

Air-Ground Collaborative Vision-and-Language Navigation via Shared Bird's-Eye Maps

Shuning Zhang, Liang Li, Yunheng Wang, Tao Wang, Yihang Kang, Renjing Xu

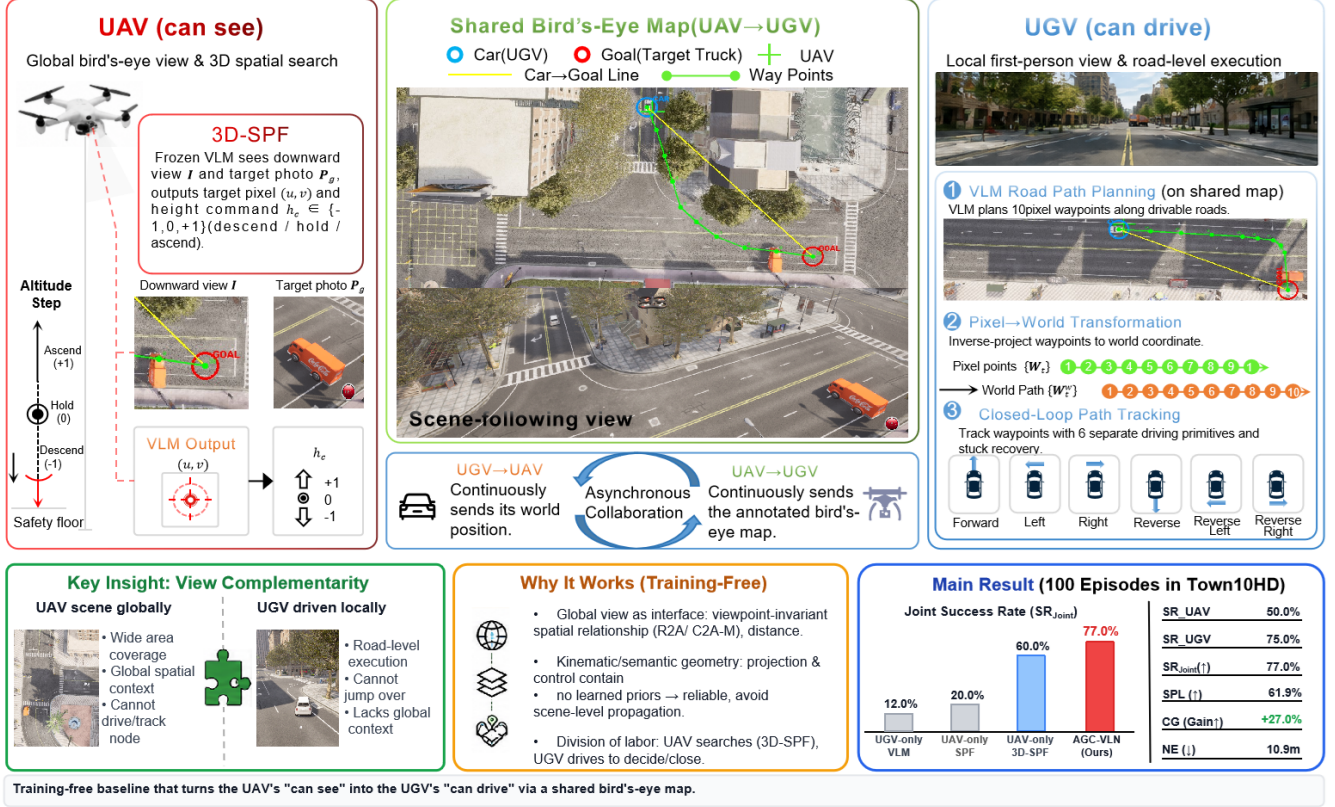


Fig. 1. The UAV turns its “can see” into the UGV’s “can drive”. View asymmetry: the UAV sees the whole block from a global bird’s-eye view while the UGV drives with a local first-person view. Shared map: the UAV renders the UGV pose and the target’s position as CAR/GOAL markers into a shared bird’s-eye map. Parallel execution: the UAV flies 3D-SPF toward the target while the UGV follows a VLM road path on the map, and either arriving counts as success, a training-free air-ground collaboration.

Abstract—Air-ground collaborative Vision-and-Language Navigation (VLN) pairs an unmanned aerial vehicle (UAV) with a global bird’s-eye view and an unmanned ground vehicle (UGV) with a local first-person view, yet the setting remains largely unexplored: existing training-free methods solve single-agent tasks but offer no collaboration mechanism, and a recent CARLA-Air evaluation found no stable cooperative behavior across five state-of-the-art VLA models; naive semantic communication or bidirectional coupling even *degrades* performance. We establish AGC-VLN (Air-Ground Collaborative VLN), the first training-free baseline for air-

ground collaborative VLN. The key insight is that training-free methods decompose navigation into VLM-based semantic reasoning and deterministic geometric execution, exposing a collaboration interface: the UAV’s global view, over which it renders the UGV’s reported pose and the target’s position as CAR/GOAL markers with distance labels, yielding a shared bird’s-eye map. From this map, the UGV acquires global spatial context its first-person view cannot provide, plans a road-following path with a frozen VLM, and executes it under closed-loop control; in parallel, the UAV runs 3D-SPF, a spatial-search upgrade of SPF that localizes the target in the downward view and flies toward it. On 100 closed-loop episodes in CARLA-Air’s Town10HD scene, AGC-VLN reaches a 77.0% joint success rate, a collaboration gain of +27.0% over the weaker individual agent (the UAV, 50.0%), and exceeds the strongest published single-agent baseline (Travel UAV, 53.0%) by 24.0 points, stemming from the complementarity of the UAV’s global view and the UGV’s road-following execution. Project page: <https://github.com/ZSN2024/AGC-VLN>.

Shuning Zhang and Liang Li have the same contributions.

Shuning Zhang, Yunheng Wang, Tao Wang, Yihang Kang and Renjing Xu are with the Robotics and Autonomous Systems Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), China
szhang272@connect.hkust-gz.edu.cn,
yunhengwang1214@gmail.com, 19819824568@163.com,
KYH2107405729@163.com, renjingxu@hkust-gz.edu.cn

Liang Li is with the School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, China
lianglee2004@163.com

I. INTRODUCTION

Frozen vision-language models have rewritten the recipe for Vision-and-Language Navigation (VLN). The classic benchmarks (R2R [1], REVERIE [2]) belonged to an era of training a dedicated policy per dataset. See-Point-Fly (SPF) [3] already flies a real drone to 92.7% success by merely asking a frozen VLM to “point” at the image, with no training at all. Uni-LaViRA [4], Fly0 [5], and FineCog-Nav [6] carry the same recipe across embodiments and reasoning styles. One assumption, however, survives intact from the classic era: *a single robot navigates alone*.

