Scilight

Intelligence & Control
https://www.sciltp.com/journals/ic

Article

Check for updates

Large Language Model-Driven Autonomous UAV Systems: Technical Evolution, Core Architectures, and Critical Challenges

Hao Wang and Meijie Zhang *

College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China
* Correspondence: meijiezhang@stu.scu.edu.cn

How to Cite: Wang, H.; Zhang, M. Large Language Model-Driven Autonomous UAV Systems: Technical Evolution, Core Architectures, and Critical Challenges. *Intelligence & Control* **2026**, *2*(3), 4. https://doi.org/10.53941/ic.2026.100011

Received: 11 May 2026
Revised: 26 July 2026
Accepted: 23 August 2026
Published: 20 September 2026

Abstract: The rapid development of large language models (LLMs) has expanded the capabilities of autonomous unmanned aerial vehicle (UAV) systems in natural-language instruction understanding, multimodal perception, and decision-making. This survey reviews the technical evolution, system architectures, and deployment challenges of LLM-driven UAVs across perception, planning, control, multi-agent coordination, and edge–cloud computing. Beyond cataloguing representative systems, we separate semantic-reasoning latency, control timing, power, task outcomes, hardware, and validation settings to avoid misleading cross-platform comparisons. We further analyze Sim2Real gaps, intermittent connectivity, hallucination-induced action risk, cyberattacks, and privacy leakage. In this survey, a semantically adaptive safety certificate denotes a runtime-verifiable CBF/MPC/STL constraint whose safe set or margin is parameterized by grounded task and scene semantics but enforced by a deterministic safety layer outside the generative model. Future directions emphasize hierarchical lightweight reasoning, communication-aware autonomy, formally bounded semantic adaptation, and airworthiness-oriented assurance.

Keywords: large language model; autonomous UAV; multimodal perception; trajectory planning; runtime assurance; edge–cloud computing

1. Introduction

Unmanned aerial vehicles (UAVs) support inspection, rescue, agriculture, delivery, and mapping, but conventional perception–mapping–planning–control pipelines remain difficult to configure through natural language and brittle when semantic targets, sensing conditions, or dynamics differ from design assumptions [1]. Geometric planners provide interpretable motion generation [2], whereas learning-based flight has improved high-speed navigation and racing [3,4]; neither route alone resolves open-vocabulary grounding, long-horizon reasoning, and verified execution.

Large language models (LLMs), vision–language models (VLMs), and vision–language–action (VLA) models introduce complementary capabilities. LLMs can translate instructions into typed subgoals or formal specifications [5]; VLMs associate visual observations with open-vocabulary concepts [6]; and VLA models learn direct observation-to-action mappings [7]. General embodied models such as PaLM-E, RT-2, and SayCan demonstrate knowledge transfer, action tokenization, and language-conditioned skill selection [8–10], motivating UAV-oriented surveys and broader language-model-based task-planning frameworks [11,12]. Their deployment nevertheless creates a rate mismatch between slow probabilistic semantic inference and high-rate stabilization, while hallucination, resource limits, Sim2Real shift, and intermittent edge–cloud links introduce additional failure modes.

Recent surveys cover broad LLM–UAV applications [13], agentic UAVs [14], Transformer/LLM taxonomies [15], publication-to-practice gaps [16], UAV operations and communications [17], and aerial vision–language navigation (VLN) [18,19]. A system-level synthesis is still needed to connect cognition, continuous control, swarm coordination, deployment, formal safety, cybersecurity, and process-oriented evaluation under a common evidence protocol.

This survey therefore makes four contributions. It (i) proposes a five-dimensional taxonomy covering language understanding and task planning, navigation and control, multi-agent coordination, onboard deployment and optimization, and benchmarks and evaluation; (ii) separates semantic latency, control timing, frequency, board power, task time, hardware, and validation setting, marking unreported values as N/R; (iii) compares decoupled and end-to-end architectures together with hard-constraint and learned-safety paradigms and their failure modes; and (iv) derives research directions for lightweight hierarchical reasoning, semantically adaptive runtime assurance, communication-aware autonomy, and airworthiness-oriented validation. Figure 1 summarizes the three-layer architecture: safety-critical estimation, control, and fallback remain onboard, while edge and cloud services provide optional lower-rate perception, coordination, and semantic reasoning behind typed interfaces and independent validation.

