agent to localize a language-described goal on a city-scale aerial map by grounding visual landmarks and reasoning over their spatial relations. Recent vision-language model (VLM)-based methods formulate goal localization as coordinate generation and optimize the policy with reinforcement learning. Their composite rewards typically calibrate goal accuracy with globally fixed distance scales, limiting discrimination beyond a cutoff and ignoring variation in landmark extent. To address this issue, we propose **GS-Nav**, a geometry-aligned learning framework for AVLN. Its core is a Scale-Adaptive Gaussian Reward, which converts the distance to the target into a dense preference signal and adjusts its bandwidth according to the spatial extent of the referred landmark. In addition, we identify a simple but effective resolution-decoupling strategy: training with high-resolution maps while performing inference at a lower resolution. On the CityNav benchmark, GS-Nav obtains the best reported SR, OSR, SPL, and mean NE point estimates among the listed learned methods on validation unseen. Meanwhile, resolution-decoupled inference uses approximately $3.1\times$ fewer visual tokens and reduces median query latency by about $4.2\times$. Controlled reward comparisons support geometry-aligned supervision within CityNav; resolution controls reproduce the inference trend at two scales of the Qwen2.5-VL family. Code and models will be released.

1 INTRODUCTION

Aerial vision-and-language navigation (AVLN) asks an unmanned aerial vehicle to reach a destination described through visual landmarks and spatial relations (Liu et al., 2023b; Lee et al., 2025). An instruction such as "behind the white warehouse and across the road from the parking lot" requires grounding multiple landmarks and inferring a target that may lie outside them. Recent VLM-based systems formulate this task as coordinate generation, followed by execution with a navigation controller (Cai et al., 2025). We study the geometric supervision used to learn these coordinates.

FlightGPT evaluates generated goals with a globally calibrated, truncated piecewise verifier (Cai et al., 2025). Its feedback is graded within its effective range, but predictions beyond the cutoff receive identical goal scores. If every coordinate in a rollout group lies beyond that range, the goal term contributes no relative signal. Moreover, the success radius, cutoff, and decay temperature are shared across instructions anchored to landmarks as different as a stadium and a bus stop. Figure 1 illustrates this lack of scale adaptation.

GS-Nav replaces the goal term with a Scale-Adaptive Gaussian Reward centered at the annotated target. Landmark width and height determine its anisotropic bandwidths. When several candidate boxes are available, their maximum selects a favorable scale, while retaining the same target center. A separate Gaussian-coverage term supervises the landmark box generated in the reasoning trace. This yields geometry-aligned supervision for both the final coordinate and intermediate grounding.

We also separate training resolution from inference resolution. For the 4k-trained policy, 2k inference improves validation-unseen SR while shortening the visual sequence; the trend recurs for the 3B Qwen2.5-VL model and an SFT control. Inference uses one autoregressive generation call per planning round and the fixed controller (Figure 2).

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.

