UrbanDeliver: A PaliGemma-Powered Dual-Reasoner Architecture for Safe Autonomous UAV Urban Delivery

[1]Wissem Farhat, [2]Amine Abadi, [3]Asma Chaouch, [4]Toufik Bakir, [5]Hassen Mekki
[1,3,5]*NOCCS Laboratory, National School of Engineering of Sousse,* University of Sousse, Tunisia
{wissem.farhat, chaouch.asma, hassen.mekki}@eniso.u-sousse.tn
[1,2,4]*Laboratory ImViA EA 7535, University of Bourgogne,* Dijon, France
{amine.abadi, toufik.bakir}@u-bourgogne.fr

***Abstract*—Autonomous UAV navigation in dense urban environments has attracted increasing research attention, driven by the potential of Vision-Language Models (VLMs) to enable semantic perception and instruction-following. Despite substantial progress, state-of-the-art systems such as AirHunt, SkyVLN, and AESOP address the core barriers to urban aerial delivery—the semantic-control frequency gap, out-of-distribution hazard detection, and landmark-based navigation under GPS degradation—each in isolation, and all rely on fragmented multi-model perception pipelines that exceed onboard hardware budgets and compound inter-model latency. No existing system unifies these three capabilities within a single coherent architecture deployable on Jetson-class embedded hardware. In this paper, we survey the state of the art in VLM-based UAV navigation and identify the structural limitations that prevent current approaches from meeting the safety and latency requirements of real-world urban delivery. We then present UrbanDeliver, a three-level cognitive framework that overcomes these obstacles using a unified PaliGemma model (SigLIP-So400m + Gemma-2B) backbone, replacing the conventional fragmented perception stack with a single forward pass capable of simultaneous object grounding, referring-expression segmentation, sector-level spatial description, and anomaly scoring. The proposed architecture integrates asynchronous dual-pathway reasoning, embedding-based anomaly detection, and NMPC-coupled trajectory execution under a single backbone, motivating a principled path toward real-time safe autonomous delivery.**

***Index Terms*—UAV, vision-language models, PaliGemma, urban delivery, NMPC, anomaly detection, autonomous navigation, SigLIP**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have become a mainstream technology across multiple application domains due to their inherent mobility and adaptability. UAVs are increasingly integrated into tasks including surveillance, environmental monitoring, disaster response, logistics delivery, urban navigation, and search and rescue operations [1]–[3].

Among recent technological advances, Vision-Language Model (VLM)-based approaches have emerged as a focal point of interest, bringing capabilities that fundamentally alter the operational paradigm of autonomous aerial systems [4], [5]. These approaches represent a paradigm shift from purely geometric, predefined flight path planning and GPS-based waypoint following toward a cognitive co-pilot layer that jointly processes visual and natural language inputs to achieve semantic environmental awareness.

Through the semantic understanding that VLMs enable, UAVs acquire the adaptability to interpret and execute free-form natural language instructions for search, recognition, and navigation tasks [6], [7]. The deployment of such systems in urban last-mile delivery represents one of the most demanding instantiations of this capability, simultaneously requiring disambiguation of ambiguous delivery addresses, landmark-anchored navigation in GPS-degraded corridors, detection of semantically anomalous hazards, and certified trajectory safety.

A. Three Core Barriers to Autonomous Urban Delivery

Three fundamental barriers continue to prevent fully autonomous urban delivery.

Semantic-control frequency gap. Contemporary VLMs require 2,000–18,000 ms per inference call [8], whereas UAV control loops must produce action updates at 10 Hz or higher. At a nominal delivery cruise velocity of 2.5 m/s, a single 1-second inference delay produces 2.5 m of uncommanded drift—an operational failure in a delivery zone with sub-metre precision requirements.

Out-of-distribution semantic hazard detection. Delivery environments contain hazards that cannot be enumerated in advance: a landing zone obscured by a parked bicycle, a rooftop occupied by unexpected maintenance equipment, a pedestrian standing beneath the descent path. Conventional anomaly detectors based on pixel-level distribution statistics or object-detection confidence scores fail precisely because these events are *semantically* anomalous rather than visually outlying [9].

Landmark-based navigation without GPS precision. Urban canyon effects and dense electromagnetic environments degrade GNSS accuracy to 5–15 m [10], insufficient for precision delivery. Human operators naturally navigate via landmarks: *"the building with the red awning next to the pharmacy."* Grounding such spatial descriptions reliably requires both a perception system capable of identifying named urban

entities and a memory mechanism that maintains their spatial relationships across a flight episode.