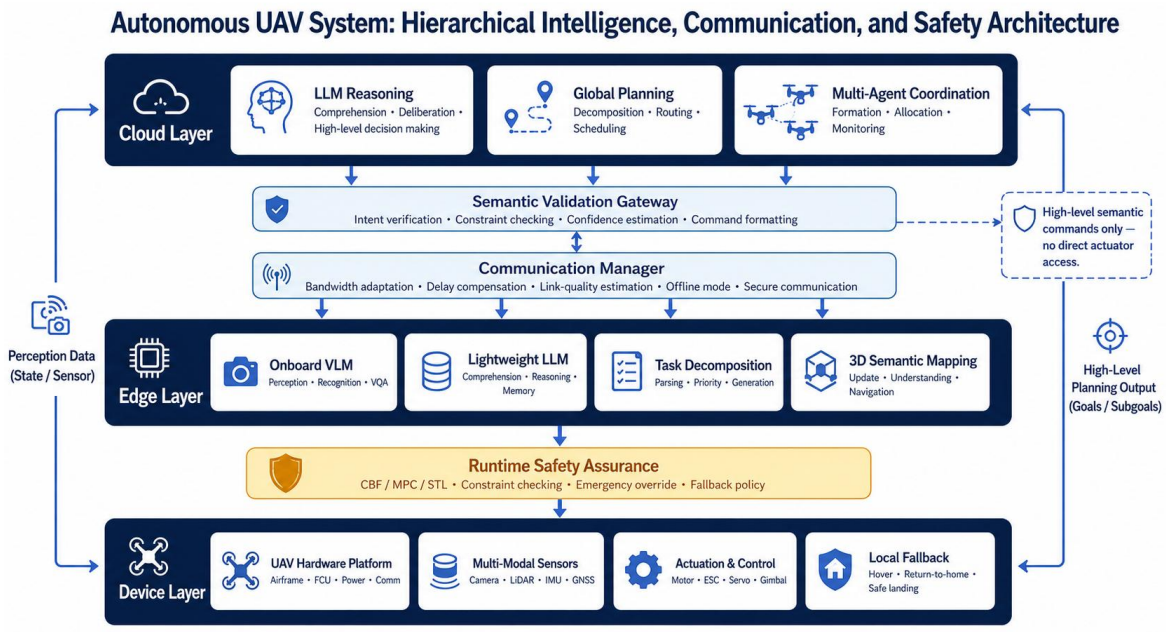

Figure 1. Three-layer LLM-enabled UAV architecture with explicit semantic validation, communication management, runtime safety assurance, and local fallback. Cloud and edge outputs remain high-level advisory inputs; actuator authority is retained by the onboard verified control path.

2. Review Scope and Evidence Audit

This survey uses a structured, topic-driven literature-selection and evidence-audit process. Sources actually consulted include arXiv and official publisher, venue, standards, and regulator pages, including IEEE Xplore, ACM Digital Library, PMLR, journal websites, and aviation-authority repositories. Relevant studies were identified through title- and topic-based lookups and backward and forward citation tracing from recent LLM–UAV and aerial vision–language navigation surveys and primary studies. The term groups below summarize the search concepts actually used; they are not presented as a single database-wide Boolean expression executed uniformly across all sources. The survey does not claim an exhaustive database census or report unsupported retrieval and screening counts.

The literature selection was guided by three groups of terms: (i) foundation-model terms, including large language model, LLM, vision–language model, VLM, vision–language–action, VLA, foundation model, and multimodal large language model; (ii) aerial-platform terms, including UAV, drone, unmanned aerial vehicle, and aerial robot; and (iii) autonomy and deployment terms, including navigation, planning, control, perception, coordination, swarm, deployment, safety, and benchmark. The main coverage window is January 2019 to July 2026. Earlier planning, control, safety, communication, and robotics studies are retained only when they provide necessary technical foundations.

Titles and abstracts were used for initial relevance checks, and primary full texts were consulted before architectural descriptions, numerical results, safety claims, or deployment evidence were included. A study is included when it directly integrates a foundation model into a UAV autonomy, interaction, networking, or multi-agent workflow, or when it introduces a UAV-specific dataset, benchmark, platform, or closely related survey with sufficient technical detail. Ground-only robotics, generic remote sensing, non-autonomous UAV applications, duplicate versions, and classical-only methods are excluded from the primary evidence set; classical planning, DRL, CBF/MPC, communication, and airworthiness studies are used only as background evidence.

For each primary study, we extract the task and scenario, model family, perception/planning/control architecture,

onboard–edge–cloud partition, hardware, branch-specific inference time, control frequency, board-level power, task outcomes, safety mechanism, communication assumptions, security/privacy considerations, and simulation/HIL/real-flight evidence. Numerical values retain the source paper's units and measurement boundary. Semantic-model latency is not conflated with control-loop frequency or mission time; board-level power is not treated as propulsion power; relative improvements are distinguished from percentage-point gains; and unreported values are marked N/R. Because hardware, environments, and success definitions differ substantially, the tables are evidence inventories rather than pooled rankings. Source-reported within-study comparisons are retained only when the baseline and evaluation protocol are explicit.

2.1. Positioning Relative to Existing Surveys

Table 1 distinguishes this work from broad application, agentic-UAV, Transformer, operations/communication, and aerial-VLN reviews. The intended contribution is the joint treatment of the autonomy stack with an auditable quantitative and deployment evidence schema, rather than a claim of being the first review.

Table 1. Positioning relative to existing LLM–UAV and aerial-VLN surveys.