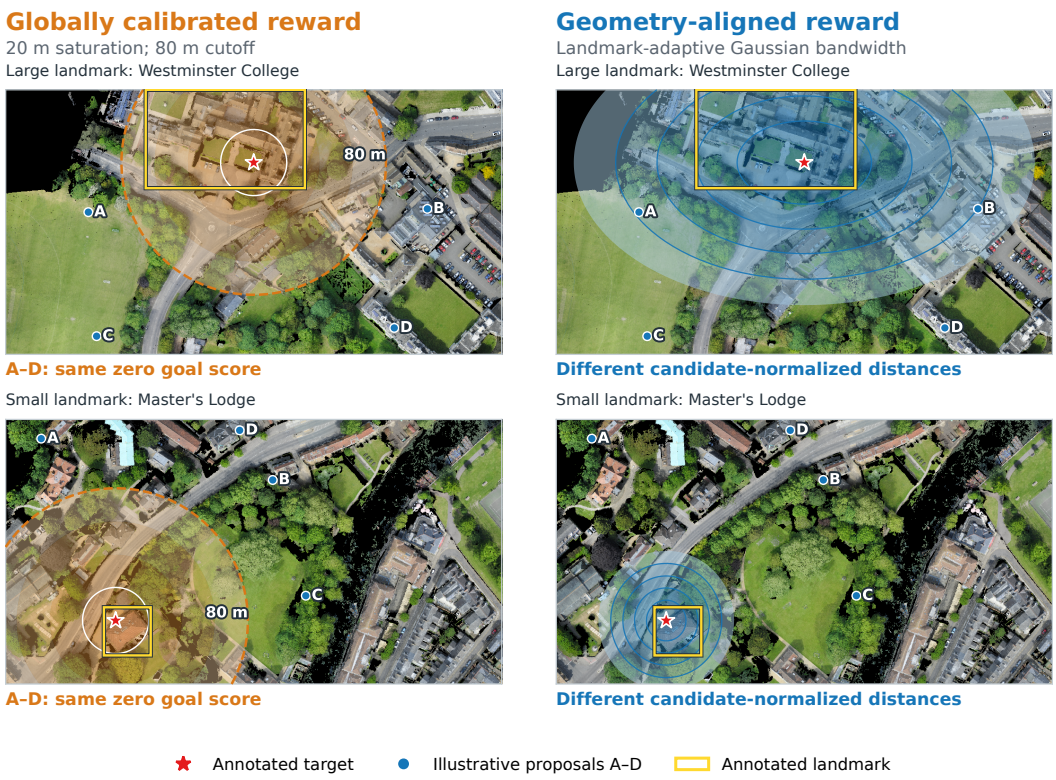

Figure 1: Goal-reward geometry on the CityNav scenes used in Figure 4. Each row shares the photograph, target, landmark box, and illustrative proposals A–D. Left: fixed distance regimes and cutoff. Right: Gaussian contours with landmark-adaptive bandwidths. Proposals and overlays illustrate the two verifiers.

We evaluate GS-Nav on CityNav (Lee et al., 2025). GS-Nav obtains the best reported point estimates for SR, OSR, SPL, and mean NE among the listed learned methods on validation unseen. In the controlled $4k \rightarrow 2k$ comparison, replacing the globally calibrated FlightGPT composite verifier with our full reward improves SR from 19.64 to 20.56 and reduces mean NE from $60.69\,\text{m}$ to $59.14\,\text{m}$. Resolution-decoupled inference reduces visual tokens from $16{,}384$ to $5{,}329$ and measured median query latency from $27{,}424\,\text{ms}$ to $6{,}602\,\text{ms}$. Reward controls, error summaries, resolution sweeps, representation similarity, and aggregate attention statistics characterize these results.

The main contributions of this work are:

- We identify a calibration limitation in the globally parameterized, truncated piecewise goal verifier used for VLM-based AVLN: its distance discrimination ends at a fixed cutoff and does not adapt to referred-landmark geometry.
- We propose the Scale-Adaptive Gaussian Reward, which has no explicit distance cutoff, adapts anisotropic tolerance to candidate landmark extent, and pairs target supervision with Gaussian landmark-box coverage.
- We document resolution-sensitive inference for VLM-based aerial localization. For the 4k-trained Qwen2.5-VL policy on CityNav, 2k inference improves validation-unseen SR while using approximately $3.1\times$ fewer visual tokens and $4.2\times$ lower measured median query latency.

2 Related Work

Vision-and-language navigation. VLN combines language grounding with embodied navigation. Indoor benchmarks include R2R (Anderson et al., 2018) and REVERIE (Qi et al., 2020); methods

have explored panoramic action spaces (Fried et al., 2018), transformer pretraining (Hong et al., 2021; Chen et al., 2021), and topological memory (Chen et al., 2022). LLM/VLM policies add pretrained knowledge and explicit reasoning (Zhou et al., 2024; Zhang et al., 2024). AerialVLN (Liu et al., 2023b) extends instruction following to aerial scenes, while CityNav (Lee et al., 2025) studies top-down urban goal localization. HETT (Ding et al., 2026) and HTNav (Fan et al., 2026) use structured navigation architectures. FlightGPT (Cai et al., 2025) instead generates destination coordinates with a VLM and optimizes a verifiable composite reward. GS-Nav studies the geometry of this supervision while retaining the coordinate-to-controller interface.