Multi-robot collaborative navigation is the natural next step, and heterogeneous **air-ground teams** are its most representative form. Search-and-rescue, last-mile delivery, and escort need wide-area coverage from the air plus ground-level reachability from a wheeled platform; pairing the two promises a division of labor no single embodiment can replicate. The infrastructure has recently arrived: CARLA-Air [7] unifies CARLA [8] and AirSim [9] in a single Unreal Engine process with zero-latency sensor synchronization, and AirGroundBench [10] provides 115 closed-loop air-ground VLN episodes across 11 environments.

The CARLA-Air cooperation study [11] evaluated five aerial VLA models (AerialVLA, OpenFly, OpenUAV, SPF, AerialVLN) on closed-loop air-ground tasks, and the verdict was blunt: **none of them could turn single-agent skill into cooperative behavior**. Worse, naive communication *hurt*: UGV text hints *degraded* most models, bidirectional coupling amplified errors for all, and even oracle geometric cues did not close the gap. The study attributes this to missing partner-state anchoring, low-latency action coordination, and team-level objective alignment, while a rule-based controller showed the tasks are solvable.

In short: *the platform exists, the benchmark exists, the failure is diagnosed, but no working air-ground collaborative VLN method has been demonstrated*.

In this paper we establish **AGC-VLN** (Air-Ground Collaborative VLN), deliberately a **training-free baseline**. Our key insight is that the very decomposition that makes training-free methods work for a single agent also solves the collaboration problem. SPF-style pipelines separate *semantic reasoning* (a frozen VLM marks the target in the image) from *geometric execution* (deterministic projection and control). We keep this decomposition but extend the VLM’s action interface. Beyond the 2D point, the VLM also outputs a discrete height command (descend, hold, or ascend). This gives the UAV active search in the vertical dimension and upgrades SPF’s fixed-altitude planar point-fly into **three-dimensional spatial search**. This spatial-aware, 3D-action-interface refinement of SPF is our **3D-SPF**. The seam between the two stages is a metric quantity: the *viewpoint-invariant spatial annotation* in the UAV’s bird’s-eye view, where the UAV renders the teammate’s reported pose and the target’s position into a shared map the UGV’s first-person view could never obtain. Collaboration therefore requires no learned cross-agent representation at all: the UAV supplies

global spatial context, and the UGV performs road-level planning and execution on top of it, each in its own role.

Our contributions are:

- 1) **3D-SPF algorithm**: we upgrade SPF [3] by adding a discrete height command (descend/hold/ascend) to the VLM’s action interface, turning fixed-altitude planar point-fly into three-dimensional spatial search; relative to native UAV-only SPF, this lifts the UAV’s success rate from 20.0% to 60.0%, a $3\times$ gain, on our air-ground search task.
- 2) **AGC-VLN system**: we couple 3D-SPF on the UAV with VLM road-path planning on the UGV through a **shared bird’s-eye map** (the UAV renders the teammate’s reported pose and the target’s position), requiring no training or learned cross-agent representation. To the best of our knowledge, at the time of writing, no UAV+UGV collaboration VLN system with a positive collaboration gain has been reported in the open literature [11], and AGC-VLN reaches a **77.0%** joint success rate with a **+27.0%** collaboration gain.
- 3) **View-complementary collaboration mechanism**: we show that the UAV’s global bird’s-eye map fills the global spatial context missing from the UGV’s first-person view, lifting the UGV’s success rate from 12.0% (UGV-only VLM) to 75.0% (AGC-VLN), enabling road-level planning and closed-loop execution.

II. RELATED WORK

A. Foundation-Model Navigation Without Training

The training-free paradigm replaces task-specific policies with frozen foundation models orchestrated by deterministic glue. SPF [3] is the purest instance: a frozen VLM marks a waypoint pixel and a depth label per frame, and fixed geometry turns the annotation into a control command (no gradient ever flows). Fly0 [5] pushes the same semantic/geometric separation; Uni-LaViRA [4] drives four embodiments with a single language-vision-action layer; FineCog-Nav [6] splits reasoning into seven inspectable modules. The learned VLM-navigation line likewise replaces discrete action spaces with continuous reasoning: NavILA [12] lifts VLM features into 3D, MapGPT [13] grounds navigation in map-guided prompting, GOAT [14] generalizes goal specification to any object, and StreamVLN [15] streams history tokens for long horizons. Large vision-language-action models (ABot-N1 [16], LongNav-R1 [17], EvolveNav [18]) push toward one general navigation policy, but stay trained and single-agent; none define what two agents should *say to each other*, the question this paper answers.

B. Cooperation Between Robots

Simulation support for UAV+UGV teams arrived with CARLA-Air [7], which runs CARLA [8] and AirSim [9] in one Unreal Engine process with synchronized physics ticks ($\Delta t = 0$ ms). The follow-up study [11] asked whether aerial VLA models can cooperate, testing three coupling modes (none, UGV-to-UAV text hints (C1), bidirectional velocity coupling (C2)) and finding that every mode failed

to improve (and often degraded) single-agent success across five models; partner-state anchoring was isolated as the root deficit. AirGroundBench [10] supplies the evaluation material: 115 closed-loop episodes plus a four-level VQA taxonomy whose hardest level probes cross-view reasoning. On the method side, collaborative VLN exists only for *homogeneous* teams: two ground robots in CoNavBench [19] and two UAVs at different altitudes in AeroDuo [20], whose high/low-viewpoint division of labor echoes our view complementarity. OmniVLN [21] spans platforms but needs rotating LiDAR, and JanusVLN [22] decouples semantics from spatiality for cross-platform navigation. The aerial single-agent line has matured, from CityNav [23] on real-world data to CLOSER-VLN [24] and FSD-VLN [25] on long-horizon aerial VLN, but none couple an aerial observer to a ground executor. Complementary benchmarks probe the building blocks: SpatialUAV [26] (low-altitude collaboration), an integrated planning framework [27] (air-ground field deployment), and a survey [28] (aerial VLN).