Study	Primary Scope	Quantitative Treatment	Deployment/Safety Emphasis	Remaining Gap Addressed Here
Javaid et al. [13]	Broad LLM architectures and UAV applications	Mainly application-level	Opportunities and constraints	Closed-loop control, process safety, and security are not central axes.
Tian et al. [14]	Agentic UAV roadmap: model, data, memory, reasoning, tools	Resources/datasets	Agent architecture	Limited audit of latency, power, Sim2Real, and formal safety.
Kheddar et al. [15]	Transformer/LLM families across UAV applications	Benchmark/ performance tables	Efficiency and real-time	Broader Transformer scope; less emphasis on end-to-end assurance.
Chen et al. [16]	Empirical study of 997 papers, 1509 public projects, and 52 practitioner responses	Task-distribution and practice-gap statistics	Maturity, safety, cost, and adoption barriers	Does not jointly audit control-loop timing, formal runtime assurance, and process-safety metrics.
Emami et al. [17]	LLM adaptation, operations, networking, MLLMs, HITL	Tutorial comparisons	Communication, reliability, ethics	Process-safety metrics and hard/soft safety integration are secondary.
Chen et al. [18]; Xia et al. [19]	UAV/aerial VLN taxonomies, datasets, simulators, metrics	Shared-benchmark/resource synthesis	Onboard deployment and Sim2Real	Narrower than the full control, cybersecurity, privacy, and airworthiness lifecycle.
This survey	Language/task planning; navigation/control; multi-agent coordination; deployment/ optimization; benchmarks/evaluation	Separates reasoning/control timing, frequency, power, task time, hardware, and validation	Formal safety, hallucination, connectivity, attacks, privacy, assurance	Unified evidence framework for architecture trade-offs and industrial practicability.

2.2. Review Limitations

The field is evolving faster than conventional indexing, and many 2025–2026 systems remain preprints whose publication status or numerical tables may change before resubmission. Because the literature selection is topic-driven rather than an exhaustive database census, relevant studies may be missed when they appear only on project pages, use different terminology, or are released after the final update. More importantly, the reviewed studies use heterogeneous vehicles, sensors, compute platforms, action spaces, environments, and success definitions, preventing a defensible meta-analysis of latency, power, or safety. This survey therefore emphasizes source-traceable evidence, explicit validation settings, and matched-protocol comparisons rather than pooled effect sizes.

3. Semantic Understanding and Task Planning Layer: From Instruction Parsing to Spatiotemporal Cognition

LLM-enabled UAV cognition extends conventional finite-state instruction handling toward open-vocabulary grounding, memory, and machine-checkable task constraints [20]. Figure 2 organizes representative systems into language understanding and task planning, navigation and control, multi-agent coordination, onboard deployment and optimization, and benchmarks and evaluation.