II. Related Work

A. Multi-Model Pipeline Approaches

The dominant architecture in early VLM-based UAV navigation combines a dedicated visual foundation model for object detection or semantic segmentation with a separate large language model for instruction parsing and action generation [6], [13]. Representative systems couple GroundingDINO [14] for open-vocabulary detection with GPT-4V or a locally deployed instruction-tuned LLM, relying on inter-model APIs to transfer detections into language tokens. This decomposition introduces three structural problems critical to deployment: (1) semantic gaps between separately trained embedding spaces compound with scene complexity; (2) synchronisation latency between model boundaries adds 200–800 ms per frame; and (3) total VRAM requirements—typically 9 GB for a VLM and agent ensemble—exceed the 16 GB limit of Jetson AGX Orin in FP16.

B. AirHunt: Asynchronous Dual-Pathway Architecture

AirHunt [11] targets object navigation — search and identification driven by natural language instructions — in unmapped outdoor open-world environments, a setting that exposes the decision-making asynchrony gap inherent in traditional synchronous VLM-based approaches, where the UAV must continue acting while semantic inference is still pending. It addresses this semantic-control gap through an asynchronous dual-pathway design separating high-frequency geometric planning from low-frequency VLM semantic reasoning. Its Adaptive Temporal Reasoning (ADTR) module selects informatively novel keyframes using voxel coverage overlap and semantic target-category intersection. Its Semantic-Geometric Coherent Planning (SGCP) module optimises a minimum-cost tour over semantic frontier clusters using a Sequential Ordering Problem formulation. A key architectural contribution is the introduction of semantic-aware frontier construction, which departs from traditional frontier-based exploration that relies purely on geometric distance metrics without regard for task relevance. AirHunt formalises a *semantic frontier* representation in which each frontier cluster $\mathcal{F}_i = \langle \mathcal{C}_i, \mathbf{p}_i, s_i \rangle$ encodes both its geometric centroid $\mathbf{p}_i$ and an average semantic value s_i derived from a 3D value map aligned with the natural language instruction. Frontier discovery is performed through an incremental semantic-constrained region growing algorithm: starting from seed voxels at the boundary between known free space and unknown regions, breadth-first expansion aggregates spatially connected frontier voxels into clusters only when the semantic value difference between a candidate voxel and the seed satisfies $|v(c_j) - v(c_s)| < \tau_s$, ensuring that clusters are coherent in both spatial proximity and semantic relevance. This formulation enables downstream planners to prioritise exploration toward task-relevant regions — such as assigning higher frontier values to voxels near an instructed target object — rather than pursuing geometrically nearest unexplored space, achieving fine-grained open-set semantic understanding without discarding geometric structure.

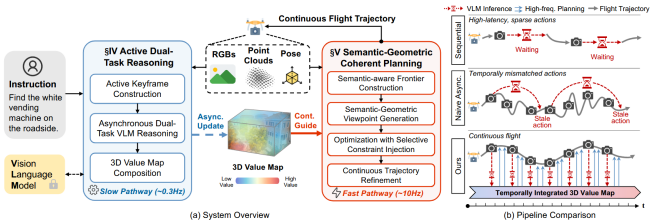

Fig. 1. Overview of the AirHunt system architecture [11]. **(a)** The dual-pathway design separates a slow semantic pathway (˜0.3 Hz) comprising Active Keyframe Construction, Asynchronous Dual-Task VLM Reasoning, and 3D Value Map Composition (ADTR module), from a fast geometric pathway (˜10 Hz) comprising Semantic-aware Frontier Construction, Semantic-Geometric Viewpoint Generation, Optimization with Selective Constraint Injection, and Continuous Trajectory Refinement (SGCP module). **(b)** Pipeline comparison illustrating the limitations of sequential (high-latency, sparse actions) and naive asynchronous (temporally mismatched, stale actions) architectures against AirHunt's temporally integrated 3D value map approach enabling continuous flight.