III. METHOD

We formalize the task (Sec. III-A), describe the UAV’s global perception and map-rendering module (Sec. III-B) and the UGV’s map path-planning and closed-loop execution module (Sec. III-C), and give the two agents’ parallel collaboration loop (Sec. III-D).

A. Problem Formulation

An episode hands a two-robot team a static target (a truck) specified by a prompt description. The team consists of a UAV hovering at altitude h_U whose RGB camera faces straight down (intrinsic $\mathbf{K}_U$), and a UGV whose RGB camera faces forward (intrinsic $\mathbf{K}_G$). The episode succeeds if, before the time budget $T_{\max}$ expires, any team member arrives within ϵ of the target without collision:

$$\text{success} \iff \min \{ \|\mathbf{x}_U - \mathbf{x}_g\|_2, \|\mathbf{x}_G - \mathbf{x}_g\|_2 \} \leq \epsilon, \quad (1)$$

where $\mathbf{x}_U, \mathbf{x}_G$ are the UAV and UGV positions and $\mathbf{x}_g$ the target position. Both agents receive synchronized poses from the simulator; no map, no UAV depth sensor, and no training data are assumed. In this implementation $\epsilon = 5$ m and $T_{\max} = 180$ s.

B. UAV Module: Global Perception and Map Rendering

At each decision step (interval ~ 3 s, about 0.33 Hz) the UAV performs two functions: rendering the shared bird’s-eye map for the UGV, and running 3D-SPF for itself.

1) *Bird’s-Eye Annotation (Shared Map Rendering)*: The UAV reads the UGV’s reported pose $\mathbf{x}_G$ from the shared state and projects $\mathbf{x}_G$ onto the current downward image I_U^t ; the target object’s position is taken from the episode’s goal coordinate $\mathbf{x}_g$ and projected onto I_U^t . The UAV then renders the annotations (Fig. 2):

- a blue CAR marker: the UGV’s current position;
- a red GOAL marker: the target truck’s position;
- a green cross: the UAV itself (image center);

- a yellow line: the CAR→GOAL straight line (for reference only, not drivable);
- distance labels: UAV→GOAL and CAR→GOAL.

The fully annotated image constitutes a **shared bird’s-eye map**, transmitted to the UGV through shared memory together with the UAV pose at annotation time (used for deterministic pixel→world inverse projection).

2) *3D-SPF (Three-Dimensional SPF)*: In parallel, the UAV runs 3D-SPF to fly toward the target itself: a frozen VLM receives the downward view I_U^t , and returns structured JSON containing the target truck’s 2D pixel position (u, v) in the downward view and a discrete height command $h_c \in \{-1, 0, 1\}$ (descend/hold/ascend). The pixel position is converted to a world coordinate $\mathbf{p}^w$ via flat-ground ray casting under a planar-ground assumption, parameterized by the downward field of view θ and the altitude h :

$$\mathbf{p}^w = \mathbf{c}_U + \mathbf{R}(\psi) h \tan \frac{\theta}{2} \begin{bmatrix} -2(\frac{v}{H} - \frac{1}{2}) \\ 2(\frac{u}{W} - \frac{1}{2}) \end{bmatrix}, \quad (2)$$

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix},$$

where ψ is the yaw and (W, H) the image size. The UAV flies horizontally toward $\mathbf{p}^w$ with proportional velocity

$$\mathbf{v} = \min(v_{\max}, k_p \|\mathbf{p}^w - \mathbf{c}_U\|) \frac{\mathbf{p}^w - \mathbf{c}_U}{\|\mathbf{p}^w - \mathbf{c}_U\|}, \quad (3)$$

then, once stabilized, descends/ascends 10 m according to h_c (with a safety floor), then hovers awaiting the next decision.

3) *Coordinate Transformation*: The UAV and UGV live in different coordinate systems: AirSim uses NED (z pointing down), CARLA uses a world frame (z pointing up). The two are converted through a constant offset $\mathbf{o}$ obtained by one-time calibration:

$$\mathbf{x}_{\text{NED}} = \mathbf{o} + \mathbf{S} \mathbf{x}_{\text{CARLA}}, \quad \mathbf{S} = \text{diag}(1, 1, -1), \quad (4)$$

$$\mathbf{o} = (\text{ap}_x - \text{dl}_x, \text{ap}_y - \text{dl}_y, \text{ap}_z + \text{dl}_z)^\top,$$

where ap and dl denote the drone position reported by AirSim and CARLA, respectively, at calibration. Pixel↔world conversion is done by flat-ground ray casting (Eq. 2 and its inverse), echoing cross-view geo-localization, which aligns drone and overhead views [29], [30].

C. UGV Module: Path Planning and Closed-Loop Execution

Upon receiving the annotated bird’s-eye map, the UGV executes three steps:

1) *Road Path Planning*: A frozen VLM combines the pixel coordinates of CAR/GOAL in the map, and the image size to plan a path of 10 pixel waypoints $\{\mathbf{w}_i\}_{i=1}^{10}$ along visible roads. The first point must equal its own position and the last must equal the target position. The intermediate points follow the road direction, turn at intersections, and never cut through buildings or leave the road. Before returning, the path is sanity-checked, discarding hallucinated paths whose first/last points clearly deviate from CAR/GOAL.

2) *Pixel Path → World Path*: Using the UAV pose at annotation time, each pixel waypoint is inverse-projected to NED coordinates, then converted through the offset $\mathbf{o}$ to CARLA world coordinates, yielding the world path $\{\mathbf{w}_i^w\}$.