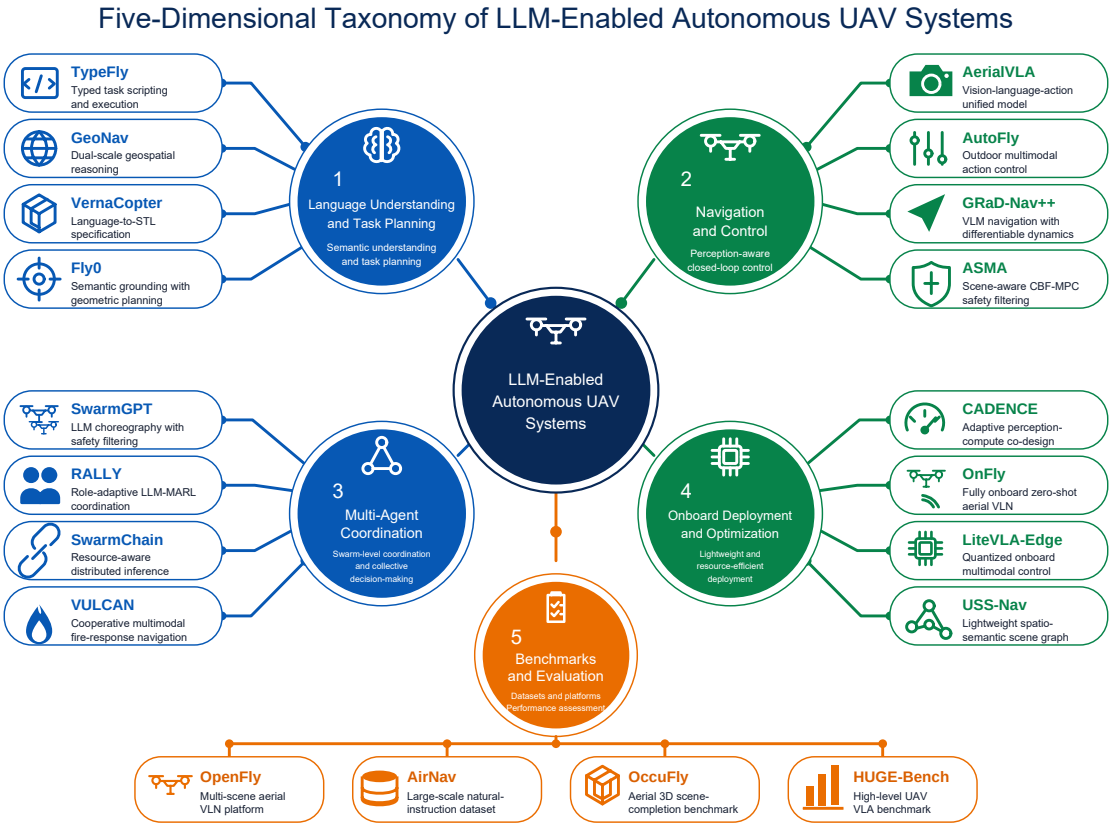

Figure 2. Five-dimensional taxonomy of LLM-enabled autonomous UAV systems. Each representative method is placed at its primary contribution node; secondary roles are discussed in the corresponding sections.

3.1. Instruction Parsing and Subgoal Decomposition

Early systems use LLMs as high-level planners that translate language into waypoints or action primitives. UAV-VLN couples instruction parsing with visual alignment [21], whereas TypeFly constrains generation through the MiniSpec language, streaming execution, probes, and replanning exceptions [22]. Such typed interfaces improve executability but do not remove cloud jitter, privacy, or syntax-coverage limits. VLN-Pilot therefore pairs an MLLM high-level operator with a finite-state low-level controller [23], illustrating a recurring design principle: probabilistic semantic decisions should not replace stabilization and obstacle-avoidance loops.

3.2. Open-Vocabulary Grounding and Semantic–Geometric Maps

APEX combines DINOv2/SAM grounding with asynchronous 3D semantic memory [24], while OnFly shares perception between a fast decision agent and a slower monitoring agent and stabilizes memory through keyframe/latest-frame caching [25]. Earlier CLIP-Fields and ConceptFusion show how language-aligned features can be embedded into 3D representations [26,27]. GeoNav adds map and landmark priors for long-range urban search [28]; SpatialFly injects geometric priors to reduce cross-view inconsistency [29]. Across these methods, the principal failure boundary remains the conversion from 2D semantic evidence to metric 3D coordinates under depth, calibration, pose, and timing errors.

3.3. Long-Horizon Memory

LongFly compresses historical views into slots and jointly encodes trajectory context [30]; grid-based memory and coarse-to-fine action refinement provide a more structured alternative [31]. Keyframe selection, asynchronous

updates, and topological–metric representations reduce repeated exploration, but fixed context budgets, pose drift, and stale observations still limit cross-session or lifelong operation. Memory should therefore be evaluated by decision benefit, update cost, freshness, and failure recovery rather than context length alone.

3.4. Formal Task Specifications

VernaCopter translates language into Signal Temporal Logic (STL) before trajectory generation [32]. Related work studies formal verification of LLM-generated code [33], while the GPT-4V system card documents multimodal-model limitations and motivates independent validation rather than treating visual-model output as a verified plan [34]. Ping et al. combine NL-to-STL learning with diagnosis and semantic repair [35]. These approaches improve traceability, but formal syntax does not itself guarantee safe flight: grounding, dynamics, solver feasibility, and real-time satisfaction remain separate obligations. The resulting design pattern is semantic reasoning followed by typed or formal intermediate representations, deterministic verification, and geometric execution.

Across the cognition layer, four error classes recur. First, linguistic ambiguity can produce an internally coherent but operationally wrong task graph. Second, semantic grounding can select the correct object class but the wrong instance or coordinate. Third, memory can preserve stale observations after the target, weather, or neighboring vehicles have changed. Fourth, a correct symbolic specification can still be infeasible for the current dynamics or actuator limits. Robust systems therefore need provenance, confidence calibration, explicit timestamps, geometric consistency checks, and reject/repair behavior at each interface. Evaluations should inject paraphrases, distractor objects, map perturbations, pose drift, observation delay, and contradictory instructions rather than reporting only average success on clean episodes.

4. Embodied Navigation and End-to-End Control: From Decoupled Planning to Safe Closed-Loop

Semantic goals must ultimately become dynamically feasible control commands. Two routes dominate: decoupled semantic–geometric pipelines and end-to-end VLA policies. Figure 3 shows the required asynchronous loop and independent runtime-assurance path.