Rewards for navigation and spatial grounding. Potential-based reward shaping (Ng et al., 1999) and progress supervision (Ma et al., 2019) provide intermediate learning signals, while RL with verifiable rewards scores structured outputs programmatically (Shao et al., 2024; Guo et al., 2025). Structured generation and reasoning also support spatial tasks (Wei et al., 2022; Liu et al., 2023a; Chen et al., 2024; Bai et al., 2025; Gou et al., 2025). FlightGPT combines graded goal feedback within fixed distance thresholds with landmark and format rewards. GUI-G^2 (Tang et al., 2026) uses related Gaussian supervision for clickable regions. In AVLN, targets may lie outside referred landmarks and share several candidate boxes. GS-Nav therefore centers its point reward at the annotated destination, uses candidates to set bandwidths, and independently scores predicted landmark coverage.

3 METHOD

3.1 PROBLEM FORMULATION

Let $I \in \mathbb{R}^{H\times W\times 3}$ denote a top-down aerial map containing the agent's initial pose, and let L be a natural-language instruction that describes a destination through spatial relations to one or more landmarks. Each training example provides a ground-truth target point

$$p^* = (x^*, y^*)$$

and K valid landmark candidates

$$\mathcal{B} = \{(c^{(k)}, w^{(k)}, h^{(k)})\}_{k=1}^{K},$$

where $c^{(k)}$, $w^{(k)}$, and $h^{(k)}$ are the center, width, and height of the k-th annotated landmark box. All candidates share the destination p^* and provide landmark scales for the point reward and reference boxes for intermediate-grounding supervision.

We model the agent as a VLM policy π_θ. Conditioned on the map and instruction, the policy autoregressively generates a reasoning trace followed by a structured target coordinate:

$$o \sim \pi_\theta(\cdot \mid I, L), \qquad o = [\tau; \texttt{</think>}; \langle \hat{x}, \hat{y} \rangle]. \tag{1}$$

The reasoning trace τ contains a predicted landmark box $\hat{b}$, and the terminal output provides the predicted target $\hat{p} = (\hat{x}, \hat{y})$. The predicted coordinate is mapped to the global metric frame and supplied as the goal of a fixed navigation controller. The learned policy is therefore responsible for grounding the instruction and localizing the destination, while trajectory execution is shared by all methods evaluated under our protocol.

3.2 FRAMEWORK OVERVIEW

Figure 2 summarizes GS-Nav. During post-training, the VLM receives a map–instruction pair and generates both an intermediate landmark box and a final target coordinate. The proposed reward evaluates these two outputs jointly: a scale-adaptive point term supervises destination localization, while a Gaussian-coverage term supervises landmark grounding. At inference, the resulting policy predicts a destination from a lower-resolution map, and the fixed controller converts that destination into executable actions. This separation lets the learning objective focus on visual–language grounding without changing the downstream navigation procedure.