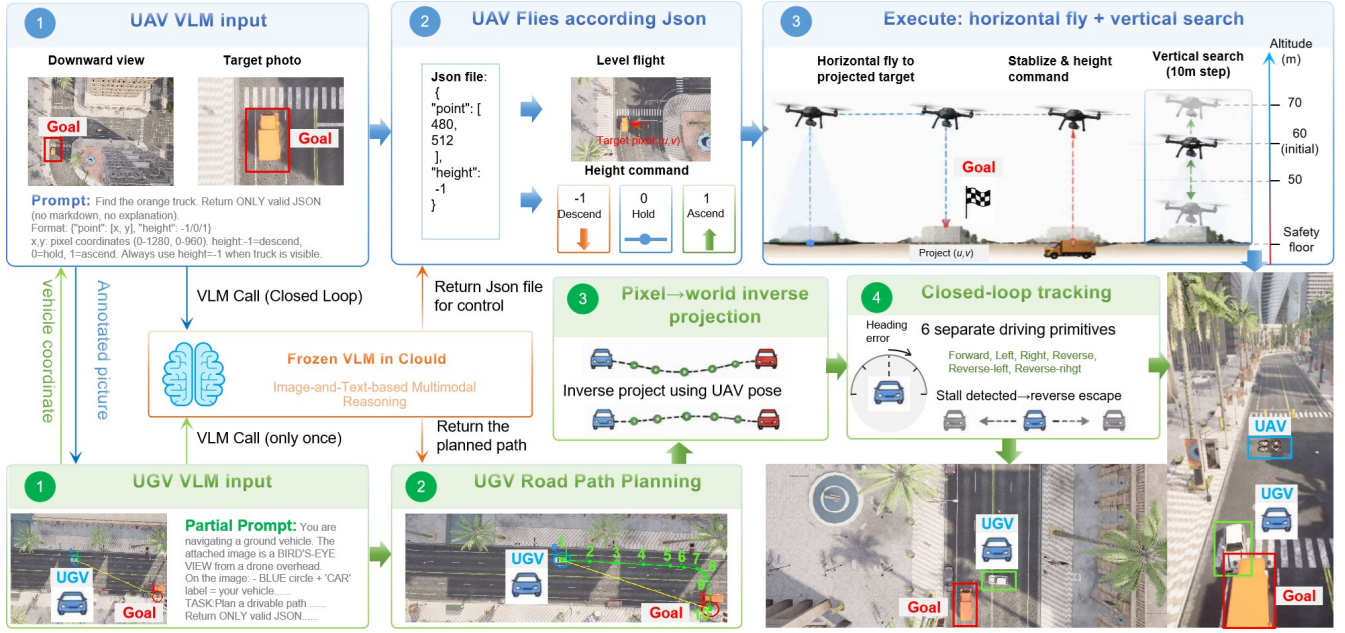


Fig. 2. Architecture of AGC-VLN. On the UAV side, 3D-SPF queries a frozen VLM in closed loop to localize the target in the downward view, returning a JSON point and a height command (descend/hold/ascend) that drive horizontal flight and 10m-step vertical search; it renders the UGV pose and the target’s position as CAR/GOAL markers into a shared bird’s-eye map. On the UGV side (right), a frozen VLM plans a road path on the map, which is inverse-projected to world coordinates and tracked by a closed-loop controller with six driving primitives and stuck recovery.

3) *Closed-Loop Path Tracking*: The UGV tracks the world path point by point with a closed-loop controller: at each tick it computes the wrapped heading error to the current waypoint and a distance-scaled throttle,

$$e_\psi = \text{wrap}_\pi(\psi_{\text{wp}} - \psi), \quad \tau = \text{clip}\left(\frac{d}{10}, 0.3, 1\right), \quad (5)$$

where ψ is the UGV heading, ψ_{wp} the heading toward the waypoint, and d the remaining distance. The error e_ψ is discretized into six driving primitives (forward, left, right, reverse, reverse-left, reverse-right) by angle thresholds. When the vehicle stalls beyond a threshold (throttle applied but near-zero displacement), a reverse escape is triggered. After reaching each waypoint, it checks whether it has entered the ϵ range of the target.

D. Collaboration Mechanism

Algorithm 1 summarizes the complete parallel loop. The two agents collaborate asynchronously through shared memory, exchanging two kinds of information:

- **UGV→UAV (pose)**: the UGV continuously reports its CARLA world coordinate, for the UAV to render the CAR marker;
- **UAV→UGV (shared bird’s-eye map)**: the UAV continuously outputs the annotated bird’s-eye image and pose at annotation time, for the UGV to plan a path.

The key to the collaboration is **view complementarity**. The UAV has a global bird’s-eye view but cannot drive along roads, while the UGV can drive along roads but has only a first-person local view. The annotated map delivers the UAV’s global spatial context (relative positions of teammate

and target, road topology) in an image form the UGV’s VLM can consume directly, thereby filling the UGV’s blind spot.

IV. EXPERIMENTS

We design experiments to answer three research questions:

- 1) **RQ1 (Feasibility)**: Can training-free single-agent methods be composed into a working air-ground collaborative VLN system, where trained VLA models failed [11]?
- 2) **RQ2 (Collaboration gain)**: Does the UAV-UGV team outperform each agent operating alone, and by how much?
- 3) **RQ3 (Failures)**: Where does the baseline still fail, and what do the failures imply for future learned components?

A. Experimental Setup

Testbed: simulation experiments run inside CARLA-Air [7], whose single-process design guarantees that UAV and UGV observations sample the same physics tick. The quadrotor carries a downward 1080p RGB camera (FOV 108°), hovering at 60 m; the UGV carries a forward camera; both read poses from the synchronized pose stream.

Episodes: 100 closed-loop episodes across 50 spawn points (2 runs each) in the Town10HD scene; a Mini Cooper (UGV) is spawned at the start and an HGV truck at the goal.

Metrics: we report the UAV success rate (SR_{UAV} , the UAV arrives within ϵ of the target), the UGV success rate (SR_{UGV}), and the joint success rate (SR_{joint} , either member arrives), with $\epsilon = 5$ m. We also report Success weighted by Path Length (SPL), Navigation Error (NE), and the **collaboration**

Algorithm 1 AGC-VLN: Training-Free Air-Ground Collaborative VLN (Bird’s-Eye Map Sharing)