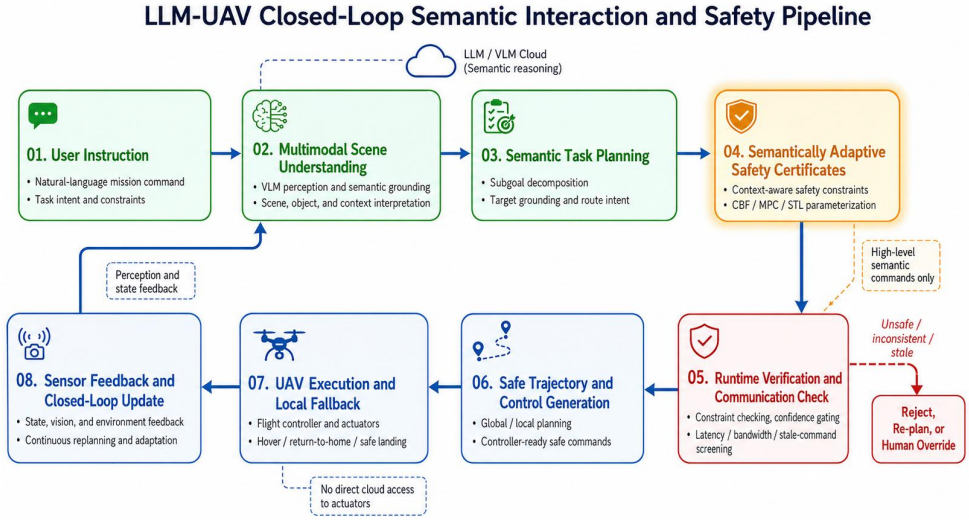

Figure 3. Closed-loop semantic interaction and runtime-safety pipeline. High-level semantic proposals are checked for grounding, freshness, communication validity, feasibility, and safety before execution; invalid or stale proposals are rejected, replanned, or redirected to local fallback.

4.1. Decoupled Semantic and Geometric Planning

Fly0 constrains the MLLM to low-frequency structured grounding, lifts accepted regions into uncertainty-aware world-frame anchors, and delegates high-rate LiDAR planning and PX4 control (open-source autopilot firmware, https://px4.io) to deterministic onboard modules [36]; a related safe-planning system combines detection, tracking, prediction, and B-spline trajectory generation [37]. The separation supports explicit checks and fallback but creates fragile interfaces at calibration, depth, pose, and clock synchronization. SoraNav switches between VLM reasoning and geometry-driven exploration when semantic navigation becomes unreliable [38]. ViSA adds structured visual prompting and verification [39]: on CityNav Test-Unseen, SR rises from 21.20% to 36.11%, a 14.91-percentage-point (70.3% relative) improvement. Relative and absolute gains must not be conflated.

4.2. End-to-End VLA Control

AerialVLA maps dual-view images and language to 3-DoF commands and landing decisions [40]; AutoFly adds pseudo-depth features and progressive multimodal/action training for outdoor velocity control [41]. Diffusion policies can model multimodal action sequences [42], while RT-1-style action tokenization links discrete semantics and physical quantities [43]. GRaD-Nav++ combines CLIP, a compact MoE policy, differentiable dynamics, and 3DGS simulation [44]; nominal policy steps are about 30 ms, periodic CLIP updates exceed 120 ms, the stack runs at 25 Hz, and trained/unseen SR drops from 83%/75% in simulation to 67%/50% on real hardware. VLA-AN reports an 8.3× throughput increase and a 98.1% maximum on one object-navigation task, not an aggregate cross-task success rate [45]. End-to-end policies reduce interfaces but remain difficult to verify under OOD perception and dynamics.

4.3. Cross-Scenario Architecture Trade-offs

The decoupled and end-to-end routes should be compared by failure containment rather than by a single success rate. Decoupled systems expose intermediate targets, maps, trajectories, and constraint violations, enabling module-level diagnostics and replacement. They are attractive for long-range inspection, search, and regulated operations in which target coordinates and trajectories must be audited. Their main weaknesses are interface accumulation, asynchronous state mismatch, calibration dependence, and repeated perception/planning cost. A semantic output can be plausible in image space yet infeasible after 3D back-projection, and a safe geometric trajectory can become stale while the model is reasoning.

End-to-end VLA policies amortize perception and action selection and can exploit task-specific visual–motor correlations that are difficult to model explicitly. They are promising for short-horizon reactive flight and compact onboard execution, especially when training data represent the deployment distribution. Their failures are less localized: action errors may arise from language, visual representation, dynamics, action discretization, or training imbalance, and confidence estimates are not automatically calibrated for physical risk. Sim2Real loss is therefore not a single domain-gap number; it can originate at sensing, rendering, vehicle dynamics, actuator response, communication, or hardware timing.

A fair cross-scenario comparison should report at least four layers. The semantic layer reports instruction/grounding accuracy and reasoning latency; the planning layer reports feasibility, path quality, replanning, and stale-plan rejection; the control layer reports update rate, tracking error, minimum separation, intervention, and fallback; and the deployment layer reports model partition, memory, board power, link conditions, and real-flight evidence. Decoupled and VLA systems may then be compared on a shared mission and hardware profile, but heterogeneous published values should remain an evidence inventory rather than a league table.