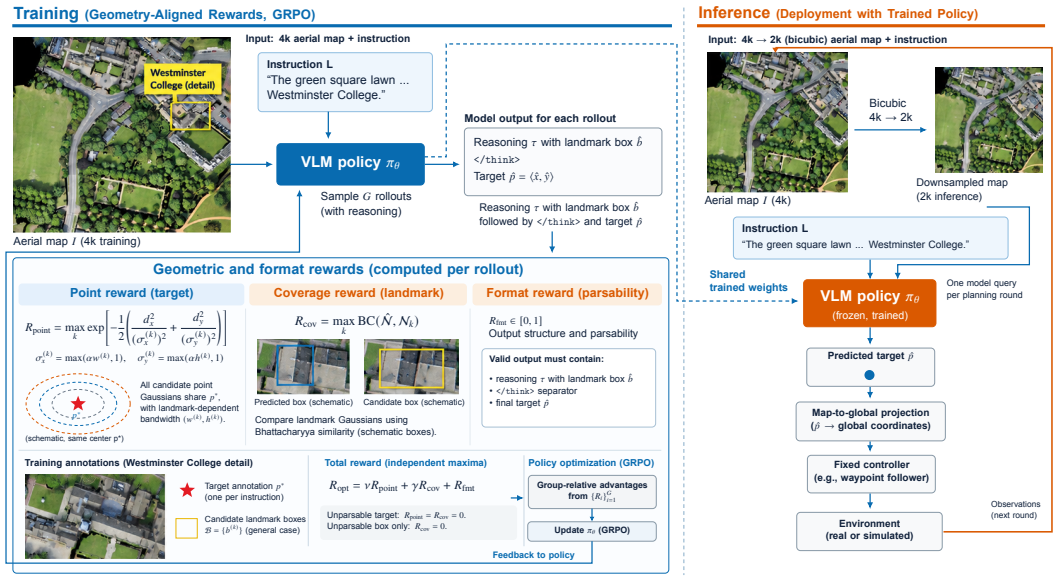

Figure 2: GS-Nav training and inference. The training-only verifier combines scale-adaptive point, Gaussian landmark-coverage, and format rewards, with independent candidate maxima. At inference, the same trained policy receives a 2k map and supplies a goal to the fixed controller through coordinate projection.

3.3 Scale-Adaptive Gaussian Reward

3.3.1 From Global Calibration to Geometry-Aligned Supervision

FlightGPT evaluates final goal localization, intermediate landmark grounding, and output-format compliance (Cai et al., 2025). Its piecewise goal verifier saturates within 20 m, decays exponentially until an 80 m cutoff, and is zero thereafter. The success radius, cutoff, and decay temperature are fixed across landmarks of different extents. GS-Nav adapts this spatial calibration to landmark geometry.

GS-Nav retains the composite-reward structure while reformulating its geometric supervision. The final target is evaluated by a Gaussian point verifier with no explicit distance cutoff and with bandwidth determined by landmark scale; the intermediate landmark box is evaluated by smooth Gaussian coverage. Both terms are candidate-aware and computed from the structured model output. We retain a format-validity reward, following prior structured-output training (Cai et al., 2025), solely to enforce a parsable reasoning–answer format.

3.3.2 Scale-Adaptive Point Reward

Let $d_x = \hat{x} - x^*$ and $d_y = \hat{y} - y^*$. For candidate k, we define anisotropic bandwidths

$$\sigma_x^{(k)} = \max(\alpha w^{(k)}, 1), \qquad \sigma_y^{(k)} = \max(\alpha h^{(k)}, 1), \tag{2}$$

where $\alpha > 0$ controls the overall tolerance and the one-pixel floor prevents degenerate bandwidths. Each candidate defines an unnormalized axis-aligned Gaussian with a different bandwidth but the same center p^*. We aggregate candidates by selecting the highest candidate-conditioned score:

$$R_{\text{point}} = \max_k \exp\left[-\frac{1}{2}\left(\frac{d_x^2}{(\sigma_x^{(k)})^2} + \frac{d_y^2}{(\sigma_y^{(k)})^2}\right)\right]. \tag{3}$$

The anisotropic bandwidth yields broader tolerance along a large landmark's extent and finer discrimination around a small landmark. With all Gaussians centered at p^*, the maximum selects the most favorable candidate-dependent bandwidth. The reward decays continuously with normalized distance; its numerically effective range depends on bandwidth and floating-point precision. The coverage term below evaluates candidate-box agreement.

3.3.3 LANDMARK-COVERAGE REWARD

The coverage term supervises the landmark box in the reasoning trace. Each candidate box is represented as an axis-aligned Gaussian with mean $c^{(k)}$ and covariance

$$\Sigma^{(k)} = \mathrm{diag}\left((\alpha w^{(k)})^2, (\alpha h^{(k)})^2\right).$$

Writing the parsed predicted box as $\hat{b} = (\hat{c}, \hat{w}, \hat{h})$, we represent it analogously with $\hat{\Sigma} = \mathrm{diag}((\alpha\hat{w})^2, (\alpha\hat{h})^2)$. Its similarity to candidate k is measured by the Bhattacharyya coefficient, and the best-matching candidate determines the reward:

$$R_{\mathrm{cov}} = \max_k \int_{\mathbb{R}^2} \sqrt{\mathcal{N}(u; \hat{c}, \hat{\Sigma})\mathcal{N}(u; c^{(k)}, \Sigma^{(k)})}\, du. \tag{4}$$

The coverage maximum selects the annotated box that best matches the predicted landmark center and extent. If no landmark box can be parsed from the reasoning trace, we set $R_{\mathrm{cov}} = 0$.