Require: Goal position $\mathbf{x}_g$, budget $T_{\max}$

Ensure: Episode outcome $\in \{\text{success}, \text{failure}\}$

```

1: Start UAV and UGV threads in parallel, sharing state  $\mathcal{S}$ ;
   clear both success flags
2: while  $t < T_{\max}$  and not both agents succeeded do
3:   UAV thread: read  $\mathcal{S}.\text{ugv\_pos}$ ; write its own pose
4:    $(u, v, h_c) \leftarrow \text{VLM.Locate}(I_U^t)$  // target pixel
    $(u, v)$ ; height  $h_c \in \{-1, 0, 1\}$ 
5:    $M^t \leftarrow \text{Annotate}(I_U^t, \mathbf{x}_G, \mathbf{x}_g)$  // CAR + GOAL
   markers; GOAL at the target position  $\mathbf{x}_g$ 
6:    $\mathcal{S}.\text{map} \leftarrow (M^t, \text{pose}_t)$  // shared bird’s-eye map +
   pose at annotation time
7:   fly toward  $\mathbf{p}^w = \text{Project}(u, v)$ ; adjust altitude by  $h_c$ 
   // 3D-SPF flight
8:   if UAV within  $\epsilon$  of target: mark UAV success (keep
   annotating)
9:   UGV thread: read  $\mathcal{S}.\text{map}$ 
10:   $\{\mathbf{w}_i\}_{i=1}^{10} \leftarrow \text{VLM.PlanPath}(M^t)$  // road path
   planning
11:   $\{\mathbf{w}_i^w\} \leftarrow \text{InverseProject}(\{\mathbf{w}_i\}, \text{pose}_t)$ 
12:  FollowPath( $\{\mathbf{w}_i^w\}$ ) // closed-loop tracking + stuck
   recovery
13:  if UGV within  $\epsilon$  of target: mark UGV success (keep
   reporting pose)
14: end while
15: return success if either agent marked success, else
   failure

```

gain, $\text{CG} = \text{SR}_{\text{joint}} - \min(\text{SR}_{\text{UAV}}, \text{SR}_{\text{UGV}})$, which measures how much the team exceeds the weaker single agent.

Configuration: gemini-3.7-flash [31] (temperature 0.1) serves as the frozen VLM on both platforms; the UAV/UGV decision interval is 3 s (~ 0.33 Hz); the time budget is 180 s.

B. Baselines

(1) **Published single-agent aerial VLN methods:** OpenFly [32], FineCog-Nav [6], 3DG-VLN [33], and Travel UAV [34], reproduced under the same single-agent evaluation protocol for comparison.

(2) **Single-agent (collaboration lower bound):** UAV-only SPF, UAV-only 3D-SPF, and UGV-only VLM, each solving the whole episode alone.

C. Main Results (RQ1, RQ2)

Table I reports the main results. We measure the collaboration gain as the amount by which the joint success rate exceeds the weaker individual agent:

$$\text{CG} = \text{SR}_{\text{joint}} - \min(\text{SR}_{\text{UAV}}, \text{SR}_{\text{UGV}}). \quad (6)$$

Two findings follow: (1) **positive collaboration gain.** AGC-VLN attains $\text{SR}_{\text{joint}} = 77.0\%$ ($\text{SR}_{\text{UGV}} = 75.0\%$, $\text{SR}_{\text{UAV}} = 50.0\%$), a collaboration gain of $\text{CG} = +27.0\%$ over the

weaker agent, together with an SPL of 62.0% and an NE of 10.9 m; the joint rate thereby exceeds the strongest published single-agent baseline (Travel UAV, 53.0%) by 24.0 points, unlike the VLA baselines whose coupling degraded performance; (2) **success means either member arrives.** The task is judged successful if either agent reaches the target, so SR_{joint} is the *union* of the per-agent rates.

D. Ablation: VLM Backbone

Following SPF’s cross-VLM study, we swap the frozen VLM on both platforms across four backbones (gpt-5.6-luna, gemini-2.5-flash, gemini-3.7-flash, and qwen-vl-max [31]) and report success rates and costs (Table II). gemini-3.7-flash attains the highest joint success rate (77.0%) as the most balanced backbone, keeping both agents strong ($\text{SR}_{\text{UGV}} = 75.0\%$, $\text{SR}_{\text{UAV}} = 50.0\%$); the alternatives leave the UAV weaker ($\text{SR}_{\text{UAV}} = 7.0\%$ –47.0%), so their joint rates fall short (67.0%–73.0%).

E. Ablation: UAV Altitude

To test 3D-SPF’s sensitivity to the UAV’s initial altitude, we vary it across 30/60/90/120 m while keeping the UGV module and the height command (descend/hold/ascend) unchanged (Table III). 3D-SPF is most effective at 60 m, where both agents contribute: the UAV’s success rate peaks at 50.0% and the UGV’s road-following reaches 75.0%, yielding the best joint rate of 77.0%. At 30 m the narrowed field of view drops the UAV to 20.0% and the UGV to 14.0%, collapsing the joint rate to 28.0%; at 90/120 m the truck shrinks to a handful of pixels and UAV localization fails entirely (0.0%), so the high joint rate (77.0%/72.0%) is delivered by the UGV alone, with no aerial contribution.

F. Ablation: Map Annotation Richness

To test the necessity of the shared map’s annotation content, we degrade the bird’s-eye annotation stepwise through five levels of richness: full annotation (CAR+GOAL+distance labels+reference line), CAR+GOAL+reference line, CAR+GOAL, GOAL only, and finally a completely unannotated raw bird’s-eye image (Table IV). Adding the CAR marker lifts the UGV’s success rate from 60.0% (GOAL only) to 70.0% (CAR+GOAL), the reference line further to 75.0%, and full annotation (with distance labels) reaches 77.0% joint success. Notably, GOAL-only (60.0%) underperforms the unannotated raw image (65.0%), indicating that a lone GOAL marker without the teammate’s CAR anchor misleads the UGV’s path planner. Figure 3 summarizes the three ablation axes.

G. Failure Source Analysis (RQ3)

Figure 4 shows representative episodes. We attribute each failed episode to the *first* pipeline stage that deviated from ground truth (Table V): **global localization** (the UAV’s VLM marks a wrong target region), **path planning** (an undrivable path), or **local execution** (failing to close the final meters).

Measured across the 23 jointly-failed episodes, path planning is the dominant stage (14/23 = 61%; Table V); the

TABLE I

MAIN RESULTS OF AIR-GROUND COLLABORATIVE VLN (100 EPISODES). SEE THE NOTE BELOW THE TABLE FOR COLUMN DEFINITIONS.