4.4. Safety Constraint Embedding: Hard Constraints, Learned Safety, and Runtime Assurance

The high-speed motion of UAVs in three-dimensional space makes safety a closed-loop property rather than an endpoint metric. A planner can produce a semantically plausible route that is nevertheless dynamically infeasible, stale, or unsafe under state-estimation error. The literature therefore spans four related but non-equivalent paradigms: hard control-theoretic constraints, reward-shaped learning, constrained reinforcement learning, and hybrid runtime shielding. Table 2 summarizes their guarantee scope and characteristic failure modes. Trade-offs among safety paradigms are discussed below.

Hard-constrained safety. For a control-affine system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$, a control barrier function (CBF) defines a safe set $\mathcal{C} = \{\mathbf{x} : h(\mathbf{x}) \geq 0\}$ and imposes

$$L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} \geq -\alpha(h(\mathbf{x})), \tag{1}$$

where $L_f h$ and $L_g h$ denote the Lie derivatives of h along f and g, respectively, and $\alpha(\cdot)$ is an extended class-$\mathcal{K}$ function. When the model, state estimate, input bounds, relative-degree assumptions, and numerical solution are valid, satisfying (1) yields forward invariance of the encoded safe set [46]. In LLM/VLM-enabled UAV systems, ASMA embeds a scene-aware CBF into MPC, while SAGE-LLM applies fuzzy-CBF verification to LLM-proposed decisions [47,48]. The principal advantage is an explicit admissible-action set and an auditable intervention. The guarantee is nevertheless conditional: perception error can define the wrong obstacle geometry, model mismatch can invalidate the derivative condition, actuator saturation can make the quadratic program infeasible, and delayed updates can allow the physical state to cross the assumed boundary before the next filter step.

Table 2. Safety paradigms for learning-enabled UAV control. Guarantee statements are conditional on each method's assumptions and should not be compared as a single cross-paper scalar.

Paradigm	Safety Encoding	Guarantee Scope	Main Strengths	Characteristic Failure Modes	Recommended Role
CBF/MPC hard filter [46,47]	State/action inequality and actuator constraints	Forward invariance or recursive feasibility when the model, state, bounds, and solver assumptions hold	Explicit veto, interpretable margin, real-time optimization	Perception/model error; relative-degree mismatch; conflicting constraints; actuator saturation; solver delay or infeasibility	Final action filter and emergency envelope
Reward-shaped DRL/VLA [49,50]	Collision, proximity, smoothness, and task terms in a scalar objective	Empirical risk reduction on the training/evaluation distribution; no hard state-wise guarantee	Flexible behavior, implicit dynamics adaptation, fast policy execution	Reward hacking; weight sensitivity; sparse hazard coverage; OOD failure; unsafe exploration	Nominal proposal policy inside a verified envelope
Constrained RL [51]	Expected cumulative cost constraint separate from reward	Near-constraint satisfaction under algorithmic and approximation assumptions; generally not per-state invariance	Separates performance from safety cost and reduces penalty tuning ambiguity	Distribution mismatch; function-approximation error; average-cost compliance can hide rare severe events	Training-time risk control combined with runtime shielding
Semantically adaptive hybrid [32,35,48]	Bounded semantic parameters instantiate CBF/MPC/STL constraints	Inherits only the guarantee of the deterministic verifier after grounding, freshness, and feasibility checks	Context-sensitive margins with explicit provenance and fallback	Hallucinated labels; ambiguous language; stale context; unsafe parameter expansion if bounds are not enforced	Preferred architecture for LLM/VLA-enabled safety-critical autonomy

UAV-VLRR illustrates a related decoupled design in which multimodal reasoning supplies target and obstacle coordinates to an NMPC controller [52]. Its multimodal subsystem runs remotely on an RTX 4090/i9 server (NVIDIA & Intel hardware) , whereas an Orange Pi 5 (Shenzhen Xunlong) executes NMPC onboard. The source reports mission times of 28 s and 30 s in two indoor scenarios, but not standalone semantic-inference latency or power. Consequently, the result demonstrates rapid execution under that protocol, not fully onboard multimodal assurance; coordinate-conversion error, server unavailability, and stale world-state estimates remain relevant failure modes.

Reward-shaped and constrained learning. Reward-shaped DRL internalizes safety through penalties such as collision cost, near-obstacle potential, control smoothness, and progress:

$$R_t = w_1 R_{\text{goal}} + w_2 R_{\text{potential}} - w_3 C_{\text{collision}} - w_4 \|\mathbf{u}_t - \mathbf{u}_{t-1}\|^2. \tag{2}$$

This formulation is flexible and can absorb unknown dynamics, but a finite penalty does not make a state unreachable. The learned policy may trade a rare collision against accumulated reward, exploit an unintended reward shortcut, or fail under distribution shift [49]. Constrained policy optimization provides a stronger intermediate formulation by optimizing reward subject to expected cost constraints and deriving near-constraint-satisfaction guarantees during policy updates [51]. These guarantees are valuable but differ from per-state forward invariance: they depend on the constrained Markov decision process, approximation quality, and training distribution. Outdoor DRL evaluations based on privileged learning and domain randomization have accumulated more than 650 m across multiple field trials, but this should not be interpreted as a single uninterrupted 650 m mission or as proof of worst-case safety [50].