3.3.4 TOTAL REWARD

The complete optimization reward combines geometric supervision with format validity:

$$R_{\mathrm{opt}} = \nu R_{\mathrm{point}} + \gamma R_{\mathrm{cov}} + R_{\mathrm{fmt}},$$

where $R_{\mathrm{fmt}} \in [0, 1]$ checks the required reasoning–answer structure and output parsability. We use $\nu = \gamma = 1$ in all experiments. The maxima in Eqs. (3) and (4) are computed independently: one candidate may provide the most suitable scale for target localization, while another may best match the generated landmark box. An unparsable target or an empty candidate set yields zero geometric reward; an unparsable landmark box yields $R_{\mathrm{cov}} = 0$.

3.4 RESOLUTION-DECOUPLED INFERENCE

High-resolution maps preserve fine-grained landmark details during training. For the 4k-trained policy, localization accuracy varies non-monotonically with inference resolution and peaks at a moderately downsampled input. GS-Nav trains on 4k maps and performs inference on bicubically downsampled 2k maps, reducing visual tokens and inference cost. Each planning round uses one autoregressive generation call followed by the fixed navigation controller.

4 EXPERIMENTS

Our evaluation comprises three aspects: comparison with existing AVLN methods; controlled analysis of reward support, globally calibrated composite verification, and geometry-aware calibration; and investigation of inference-resolution effects across policies trained at different resolutions and with different objectives.

4.1 EXPERIMENTAL SETUP

Dataset and metrics. We conduct experiments on CityNav (Lee et al., 2025), a city-scale AVLN benchmark containing $4{,}755$ training instructions and three evaluation splits over 34 disjoint city blocks. Validation seen evaluates navigation in familiar environments, whereas validation unseen and test unseen measure generalization to novel urban scenes. Following prior work (Lee et al., 2025; Cai et al., 2025), we report navigation error (NE), success rate within $20\,\mathrm{m}$ (SR), oracle success rate (OSR), and success weighted by path length (SPL).

Protocol and baselines. Metrics are computed from executed trajectories. Each episode contains an initial planning round and one intermediate re-plan. At each round, the model greedily generates a target coordinate for the shared non-learned lookahead controller. Table 1 compares Random and Human references, specialized navigation methods (Lee et al., 2025; Liu et al., 2023b; Ding et al., 2026; Fan et al., 2026), and FlightGPT (Cai et al., 2025) under their standard protocols. FlightGPT-RL (standard) uses matched training and inference resolutions. The controlled verifier

Table 1: Overall system comparison on CityNav using the original annotations. Prior systems are shown under their standard reported or reproduced protocols; FlightGPT-RL (standard) denotes the matched-resolution reproduction and is distinct from the 4k→2k controlled row in Table 2. NE is the mean; Human is a reference; best point estimates among learned methods are in **bold**.

	Validation Seen				Validation Unseen				Test Unseen			
Method	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑
Random	222.3	0.00	1.15	0.00	223.0	0.00	0.90	0.00	208.8	0.00	1.44	0.00
Human	9.1	89.31	96.40	60.17	9.4	88.39	95.54	62.66	9.8	87.86	95.29	57.04
Seq2Seq (Lee et al., 2025)	257.1	1.81	7.89	1.58	317.4	0.79	8.82	0.61	245.3	1.50	8.34	1.30
CMA (Lee et al., 2025)	240.8	0.95	9.42	0.92	268.8	0.65	7.86	0.63	252.6	0.82	9.70	0.79
AerialVLN (Liu et al., 2023b)	65.6	9.77	23.77	8.64	81.8	6.79	17.91	5.73	**64.1**	8.09	19.13	5.91
MGP (Lee et al., 2025)	53.0	16.93	29.90	14.38	73.8	8.35	17.91	7.07	86.1	10.90	20.24	9.94
FlightGPT (SFT+RL) (Cai et al., 2025)	66.1	17.57	30.26	15.78	68.1	14.69	29.33	13.24	76.2	21.20	35.38	19.24
FlightGPT-SFT (standard)	97.6	10.29	18.45	9.46	101.7	10.51	18.54	9.70	117.4	11.20	21.24	10.78
FlightGPT-RL (standard)	74.90	13.27	27.13	12.59	71.40	12.87	25.82	12.27	76.50	19.80	32.26	18.91
HTNav (Fan et al., 2026)	47.2	**28.30**	47.92	**25.07**	59.8	15.85	34.43	13.55	68.5	22.23	**41.02**	19.63
HETT (Ding et al., 2026)	**45.2**	25.16	**48.40**	23.01	62.1	17.48	25.09	14.46	72.9	22.97	39.30	17.01
GS-Nav (Ours, 4k→2k)	59.08	17.97	32.83	16.35	**59.14**	**20.56**	**35.56**	**18.95**	64.94	**23.82**	37.94	**21.78**

study in Table 2 uses 4k training and 2k inference throughout, including the FlightGPT composite + Res.-Dec. row.