Method	SR _{UGV} ↑	SR _{UAV} ↑	SR _{joint} ↑	SPL ↑	NE (m) ↓	CG ↑	Time (s) ↓	VLM calls ↓	Path _{UAV} (m) ↓	Path _{UGV} (m) ↓
<i>Published single-agent VLN baselines</i>										
OpenFly [32]	—	0.0%	0.0%	0.0%	78.4±18.7	0.0%	—	34.1±7.3	56.4±16.0	—
FineCog-Nav [6]	—	13.0%	13.0%	10.0%	28.4±12.9	0.0%	177.2±35.4	58.0±12.4	19.9±10.4	—
3DG-VLN [33]	—	33.0%	33.0%	26.0%	49.6±32.1	0.0%	142.1±60.4	11.7±5.2	49.2±21.7	—
Travel UAV [34]	—	53.0%	53.0%	45.0%	30.3±24.3	0.0%	115.8±63.4	21.0±11.5	42.24±27.70	—
<i>Single-agent ablations (our components)</i>										
UGV-only VLM	12.0%	—	12.0%	11.6%	115.2±65.6	0.0%	28.2±3.0	12.1±4.0	—	49.4±4.0
UAV-only SPF [3]	—	20.0%	20.0%	18.0%	155.5±94.9	0.0%	58.5±9.6	12.9±6.5	119.1±74.1	—
UAV-only 3D-SPF	—	60.0%	60.0%	46.0%	30.0±40.9	0.0%	104.0±8.0	13.0±7.1	84.3±36.7	—
<i>Ours: Air-Ground Collaborative VLN</i>										
AGC-VLN (ours)	75.0%	50.0%	77.0%	62.0%	10.9±16.2	+27.0%	82.2±27.0	12.1±2.9	65.1±22.2	43.5±14.0

Note: ↑/↓ = higher/lower is better. **SR_{UAV}/SR_{UGV}/SR_{joint}:** UAV/UGV/joint success rate (goal within 5 m). **SPL:** success weighted by path length; **NE:** navigation error; **CG:** collaboration gain = $\text{SR}_{\text{joint}} - \min(\text{SR}_{\text{UAV}}, \text{SR}_{\text{UGV}})$; **Time:** mean arrival time over successful episodes; **VLM calls:** inference count; **Path_{UAV}/Path_{UGV}:** distance traveled. SR/SPL/CG are percentages; NE/VLM calls/path are mean±std over all episodes; Time is mean±std over successful episodes; “—” = not applicable/reported.

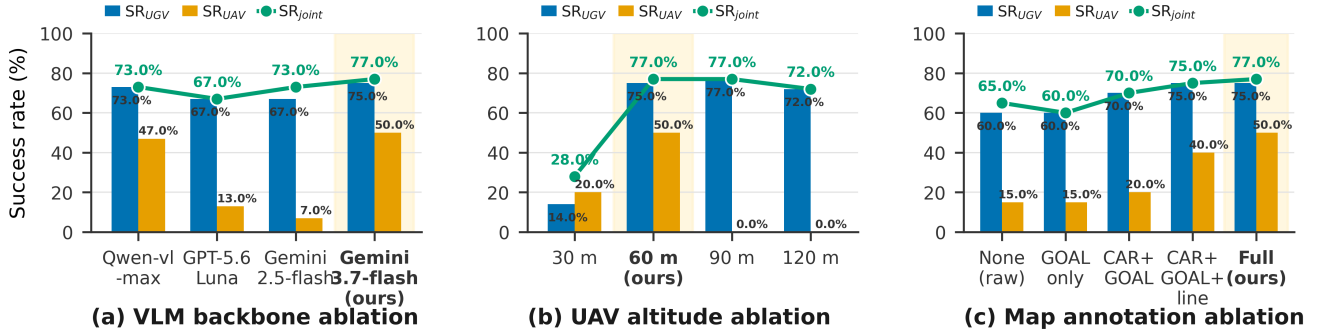


Fig. 3. Ablation across the three design axes. (a) VLM backbone (Qwen-vl-max, GPT-5.6 Luna, Gemini 2.5-flash, and Gemini 3.7-flash). (b) UAV initial altitude (30/60/90/120 m). (c) Map annotation richness (raw, goal-only, CAR+GOAL, CAR+GOAL+line, and full). Each panel reports SR_{joint} (green line) together with the per-agent success rates SR_{UGV} and SR_{UAV} (bars), with percentage value labels and the adopted configuration (“ours”) highlighted.

TABLE II
VLM BACKBONE ABLATION.

VLM	SR _{UGV} ↑	SR _{UAV} ↑	SR _{joint} ↑	CG ↑	Time (s) ↓
qwen-vl-max	73.0%	47.0%	73.0%	+26.0%	75.23±68.80
gpt-5.6-luna	67.0%	13.0%	67.0%	+54.0%	102.97±63.10
gemini-2.5-flash	67.0%	7.0%	73.0%	+66.0%	91.91±63.36
gemini-3.7-flash	75.0%	50.0%	77.0%	+27.0%	82.2±27.0

TABLE III
3D-SPF ALTITUDE ABLATION

Initial altitude	SR _{UGV} ↑	SR _{UAV} ↑	SR _{joint} ↑	CG ↑	Time (s) ↓
30 m	14.0%	20.0%	28.0%	+14.0%	141.3±29.7
60 m (Ours)	75.0%	50.0%	77.0%	+27.0%	82.2±27.0
90 m	77.0%	0.0%	77.0%	+77.0%	66.1±29.5
120 m	72.0%	0.0%	72.0%	+72.0%	70.6±35.0

UGV’s VLM emits a path that collapses onto a building, tree, or the map border, and the vehicle crashes and stalls 16–73 m from the goal. Local execution (5/23 = 22%) closes within 15 m but hits the 180 s limit. Global localization is the residual UAV-side failure (4/23 = 17%), split between unstable tracking (3/23 = 13%), whose points jump across frames, and mis-anchoring (1/23 = 4%), the reference-binding failure we had hypothesized, in which the UAV locks

onto a similar but wrong truck and flies toward it. Map rendering/projection contributed no failures and is omitted.

Per-agent, the UAV fails from global localization (76%) and the UGV from path planning (60%); the UGV rescues 27 of the UAV’s 50 failures versus 2 the other way. Cross-view target re-identification is the highest-leverage learned component [10].

TABLE IV

MAP ANNOTATION RICHNESS ABLATION (BIRD’S-EYE MAP SHARING)

Map annotation	SR _{UGV} ↑	SR _{UAV} ↑	SR _{joint} ↑	CG ↑	Time (s) ↓
None (raw)	60.0%	15.0%	65.0%	+50.0%	118.8±53.4
GOAL only	60.0%	15.0%	60.0%	+45.0%	116.0±55.4
CAR+GOAL	70.0%	20.0%	70.0%	+50.0%	106.2±51.8
CAR+GOAL+line	75.0%	40.0%	75.0%	+35.0%	95.6±50.6
Full (ours)	75.0%	50.0%	77.0%	+27.0%	82.2±27.0



Fig. 4. Representative episodes. Success: the annotated map drives a VLM road path that the UGV tracks to the goal, with the UAV and UGV both arriving. Mis-anchoring failure: the UAV’s VLM anchors onto a similar but wrong truck and flies toward the wrong object. Path-planning failure: the UGV’s VLM plans a path that collides with a tree, stalling the UGV.