Operational definition of semantically adaptive safety certificates. We use this phrase as a proposed system concept rather than an established certification term. A semantically adaptive safety certificate is a timestamped, runtime-verifiable object that maps grounded instruction, scene, and mission context to bounded parameters of a CBF, MPC constraint, or temporal-logic specification. Examples include increasing stand-off distance near people, instantiating a no-fly region from a verified map label, or tightening landing-surface constraints under low confidence. The generative model may propose semantic labels or candidate margins, but it must not directly expand the admissible set. A deterministic monitor checks provenance, confidence, parameter bounds, constraint feasibility, and freshness before the certificate can modify the safety filter; otherwise, the system retains a conservative default or executes a fallback behavior. Formal-language translation and repair mechanisms provide one pathway for converting natural-language intent into machine-checkable specifications [32,35], while CBF-based filters provide a continuous-control enforcement layer [47,48]. At present, this concept remains a system-level design paradigm; full end-to-end validation on physical UAV hardware under complex, safety-critical aerial scenarios requires further research.

The most defensible architecture is therefore hierarchical: an LLM/VLA policy proposes a task or action; a grounded semantic module instantiates only bounded safety parameters; a CBF/MPC/STL monitor accepts, repairs, or rejects the proposal; and a high-rate controller executes the verified command. This design preserves the adaptability of learned policies while keeping the final authority in a deterministic runtime-assurance layer. It also makes failure attribution clearer: semantic misgrounding affects certificate parameters, model mismatch affects the control guarantee, and infeasibility triggers an explicit degraded mode rather than an unobserved policy failure.

4.5. Perception-Computation Joint Adaptation and Dynamic Slimming

The real-time operation of end-to-end VLA and safety controllers heavily depends on onboard computational resources. CADENCE closes the loop between a dynamically slimmable monocular-depth-estimation (MDE) network and a navigation policy: the policy jointly outputs the motion action and the MDE slimming factor, including a zero-perception mode [53]. On its Jetson Orin Nano (NVIDIA Corporation, Santa Clara, CA, USA) hardware-in-the-loop testbed with Microsoft AirSim (v1.8.1) , the individual MDE configurations span 6.0–117.1 ms inference latency and 4.788–18.351 W board-level power. Relative to the static baseline used in the source study, the adaptive system reduces sensor acquisitions by 9.67%, inference latency by 74.8%, power consumption by 16.1%, and computing energy by 75.0%, while increasing navigation accuracy by 7.43%. These are HIL and board-benchmark results, not measurements of total propulsion energy or unrestricted outdoor flight. The evidence therefore supports context-aware perception–computation co-adaptation, while also illustrating why latency and power must be reported with their measurement boundary and validation setting.

To compare the core methods discussed in this section without conflating incompatible measurements, Table 3 summarizes architectural characteristics and Table 4 separates semantic/policy latency, control frequency, board power, task-level outcomes, hardware partition, and validation setting. The entries reproduce values explicitly reported by each source; N/R denotes not reported. Because the studies use different tasks, hardware, action rates, and measurement boundaries, the table is an evidence inventory rather than a cross-platform ranking.

Table 3. Comparison of core algorithms and task characteristics.

Category	Ref.	Inputs	Outputs	Scenarios	Core Innovations	Limitations/Gap
Semantic Planning	Fly0 [36], GeoNav [28], VernaCopter [32]	RGB, Text, Landmark priors	3D Waypoints, STL specs	Urban/Indoor long-range	Decoupled semantic/geometric & formal constraints	Rely on high-precision pose/prior maps
End-to-End VLA	AerialVLA [40], AutoFly [41], GRaD-Nav++ [44]	Dual-view RGB, Pseudo-depth	Continuous vel., Motor RPM	Outdoor/Racing/Avoidance	Direct mapping, Diffusion policy, Zero-shot	OOD conservatism, weather degradation
Safety Control	ASMA [47], UAV-VLRR [52], DRL-Nav [49]	RGB-D, Language, Depth	MPC trajectories, CBF constraints	Dynamic obstacles, Search & Rescue	Scene-aware CBF, Semantic NMPC	Unmodeled aerodynamic dist., High compute demand
Dynamic Adaptive	CADENCE [53], Vision-Guided Flight [50]	Monocular RGB, Stereo vision	Adaptation factor, Velocity cmds	Resource-limited/Outdoor	Perception-compute joint optimization	Textureless degradation, Domain gap

Table 4. Evidence-aware comparison of runtime, resources, task outcomes, and deployment settings.