Implementation. GS-Nav uses Qwen2.5-VL-7B-Instruct (Bai et al., 2025) with a frozen vision encoder and rank-64 LoRA on the language model. We optimize with GRPO (Shao et al., 2024) for one epoch on nominal 4k maps, using group size 4, learning rate 10^{-5}, $\alpha = 0.5$, and $\nu = \gamma = 1$. The controlled reward study holds the base model, initialization, data, optimizer, group size, and inference protocol fixed. Inference uses 2k maps unless specified otherwise. Optimization and compute details appear in the Supplementary Material.

4.2 Overall Results

Overall navigation performance. Table 1 compares systems under their standard evaluation protocols. On validation unseen, GS-Nav achieves the best NE, SR, OSR, and SPL point estimates among the listed learned methods, improving SR by 3.08 percentage points and SPL by 4.49 points over the strongest competing results and reducing NE by 0.66 m. On test unseen, it achieves the highest SR and SPL, with a 2.15-point SPL gain; HTNav achieves the highest OSR and AerialVLN the lowest NE. HTNav and HETT achieve higher SR in familiar environments, with validation-seen scores of 28.30% and 25.16%, respectively, compared with 17.97% for GS-Nav. On validation unseen, their SR decreases to 15.85% and 17.48%, while GS-Nav reaches 20.56% and maintains similar mean NE across the two validation splits (59.08 m and 59.14 m). This change in ranking highlights GS-Nav's strongest performance in previously unseen urban environments.

Comparison with VLM-based training. GS-Nav improves SR and SPL over the standard FlightGPT variants on both unseen splits. Table 2 compares reward designs under the shared 4k→2k protocol.

4.3 Ablation Studies

4.3.1 Reward ablation.

Table 2 compares binary success supervision, distance-based rewards, the FlightGPT composite verifier, and Gaussian rewards under a shared training and inference protocol. The full GS-Nav reward combines landmark-adaptive bandwidth with Gaussian landmark coverage and achieves the best SR, SPL, mean NE, and median NE in this comparison.

Error-distribution analysis. The full reward achieves a median NE of 40.74 m, compared with 42.50 m for the FlightGPT composite and 41.52 m for the fixed Gaussian. Together with the low-

Table 2: Controlled verifier comparison on CityNav validation unseen using the original annotations (n=2,826). All rows share the base model, initialization, RL recipe, and 4k→2k resolution-decoupling protocol. Success-only uses a binary goal reward based on the 20 m success threshold. FlightGPT composite + Res.-Dec. applies FlightGPT's globally calibrated truncated goal term, IoU landmark term, and format term (Cai et al., 2025) under 2k inference. Dense indicates graded goal feedback within its effective range; Adapt. indicates landmark-adaptive bandwidth; Landmark names the intermediate grounding score. Med. denotes median navigation error.

Reward	Dense	Adapt.	Landmark	SR↑	SPL↑	NE↓	Med.↓
Success-only	✗	–	–	13.62	12.31	89.01	64.37
Distance ($-d$)	✓	–	–	18.29	16.79	60.66	41.24
FlightGPT comp. + Res.-Dec.	✓	✗	IoU	19.64	17.99	60.69	42.50
Fixed Gaussian	✓	✗	–	20.10	18.21	59.33	41.52
GS-Nav (full)	✓	✓	Gaussian	**20.56**	**18.95**	**59.14**	**40.74**

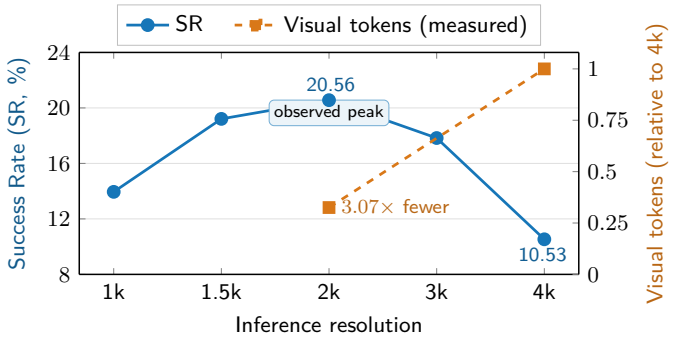

Figure 3: Measured inference-resolution sweep for the 4k-trained model on CityNav validation unseen. The SR point estimate peaks at 2k; visual-token counts are measured at 2k and 4k.