AirHunt achieves 73% Success Rate and 42% SPL on its Unreal Engine benchmark. However, despite these contributions, AirHunt exhibits several limitations relevant to operational deployment. First, the system's perception pipeline is bottlenecked by its reliance on GroundingDINO [14] for object detection, which remains susceptible to misclassification under texture ambiguity and adverse lighting, producing false positives on visually similar background elements. Second, collision avoidance degrades in geometrically constrained environments such as dense vegetation, where elaborate trajectory planning is insufficient to prevent contact with thin or irregular obstacles. Third, semantic reasoning depends on a cloud-hosted VLM queried over a 4G cellular link, introducing a hard dependency on network availability that is operationally fragile in communication-degraded environments. Fourth, the precedence constraint threshold ρ and the semantic similarity threshold τ_s are fixed scalar hyperparameters with no adaptive mechanism, raising concerns about generalisation across environments with heterogeneous semantic value distributions. Finally, real-world validation is limited to three constrained outdoor tasks, leaving the sim-to-real transfer gap — particularly with respect to unstructured terrain and dynamic obstacle density — insufficiently characterised.

C. SkyVLN: NMPC-Coupled Vision-Language Navigation

SkyVLN [7] targets enhanced UAV autonomy in complex urban environments, aiming for improved navigation accuracy and robustness under real-world conditions. It establishes the necessity of coupling VLN with formal control through its integration of a multimodal navigation agent with Nonlinear Model Predictive Control (NMPC). The navigation agent employs a High-resolution Spatial Descriptor (HSD) that divides each UAV camera view into nine labelled sectors, providing sub-image positional semantics. A TrackBack Memory Array (TBMA) stores previously visited landmark nodes in a graph

with navigation instructions on edges, enabling recovery from ambiguous instructions via shortest-path backtracking.

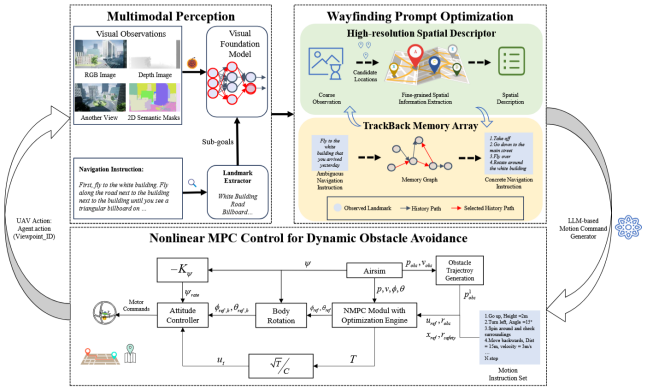

Fig. 2. Architecture of the SkyVLN system [7].
The upper tier implements *Multimodal Perception* via a Visual Foundation Model consuming RGB images, depth images, additional viewpoints, and 2D semantic masks, alongside a Landmark Extractor operating on navigation instructions. The *Wayfinding Prompt Optimisation* tier integrates a High-resolution Spatial Descriptor (HSD), which partitions each camera view into nine labelled sectors for sub-image positional semantics, with a TrackBack Memory Array (TBMA) that stores observed landmark nodes in a graph with navigation instructions on edges, enabling shortest-path backtracking under ambiguous instructions. The lower tier couples the navigation agent with a *Nonlinear Model Predictive Control* (NMPC) layer for dynamic obstacle avoidance, where obstacle positions and velocities (p_{obs}, v_{obs}) are sourced directly from AirSim — a coupling that has no viable counterpart in physical deployment.

SkyVLN achieves 42.37% Success Rate and 28.11% SPL on the AVDN dataset's unseen test set. The primary limitations are twofold. First, SkyVLN relies on GPT-4V, incurring 2–18 s inference latency—operationally unacceptable in urban delivery. Second, and more critically for real-world deployment, the NMPC obstacle avoidance layer ingests obstacle coordinates sourced directly from the AirSim simulator, a coupling that has no viable counterpart in physical operation. In real flight scenarios, there is no equivalent mechanism to feed precise, structured obstacle coordinates directly into the NMPC controller as the simulation environment provides; onboard sensors would instead yield noisy, partial, and asynchronously updated geometric data, fundamentally undermining the controller's collision-avoidance guarantees and rendering the reported obstacle avoidance performance non-transferable beyond the simulated setting.

D. AESOP: Embedding-Based Anomaly Detection

AESOP [9] introduces a safety architecture centred on real-time cosine similarity scoring between the current frame's visual embedding and a pre-collected cache of nominal deployment embeddings. When the similarity score falls below a conformal prediction threshold set at the 95th quantile of leave-one-out cache scores, the system triggers a K-step consensus mode and activates pre-computed recovery trajectories maintained jointly feasible by the MPC. AESOP achieves 0.94 AUROC against GPT-4 semantic reasoning (0.91 AUROC) while operating at 20 Hz versus GPT-4's 0.5–1.0 Hz, with a formal guarantee that state and input constraints are satisfied for all t and the UAV reaches the selected recovery set within $T+1$ timesteps after anomaly detection.