V. REAL-ROBOT EXPERIMENT CASE

Beyond the simulation, we deploy the same training-free pipeline on real hardware to verify its feasibility once decoupled from the simulator’s synchronized pose stream. A UAV (downward RGB camera) and a UGV (forward camera) form the team. The UAV runs 3D-SPF, and the UGV runs road-level path planning on the shared bird’s-eye map. The frozen VLM remains *gemma-3.7-flash*, and the decision cadence matches the simulation. In the deployment the UAV pose is provided by LiDAR odometry, while the stages (annotation, projection, closed-loop execution) are identical to the simulation.

The physical deployment confirms that the training-free pipeline drives both agents to the target, as shown in Figure 5, which depicts the real-world air-ground collaborative VLN hardware and scene. The UAV runs Linux and carries a Mid360 LiDAR and a RealSense D435i, while the UGV carries a forward camera. The scene spans six time steps in both top-down and third-person views: the UAV first climbs to altitude, calls the VLM to annotate the bird’s-eye view, and sends it to the UGV, which calls the VLM to plan a road path (the green point set); the UAV then approaches the target with the proposed 3D-SPF, while the UGV automatically steers

TABLE V

FAILURE ATTRIBUTION BY PIPELINE STAGE

Stage	Share	Typical case
Global localization	17%	
mis-anchoring	4%	locks onto a wrong target
unstable tracking	13%	points jump across frames
Path planning	61%	undrivable path; vehicle crashes and stalls
Local execution	22%	final-approach dead end or timeout
Total	100%	23 jointly-failed episodes

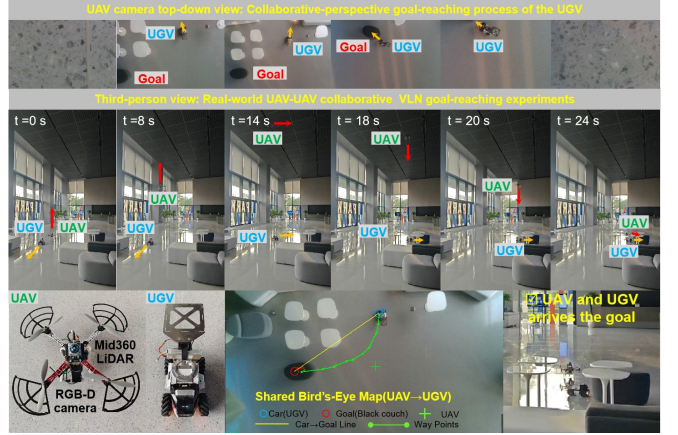


Fig. 5. Real-robot case: the same training-free pipeline (3D-SPF + shared bird’s-eye map road planning) running on a physical UAV+UGV team. Six time steps in top-down and third-person views show the quadrotor (Mid360 LiDAR + RealSense D435i) annotating the bird’s-eye map and flying 3D-SPF toward the target, while the omnidirectional robot follows its VLM-planned road path.

along the planned path and moves toward the target.

VI. LESSONS AND LIMITATIONS

A. Lessons for Air-Ground Collaboration

Lesson 1: put the interface at the global view. The UAV’s bird’s-eye view naturally carries the global relative positions of teammate and target, spatial relationships that mean the same thing in both agents’ frames, so choosing it as the communication interface (rather than text, features, or control signals) is what turns collaboration from hurting into helping relative to [11].

Lesson 2: view complementarity, not message richness. The gain comes not from transmitting richer information but from supplying the viewpoint the receiver lacks: the global spatial context invisible in the UGV’s first-person view, which lets the UGV’s VLM turn the UAV’s “can see” into the UGV’s “can drive”.

Lesson 3: determinism confines error propagation. Because projection and inverse projection contain no learned parts, a VLM mistake on one platform cannot propagate to the other, structurally the opposite of the C2 velocity coupling that amplified errors in [11].

B. Limitations

The baseline adopts three deliberate simplifications. The shared map marks the target with the episode-provided goal position, which keeps the collaboration interface deterministic, while the UAV’s own flight (3D-SPF) still inherits the VLM’s mis-anchoring risk. The 3 s VLM latency confines the agents to quasi-static scenes, leaving fast-moving targets out of reach. And CARLA-Air’s synchronized poses stand in for the GPS/SLAM a real field deployment would need. Each simplification is intentional: a minimal baseline keeps the remaining gaps measurable.

VII. CONCLUSION

We presented **AGC-VLN** (Air-Ground Collaborative VLN), the first training-free baseline for air-ground collaborative VLN, which couples 3D-SPF on the UAV with VLM road-path planning on the UGV through a **shared bird’s-eye map**. On 100 closed-loop episodes in Town10HD it achieves a clear positive collaboration gain, in contrast to trained VLA models whose coupling degraded single-agent performance. Residual failures concentrate on global localization, chiefly mis-anchoring onto a wrong truck (reference binding), which delineates a concrete agenda: robust cross-view target re-identification, learned verification, and latency-robust coordination. We release the full system as a reproducible starting point for the field.