Table 3: Training and inference resolutions on CityNav validation unseen. Bold denotes the best result within each model and training-objective group.

Variant	Train	Infer	SR↑	OSR↑	SPL↑	NE↓
Ours, 7B	2k	2k	20.28	35.33	18.82	63.13
Ours, 7B	2k	4k	7.97	21.39	7.22	73.96
Ours, 7B	4k	4k	10.53	26.25	9.70	60.80
Ours, 7B	4k	2k	**20.56**	**35.56**	**18.95**	**59.14**
Ours, 3B	4k	4k	9.64	25.29	8.59	73.39
Ours, 3B	4k	2k	**13.46**	**28.77**	**12.08**	**68.97**
SFT, 7B	4k	4k	10.51	18.54	9.70	101.70
SFT, 7B	4k	2k	**11.70**	**20.50**	**10.60**	**95.04**

est mean NE of 59.14 m, these results show improved localization accuracy in both the mean and median.

4.3.2 Resolution decoupling.

Figure 3 shows a non-monotonic relationship between inference resolution and navigation accuracy for the 4k-trained model, with the highest SR at 2k. Table 3 shows the same 2k-over-4k inference trend for the 4k-trained 3B model and SFT control. For the 7B model trained at 2k, 2k inference yields higher SR (20.28% vs. 7.97%) and lower NE (63.13 m vs. 73.96 m) than 4k inference. At fixed 2k inference, 4k training improves SR from 20.28% to 20.56% and reduces NE from 63.13 m to 59.14 m. Lower-resolution inference also shortens the visual sequence while retaining one gener-

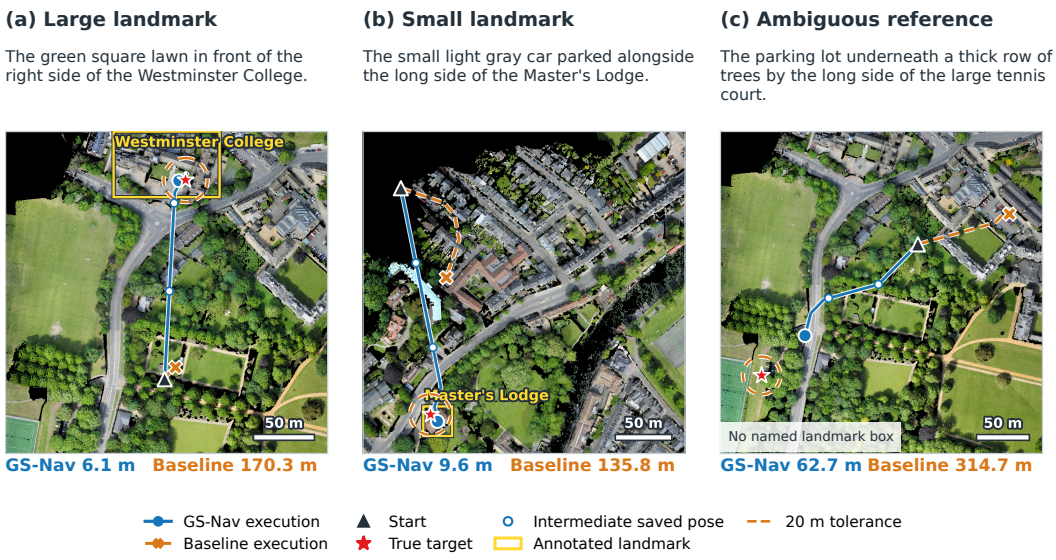

Figure 4: CityNav medium cases with recorded execution trajectories. Blue/orange paths and filled endpoints denote GS-Nav/the comparison run; triangles mark shared starts, hollow circles mark sampled saved poses, red stars mark targets, and yellow boxes mark annotated landmarks.