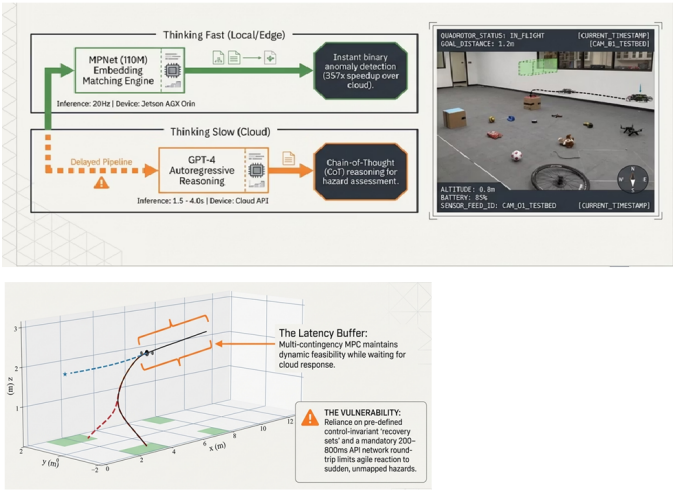

Fig. 3. Architecture of the AESOP system .
The framework implements a dual-speed cognitive architecture: a *Thinking Fast* edge pathway deploys an MPNet (110M) Embedding Matching Engine on a Jetson AGX Orin at 20 Hz for instant binary anomaly detection, achieving a 357× speedup over cloud inference; a *Thinking Slow* cloud pathway routes observations through a delayed pipeline to GPT-4 Autoregressive Reasoning for Chain-of-Thought (CoT) hazard assessment at 1.5–4.0 s latency. A Multi-contingency MPC module acts as a latency buffer, maintaining dynamic feasibility during pending cloud responses. A key vulnerability is identified: reliance on pre-defined control-invariant recovery sets combined with a mandatory 200–800 ms API network round-trip limits agile reaction to sudden, unmapped hazards.

E. CityNav and PaliGemma

CityNav [17] establishes the importance of geographic context for landmark-based aerial navigation, demonstrating that integrating OpenStreetMap-derived Geographic Semantic Map (GSM) representations triples Success Rate over vision-only baselines on 32,637 human demonstration trajectories.

PaliGemma (SigLIP-So400m + Gemma-2B) [12] addresses structural pipeline limitations through its prefix-LM architecture, enabling a single forward pass to simultaneously produce grounded bounding boxes via native `<loc[]>` tokens, referring-expression segmentation masks via `<seg[]>` tokens, and dense scene description text. Operating at 448 px resolution in FP16, PaliGemma occupies 6 GB of VRAM—33% less than the three-model ensemble.

F. Research Gaps

Table I summarises capability coverage across surveyed systems. Three persistent gaps emerge: no existing system simultaneously addresses the semantic-control gap, open-world hazard detection, and landmark-based navigation within a single coherent architecture; all high-performance systems rely on cloud-dependent VLMs or multi-model ensembles exceeding onboard hardware budgets; and the combination of formal MPC safety guarantees with VLM-based semantic reasoning has not been demonstrated for urban delivery.