REFERENCES

- [1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” in *CVPR*, 2018.
- [2] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. van den Hengel, “Reverie: Remote embodied visual referring expression in real indoor environments,” in *CVPR*, 2020.
- [3] C.-Y. Hu, Y.-S. Lin, Y. Lee, C.-H. Su, J.-Y. Lee, S.-R. Tsai, C.-Y. Lin, K.-W. Chen, T.-W. Ke, and Y.-L. Liu, “See, point, fly: A learning-free vlm framework for universal unmanned aerial navigation,” in *Proceedings of the 9th Conference on Robot Learning (CoRL)*. PMLR, 2025, pp. 4697–4708.
- [4] H. Ding, S. Zhang, Z. Xu, *et al.*, “Uni-lavira: Language-vision-robot actions translation for unified embodied navigation,” *arXiv preprint arXiv:2605.27582*, 2026.
- [5] Z. Xu, Y. Lu, W. Bao, *et al.*, “Fly0: Persistent metric anchoring for zero-shot aerial vision-language navigation,” *arXiv preprint arXiv:2602.15875*, 2026.
- [6] D. Shao, Z. Xu, P. Wang, L. Liu, Y. Wang, J. Shi, and J. Huo, “Finecog-nav: Integrating fine-grained cognitive modules for zero-shot multimodal uav navigation,” in *CVPR 2026 Findings*, 2026.
- [7] T. Zeng, Y. Wen, H. Chen, and H. Zhang, “Carla-air: Fly drones inside a carla world – a unified infrastructure for air-ground embodied intelligence,” *arXiv preprint arXiv:2603.28032*, 2025.
- [8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” 2017.
- [9] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” 2018.
- [10] H. Li, Y. Wang, L. Wang, *et al.*, “Airgroundbench: Probing spatial intelligence in multimodal large models under heterogeneous multi-view embodied collaboration,” *arXiv preprint arXiv:2606.28049*, 2026.
- [11] T. Zeng, Y. Wen, X. Yu, and H. Zhang, “Can aerial vla models cooperate? evaluating closed-loop air-ground coordination with carla-air,” *arXiv preprint arXiv:2605.31066*, 2026.
- [12] A.-C. Cheng, Y. Ji, Z. Yang, *et al.*, “Navila: Legged robot vision-language-action model for navigation,” in *Conference on Robot Learning (CoRL)*, 2024.
- [13] J. Chen, B. Lin, R. Xu, Z. Chai, X. Liang, and K.-Y. K. Wong, “Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation,” in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
- [14] M. Chang, T. Gervet, M. Khanna, *et al.*, “Goat: Go to any thing,” in *Robotics: Science and Systems (RSS)*, 2024.
- [15] M. Wei, C. Wan, X. Yu, *et al.*, “Streamvln: Streaming vision-and-language navigation via slowfast context modeling,” *arXiv preprint arXiv:2507.05240*, 2025.
- [16] R. Gong, Y. Guo, J. Hu, *et al.*, “Abot-nl: Toward a general visual language navigation foundation model,” *arXiv preprint arXiv:2607.10383*, 2026.
- [17] Y. Hu, A. Xi, Q. Xiao, S. Isaacson, H. X. Liu, R. Vasudevan, and M. Ghaffari, “Longnav-r1: Horizon-adaptive multi-turn rl for long-horizon vla navigation,” *arXiv preprint arXiv:2602.12351*, 2026.
- [18] B. Lin, Y. Nie, K. L. Zai, *et al.*, “Evolvenav: Empowering llm-based vision-language navigation via self-improving embodied reasoning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2026.
- [19] T. Wang, X. Li, F. Lu, T. Gong, J. Dong, W. Xue, S. Qu, C. Bai, and G. Chen, “Conavbench: Collaborative long-horizon vision-language navigation benchmark,” in *International Conference on Learning Representations (ICLR)*, 2026.
- [20] R. Wu, Y. Zhang, J. Chen, *et al.*, “Aeroduo: Aerial duo for uav-based vision and language navigation,” in *ACM Multimedia (MM)*, 2025.
- [21] Z. Liu, M. He, S. Yu, *et al.*, “Omnivln: Omnidirectional 3d perception and token-efficient llm reasoning for visual-language navigation across air and ground platforms,” *arXiv preprint arXiv:2603.17351*, 2026.
- [22] S. Zeng, D. Qi, X. Chang, *et al.*, “Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation,” in *International Conference on Learning Representations (ICLR)*, 2026.
- [23] J. Lee, T. Miyayoshi, S. Kurita, K. Sakamoto, D. Azuma, Y. Matsuo, and N. Inoue, “Citynav: A large-scale dataset for real-world aerial navigation,” in *International Conference on Computer Vision (ICCV)*, 2025.
- [24] S. Li, X. Dong, X. Ma, J. Chen, H. Zhao, and Y. Zhou, “Closervln: Closed-loop self-verified retrieval-augmented reasoning for aerial vision-language navigation,” *arXiv preprint arXiv:2606.28397*, 2026.
- [25] X. Zhu, Q. Meng, L. Yu, W. Zhang, Z. Ma, H. Zhou, and Y. Tian, “Fsdvln: Fast-slow dual-system modeling for aerial long-horizon vision-language navigation,” *arXiv preprint arXiv:2607.08359*, 2026.
- [26] H. Zhang, M. Liu, Q. Xiang, K. Wang, Y. Wang, and L. Nie, “Spatialuav: Benchmarking spatial intelligence for low-altitude uav perception, collaboration, and motion,” *arXiv preprint arXiv:2606.27876*, 2026.
- [27] M. S. Mondal, L. Russo, J. D. Humann, J. M. Dotterweich, and P. Bhounsule, “Towards reliable aerial ground vehicle collaboration: An integrated planning and autonomy framework for field deployment,” *arXiv preprint arXiv:2607.07350*, 2026.
- [28] H. Chen, J. Zheng, S. Yang, *et al.*, “Vision-and-language navigation for uavs: Progress, challenges, and a research roadmap,” *arXiv preprint arXiv:2604.13654*, 2026.
- [29] Y. Zhang, X. Zhang, G. Sun, *et al.*, “Geo²: Geometry-guided cross-view geo-localization and image synthesis,” in *CVPR*, 2026.
- [30] R. Wang, W. Yuan, W. Yuan, *et al.*, “Gengeo: Robust cross-view geo-localization via foundation model and dynamic feature aggregation,” *Remote Sensing*, vol. 18, no. 8, p. 1116, 2026.
- [31] G. Gemini Team, “Gemini: A family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [32] Y. Gao, C. Li, Z. You, *et al.*, “Openfly: A comprehensive platform for aerial vision-language navigation,” in *International Conference on Learning Representations (ICLR)*, 2026.
- [33] F. Xue, E. Yu, Y. Shen, Z. Hu, H. Wang, Y. Yang, X. Wang, and J. Sun, “See-and-reach: Precise vision-language navigation for uavs within the field of view,” *arXiv preprint arXiv:2606.20045*, 2026.
- [34] X. Wang, D. Yang, Z. Wang, *et al.*, “Towards realistic uav vision-language navigation: Platform, benchmark, and methodology,” in *International Conference on Learning Representations (ICLR)*, 2025.