Table 4: Linear CKA of spatially aligned visual-token representations on 12 maps. Columns compare the same map across resolutions and different scenes at a common resolution.

Stage	Same image 4k–2k	Different scenes 2k–2k	Different scenes 4k–4k
Projector	0.163	0.796	0.889
Layer 14	0.195	0.722	0.840
Layer 27	0.082	0.429	0.558

ation call per planning round. With Transformers greedy generation on one NVIDIA H20, median query latency decreases from 27.424 s at 4k to 6.602 s at 2k.

Representation analysis. We measure linear centered kernel alignment (CKA) (Kornblith et al., 2019) on 12 maps after bilinearly aligning visual-token grids. Table 4 shows lower similarity for the same map across 4k and 2k than for different scenes at a common resolution. At layer 27, the corresponding values are 0.082, 0.429, and 0.558. Nearby-resolution comparisons further reveal sensitivity to token-grid geometry: 4k and 3.8k share a 128×128 grid and yield layer-27 CKA of 0.389, whereas 2k and 1.9k use 73×73 and 69×69 grids and yield 0.097. In these comparisons, a shared token grid is associated with greater representation similarity.

Token compression. On CityNav validation-unseen episodes, we compare visual-token compression using a shared checkpoint and Transformers greedy generation, with trained weights fixed. Spatial mean pooling reduces the mean token count from 16,310.8 to 8,179.4, increases SR from 10.90% to 12.61%, and increases SPL from 9.91% to 11.46%. The Supplementary Material provides the full compression comparison, attention statistics, and diagnostic settings.

4.4 Trajectory Analysis

Figure 4 shows two successful approaches and one failure. GS-Nav reaches the lawn near Westminster College and the car beside the Master's Lodge with final errors of 6.1 m and 9.6 m, respectively. The corresponding path lengths are 1.10 and 1.05 times the initial straight-line distances, indicating largely direct routes. In the parking-lot case, GS-Nav reduces the target distance from 217.2 m to 62.7 m but stops outside the 20 m success region. The saved comparison trajectories end 170.3 m,

135.8 m, and 314.7 m from the targets. The Supplementary Material provides case details and trajectory statistics.

5 LIMITATIONS

Our experiments evaluate CityNav with 3B and 7B Qwen2.5-VL models. Additional environments and model families would help establish how the benefits vary with scene layout, landmark distribution, and backbone architecture. The interaction among inference resolution, visual-token grids, and spatial representations remains a direction for further study, particularly across landmark scales and training resolutions.

6 CONCLUSION

We presented GS-Nav, a geometry-aligned learning framework for aerial vision-and-language navigation. Its Scale-Adaptive Gaussian Reward calibrates target supervision to landmark extent and complements it with Gaussian landmark-coverage supervision. On CityNav, GS-Nav achieves the highest SR and SPL among the compared learned methods on both unseen splits, with controlled reward comparisons demonstrating gains over the FlightGPT composite verifier. Resolution-decoupled inference combines 4k training with 2k inference to improve navigation accuracy while using approximately $3.1\times$ fewer visual tokens and reducing median query latency by approximately $4.2\times$. Together, these results demonstrate the value of geometry-aligned supervision and inference-resolution selection for accurate and efficient aerial navigation.

AI USE STATEMENT

We used generative AI tools, primarily GPT-6, to assist with manuscript writing and figure editing. The authors carefully reviewed all AI-generated and AI-edited text and figures. The authors take responsibility for the final text, figures, and scientific content.

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*, pp. 3674–3683, 2018.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *ArXiv*, abs/2502.13923, 2025. URL https://api.semanticscholar.org/CorpusID:276449796.

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. 6670–6687, 2025.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

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.

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.

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.

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.

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.

Boyu Gou, Demi Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 30851–30883, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

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*, pp. 5912–5922, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 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*, pp. 15384–15394, 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. *arXiv preprint arXiv:1901.03035*, 2019.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

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*, pp. 9982–9991, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang, Wenqi Zhang, Yongliang Shen, Weiming Lu, et al. Gui-g^2: Gaussian reward modeling for gui grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33214–33222, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaviD: Video-based VLM plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024.