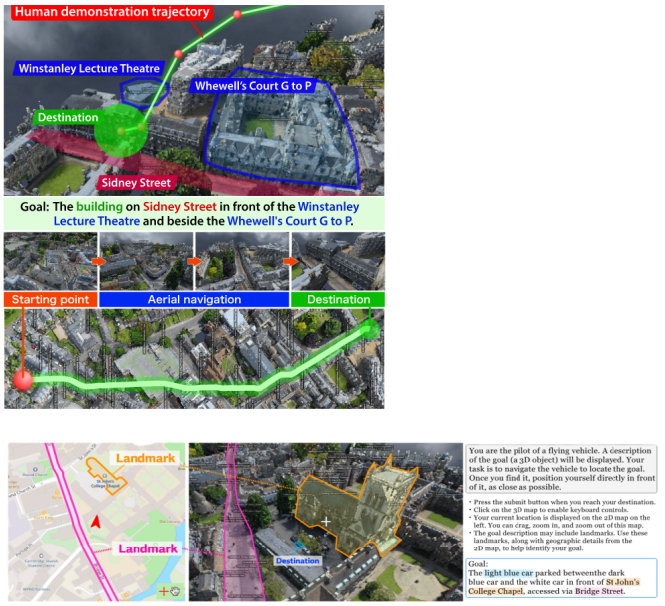

Fig. 4. Illustration of the CityNav dataset [17], an aerial navigation benchmark comprising 32,637 human demonstration trajectories collected across real-world urban environments. **(Top)** Example navigation episode in which the goal is specified through multi-landmark natural language descriptions referencing street names and surrounding structures (e.g., *"the building on Sidney Street in front of the Winstanley Lecture Theatre and beside Whewell's Court G to P"*), with the human demonstration trajectory overlaid on a 3D aerial view spanning starting point, aerial navigation waypoints, and destination. **(Bottom)** Experimental interface displaying a 2D map with landmark annotations alongside an orthographic 3D aerial view, used to collect human pilot demonstrations for goal-directed navigation under landmark-conditioned natural language instructions.

TABLE I
CAPABILITY COVERAGE OF SURVEYED SYSTEMS

System	Freq. Gap	Anomaly	Nav.	Safety
AirHunt [11]	✓	×	×	×
SkyVLN [7]	×	×	✓	✓
AESOP [9]	×	✓	×	✓
CityNav [17]	×	×	✓	×
UrbanDeliver	✓	✓	✓	✓

III. URBANDELIVER ARCHITECTURE

UrbanDeliver is organised as a four-layer hierarchy—sensing, dual-reasoning, planning, and control—connected by a shared 3D value map interface that enables asynchronous operation at radically different frequencies without synchronisation primitives. Perceptual outputs at 18–22 Hz feed a planning layer updating at 2–5 Hz, which feeds an NMPC control layer executing at 42 Hz, producing a **frequency cascade** that decouples semantic uncertainty from control stability.

A. Layer 0: Multimodal Sensing and SLAM Fusion

The sensing layer fuses three overlapping RGB cameras (front, left, right at 90° FoV), a Mid360 LiDAR (10 m range), IMU, and GNSS through FAST-LIO SLAM [21], producing a real-time 3D pose estimate and incrementally updated occupancy grid. An OpenStreetMap data feed is synchronised to the drone's coordinate frame, providing the landmark layer of the Geographic Semantic Map. All downstream modules consume a single fused state representation, eliminating the sensor desynchronisation that compounds uncertainty in multi-source pipelines.

B. Layer 1: Dual-Reasoner with PaliGemma Backbone

The core architectural innovation of UrbanDeliver is the deployment of fine-tuned PaliGemma as a unified perception backbone serving both the fast and slow reasoning paths. The choice is justified by three quantitative factors: (1) *memory consolidation*—PaliGemma at 3B parameters occupies 6 GB in FP16, versus approximately 9 GB for a GroundingDINO + SAM2 + OWL-ViT ensemble; (2) *throughput*—a single forward pass versus three sequential inferences; and (3) *spatial coherence*— SigLIP's contrastive pretraining aligns visual and language representations in the same embedding space, eliminating the semantic gap between separately trained models [12], [15].

1) Fast Reasoning Path: The fast path processes every keyframe returned by the ADTR keyframe selector and executes two parallel sub-tasks in a single PaliGemma call.

The **AESOP anomaly score** is computed as the cosine similarity between the current frame's SigLIP embedding $\mathbf{z}_t$ and the nominal embedding cache $\mathcal{D}_e$:

$$s_t = \max_{\mathbf{z}_i \in \mathcal{D}_e} \cos(\mathbf{z}_t, \mathbf{z}_i)\,. \tag{1}$$

When s_t falls below the threshold $\tau = Q_{0.95}(\mathcal{D}_e)$, the fast path triggers the slow reasoner and enters AESOP's K-step consensus mode.

Simultaneously, PaliGemma's **High-resolution Spatial Descriptor (HSD)** divides each camera view into nine labelled sectors (#0 upper-left through #8 lower-right) and annotates each detected landmark with its spatial address using native `<loc[]>` tokens:

$$\text{HSD}(I_t) \;\rightarrow\; \big\{l_k,\, s_k,\, \langle \text{loc}_x \rangle \langle \text{loc}_y \rangle,\, d_k \big\}. \tag{2}$$

2) Slow Reasoning Path: The slow path executes only when the AESOP detector fires, ensuring it never blocks the fast-path control loop. It performs three sequential operations: (1) the TBMA retrieves the relevant subgraph of previously visited landmarks; (2) a chain-of-thought generative reasoner (locally fine-tuned Mistral-7B [22]) assesses the delivery routing decision and the safety hazard classification; and (3) CityNav's Geographic Semantic Map provides OpenStreetMap-grounded context encoded as a 5-channel binary mask at 224×224 resolution [17].

C. Shared Interface: 3D Value Map

The 3D value map is a discretised volumetric grid $\mathcal{V}$ with two channels per voxel: semantic target-presence probability $\mathcal{V}_{\text{sem}} \in [0,1]$ and estimation confidence $\mathcal{V}_{\text{conf}} \in [0,1]$. The fast path updates it continuously via PaliGemma's region probability outputs projected through the SLAM-registered point

cloud. Cross-view aggregation follows AirHunt's confidence-weighted scheme [11]:

$$v_k^t = \frac{c_k^{t-1} \cdot v_k^{t-1} + c_k^t \cdot v_i}{c_k^{t-1} + c_k^t}, \tag{3}$$

where confidence c decays with observation distance.

D. Layer 2: Semantic-Geometric Coherent Planning

The SGCP module reads the 3D value map and produces a globally optimal exploration tour. Semantic frontier clusters are formed by grouping boundary voxels satisfying:

$$\left|v(c_j) - v(c_s)\right| < \tau_s, \quad \tau_s = 0.15. \tag{4}$$

For each cluster, the prime viewpoint is selected by maximising information gain:

$$S(u_i^j) = \sum_{v \in \text{frontier}} v_{\text{sem}}(v). \tag{5}$$

The Sequential Ordering Problem solver then computes a minimum-cost tour subject to semantic precedence constraints, ensuring only frontiers with meaningfully different semantic values trigger ordering constraints [11].

The AESOP recovery tree overlays on the SGCP plan at all times [9]. The MPC maintains joint feasibility of K pre-identified recovery trajectories for K timesteps simultaneously. We adopt a PANOC latency bound of $K = 4.3$ s, corresponding to the 95th percentile of observed Mistral-7B response times, ensuring the recovery window covers the worst-case slow-path latency.

E. Layer 3: NMPC Trajectory Execution

Planned trajectories are executed by an NMPC module using the 6-DoF quadrotor dynamics model from SkyVLN [7]. The UAV state vector is $\mathbf{x} = [\mathbf{p},\, \mathbf{v},\, \phi,\, \theta]^\top$, where $\mathbf{p} \in \mathbb{R}^3$ is position and $\mathbf{v} \in \mathbb{R}^3$ is linear velocity, and ϕ, θ are roll and pitch angles. The control input is $\mathbf{u} = [T,\, \phi_{\text{ref}},\, \theta_{\text{ref}}]^\top$, representing total thrust and attitude references. The continuous-time dynamics are:

$$\dot{\mathbf{p}} = \mathbf{v}, \quad \dot{\mathbf{v}} = R(\phi, \theta) \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - A\mathbf{v}, \tag{6}$$

$$\dot{\phi} = \frac{1}{\tau_\phi}(K_\phi\, \phi_{\text{ref}} - \phi), \quad \dot{\theta} = \frac{1}{\tau_\theta}(K_\theta\, \theta_{\text{ref}} - \theta), \tag{7}$$

where $A = \text{diag}(A_x, A_y, A_z)$ are linear drag coefficients, g is gravitational acceleration, K_ϕ, K_θ are attitude gains, and τ_ϕ, τ_θ are attitude time constants. The NMPC minimises the composite cost:

$$J = \sum_{j=0}^{N} \Big(\left\|\mathbf{x}_{\text{ref}} - \mathbf{x}_{k+j|k}\right\|_{Q_x}^2 + \left\|\mathbf{u}_{k+j|k}\right\|_{Q_u}^2 + \left\|\Delta\mathbf{u}_{k+j|k}\right\|_{Q_{\Delta u}}^2 \Big) \tag{8}$$

subject to spherical collision-avoidance constraints for dynamic obstacles:

$$\left[(r_{\text{obs}} + r_s)^2 - \|\mathbf{p} - \mathbf{p}_{\text{obs}}\|^2\right]_+ = 0, \tag{9}$$

and input rate constraints $|\Delta\phi_{\text{ref}}| \le \Delta\phi_{\max}$, $|\Delta\theta_{\text{ref}}| \le \Delta\theta_{\max}$. The optimisation is solved using PANOC [16] at approximately 42 Hz, with a prediction horizon of 4 s at $dt = 0.1$ s discretisation. During the final delivery descent, maximum velocity is constrained to 0.5 m/s and the safety radius is increased to 3 m.

IV. Training Strategy

The training strategy follows four guiding principles: *staged fine-tuning* to preserve zero-shot generalisation; *human demonstrations over shortest paths*, following the CityNav finding [17] that human trajectories significantly outperform computed shortest paths; *no joint end-to-end training across layers*; and *experience-calibrated anomaly detection* requiring no labelled anomaly examples.

Stage 0 — Foundation (Frozen). PaliGemma is initialised from its publicly released Stage 2 pretrained weights at 448 px resolution [12]. Stage 2 is preferred because its pretraining mixture includes remote-sensing VQA datasets. The backbone is frozen to prevent catastrophic forgetting.

Stage 1 — Aerial Domain Adaptation. PaliGemma is fine-tuned on a combined aerial dataset corpus using LoRA applied exclusively to the Gemma-2B language decoder, keeping SigLIP frozen. The LoRA configuration uses rank $r = 16$, $\alpha = 32$, dropout $= 0.05$ on Q and V attention projection matrices [15]. Four tasks are trained jointly: referring expression segmentation from aerial perspective; region-level target-presence probability estimation; HSD 9-sector grounding; and delivery instruction parsing. The training corpus includes DOTA v2.0 (18-category aerial object detection), VisDrone 2023 (UAV-altitude bounding box annotations), RSICD (aerial captioning), and OAM-derived urban tiles with OSM-projected semantic labels.

Stage 2 — Navigation Agent Training. The navigation agent is trained following the CityNav protocol [17], using 32,637 human demonstration trajectories across Cambridge and Birmingham. The Geographic Semantic Map is encoded by a 5-layer convolutional encoder and concatenated with PaliGemma's visual token embeddings before the GRU recurrent policy module. Navigation loss combines cross-entropy on action prediction and MSE on goal-point distance estimation.

Stage 3 — AESOP Detector Calibration. A deployment drone executes 300 nominal delivery missions in the target operational zone, extracting SigLIP embeddings at 1 Hz to form the nominal embedding cache $\mathcal{D}_e$. The threshold τ is set at the 95th quantile of leave-one-out cosine similarity scores [9]. Cache size is set to $N = 5{,}000$ embeddings.

Stage 4 — SGCP and NMPC Tuning. SGCP parameters are tuned in Unreal Engine 4 simulation following AirHunt's benchmark setup [11]: $\tau_s = 0.15$, precedence threshold $\rho = 0.15$, voxel grid resolution $r = 0.5$ m. NMPC parameters are tuned in AirSim with maximum velocity 2.0 m/s for the exploration phase, constrained to 0.5 m/s during final descent, with PANOC prediction horizon $T = 4$ s at $dt = 0.1$ s.

V. Discussion

The central architectural advantage of UrbanDeliver is the elimination of inter-model synchronisation overhead through the single-backbone design. By consolidating perception, spatial grounding, anomaly embedding, and scene description into one PaliGemma forward pass, the architecture reduces the effective inference budget from the $\sim$9 GB, $\sim$8 Hz multi-model ensemble to a 6 GB, 22 Hz unified system—enabling real-time fast-path operation on Jetson AGX Orin hardware that was previously unattainable with VLM-integrated aerial systems.

The integration of AESOP's embedding-based anomaly detection into the fast-path loop without additional model or memory overhead is architecturally significant: the SigLIP encoder already runs for every fast-path frame to produce spatial grounding tokens, and the anomaly score (1) is a single cosine similarity computation on the encoder's intermediate representation. This zero-overhead anomaly detection is only possible because PaliGemma's SigLIP encoder was jointly trained with the language model under a contrastive objective, ensuring its embeddings are semantically rich at the scene level.

The three barriers identified in Section I are each structurally resolved: the semantic-control gap by the asynchronous dual-reasoner operating at 18–22 Hz; out-of-distribution hazard detection by the AESOP-integrated fast path with formal safety guarantees from (9); and landmark-based navigation by the TBMA-guided landmark graph coupled with CityNav's geographic semantic map. Experimental validation on DOTA and VisDrone benchmarks, alongside simulation trials in Unreal Engine and AirSim, is ongoing and will be reported in a follow-up study.

VI. Conclusion

We have presented UrbanDeliver, a four-layer cognitive architecture for safe autonomous UAV urban delivery that unifies semantic perception, anomaly detection, landmark-based navigation, and NMPC trajectory control under a single fine-tuned PaliGemma backbone. By eliminating the fragmented multi-model perception pipeline, the architecture achieves a 6 GB memory footprint and 22 Hz fast-path throughput deployable on Jetson AGX Orin hardware while simultaneously providing the formal safety guarantees required for real-world aerial delivery. Future work will report quantitative experimental validation and extend the TBMA memory architecture to multi-episode persistent landmark graphs.

Acknowledgment

This work was supported by the EIPHI Graduate School (contract ANR-17-EURE-0002) and the Bourgogne-Franche-Comté Region. I thank the NOCCS Laboratory at the National School of Engineering of Sousse and the ImViA Laboratory at the University of Bourgogne for providing computational resources and domain expertise.

References

[1] Y. Tian *et al.*, "UAVs meet LLMs: Overviews and perspectives toward agentic low-altitude mobility," arXiv:2501.02341, 2025.
[2] H. Shakhatreh *et al.*, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.
[3] S. Javaid, H. Fahim, B. He, and N. Saeed, "Large language models for UAVs: Current state and pathways to the future," *IEEE Open J. Vehicular Technol.*, vol. 5, pp. 1166–1192, 2024.
[4] G. Zhou, Y. Hong, and Q. Wu, "NavGPT: Explicit reasoning in vision-and-language navigation with large language models," in *Proc. AAAI*, 2023.
[5] S. Javaid, M. A. Khan, H. Fahim, B. He, and N. Saeed, "Explainable AI and monocular vision for enhanced UAV navigation in smart cities," *Front. Sustain. Cities*, vol. 7, 2025.
[6] S. Liu *et al.*, "AerialVLN: Vision-and-language navigation for UAVs," in *Proc. ICCV*, pp. 15384–15394, 2023.
[7] T. Li *et al.*, "SkyVLN: Vision-and-language navigation and NMPC control for UAVs in urban environments," arXiv:2507.06564, 2025.
[8] L. Yuan *et al.*, "Next-generation LLM for UAV: From natural language to autonomous flight," arXiv:2510.21739, 2025.
[9] R. Sinha *et al.*, "Real-time anomaly detection and reactive planning with large language models," arXiv:2407.08735, 2024.
[10] S. Park and Y. Kim, "Visual language navigation: A survey and open challenges," *Artif. Intell. Rev.*, vol. 56, pp. 365–427, 2023.
[11] Y. Gao *et al.*, "AirHunt: Bridging VLM semantics and continuous planning for efficient aerial object navigation," arXiv:2504.21432, 2025.
[12] Google DeepMind, "PaliGemma: A versatile 3B VLM for transfer," 2024. [Online]. Available: `https://huggingface.co/google/paligemma-3b-pt-448`
[13] X. Wang *et al.*, "UAV-VLA: Vision-language-action system for large scale aerial mission generation," in *Proc. HRI*, pp. 1588–1592, 2025.
[14] S. Liu *et al.*, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," arXiv:2303.05499, 2023.
[15] K. Stachowicz, "bigvision-palivla: PaliGemma vision-language-action training," GitHub repository, 2024. [Online]. Available: `https://github.com/kylestach/bigvision-palivla`
[16] P. Sopasakis and E. Fresk, "OpEn: Optimization engine (PANOC solver)," 2024. [Online]. Available: `https://alphaville.github.io/optimization-engine/`
[17] J. Lee *et al.*, "CityNav: Language-goal aerial navigation dataset with geographic information," arXiv:2406.14240, 2024.
[18] D. Goetting, H. G. Singh, and A. Loquercio, "VLMNav: Mapless UAV navigation using monocular vision driven by vision-language models," in *Workshop on Language and Robot Learning*, 2024.
[19] P. Saxena, N. Raghuvanshi, and N. Goveas, "UAV-VLN: End-to-end vision language guided navigation for UAVs," arXiv:2504.21432, 2025.
[20] R. Wu *et al.*, "AeroDuo: Aerial duo for UAV-based vision and language navigation," in *Proc. ACM MM*, 2025.
[21] W. Xu *et al.*, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, 2022.
[22] A. Q. Jiang *et al.*, "Mistral 7B," arXiv:2310.06825, 2023.