Method	Model/Compute Partition	Reported Inference or Policy Latency	Platform	Control Rate	Board Power	Reported Task-Level Evidence	Validation Setting
CADENCE [53]	Slimmable MDE + joint navigation/adaptation DQN; onboard HIL compute	MDE: 6.0–117.1 ms across widths	Jetson Orin Nano	N/R	4.788–18.351 W	Versus static baseline: -74.8% latency, -16.1% power, -75.0% compute energy, $+7.43\%$ navigation accuracy	Microsoft AirSim + HIL; board-level benchmark
OnFly [25]	Shared-perception dual-agent VLM; fully onboard	Decision: 0.81 s; monitoring: 1.15 s with hybrid memory	Jetson Orin NX 16 GB	N/R	N/R	Simulation success increased from 26.4% to 67.8%; four onboard real-flight tasks	Simulation + fully onboard indoor/outdoor flights
Patrol Agent [54]	Florence-2-base + YOLOv8n onboard; Qwen2-72B cloud	Complete loop ∼3–4 Hz; image acquisition ∼100 ms	Processor N/R; cloud API	N/R	N/R	Patrol: 150 s/lap; caption/action 66.10%/89.47%; gated search: 45 s, ∼0.3 GB	UE4/AirSim simulation only
GRaD-Nav++ [44]	CLIP ViT-B/32 (150M) + CENet (0.5M) + MoE policy (1M); fully onboard	Policy step ∼30 ms; >120 ms when CLIP runs every 10 steps	Jetson Orin Nano; ∼6 GB memory	25 Hz	N/R (curve only)	SR trained/unseen: simulation 83%/75%; real 67%/50%	3DGS/DiffRL simulation + real UAV
UAV-VLRR [52]	Remote ChatGPT-4o + 4-bit Molmo-7B-D; onboard NMPC	Semantic and NMPC latency N/R; task time 28/30 s	RTX 4090 + i9 server; Orange Pi 5 onboard	N/R	N/R	Average task-time reduction: 33.75% vs autopilot, 54.6% vs human pilot	Indoor real flight with Vicon motion capture system; remote semantic inference

N/R means that the source paper does not explicitly report a numerical value. Latency scopes differ across rows (MDE forward pass, VLM agent step, policy step, or end-to-end task time); board power excludes propulsion power unless explicitly stated. Values are therefore not directly rank-comparable.

5. Multi-Agent Collaboration, Lightweight Deployment, and Benchmark Evaluation

Multi-UAV autonomy adds coordination and communication dependencies to the single-vehicle compute and safety problem. The key questions are whether collaboration is truly decentralized, which functions remain safe during link loss, and how resource and benchmark claims are measured.

5.1. Distributed Reasoning and Swarm Coordination

SwarmGPT combines language-generated choreography with an optimization safety filter [55]; its 200-drone evidence is simulated, whereas the 20-drone demonstration uses a central base station, Vicon, 10-Hz broadcasts, and precomputed trajectories. RALLY combines LLM priors with online role-adaptive MARL [56]. SwarmChain distributes LLM inference across resource-constrained nodes and schedules load using battery, compute, and link state [57], but synchronization remains a control-rate bottleneck. VULCAN fuses RGB-D, thermal, and radar sensing with VLM-based global planning for simulated indoor-fire response [58]. LLM-to-MARL distillation has also been explored for UAV relay-network allocation [59]. These studies progress from language orchestration to distributed inference and multimodal risk awareness, but robust operation under topology change and partition remains largely unresolved.

5.2. Onboard Lightweighting and Edge–Cloud Architecture

Patrol Agent provides a concrete but simulation-only edge–cloud example [54]. Florence-2-base (0.23B parameters) and YOLOv8n (3.2M parameters) run on board, while Qwen2-72B is accessed through a cloud API; the paper does not identify the onboard processor or report power. In its UE4/AirSim (Unreal Engine 4 by Epic Games; AirSim by Microsoft) simulation-only study, image acquisition through `simGetImages` takes about 100 ms and the complete perception–reasoning loop operates at approximately 3–4 Hz. Relative to a Qwen-VL-Max VQA baseline, Patrol Agent reduces one patrol lap from 208 to 150 s and reports caption/action correctness of 66.10%/89.47% versus 24.13%/36.67%. For target search, activating Florence-2 only after YOLO detects a person reduces search time from 64 to 45 s and reported graphics-memory use from about 4.5 to 0.3 GB. These results demonstrate conditional model activation and cloud reasoning, but they should not be interpreted as real-flight latency, power, or Sim2Real evidence.

Reported "real-time" performance is branch- and platform-specific: CADENCE measures slimmable depth inference on a Jetson Orin Nano HIL setup [53]; OnFly reports 0.81 s decision and 1.15 s monitoring steps on an Orin NX [25]; GRaD-Nav++ combines mostly 30 ms policy steps with slower periodic CLIP updates [44]. A defensible architecture retains estimation, stabilization, braking, and fallback onboard; uses edge resources for optional accelerated perception or map fusion; and reserves cloud models for lower-rate semantic services [60]. Complementary 2024 robotics and edge-AI work provides reusable efficiency baselines: OpenVLA supports low-rank adaptation and quantized serving, TinyVLA uses compact multimodal backbones with a diffusion action decoder, and Edge-LLM jointly allocates pruning sparsity and quantization bit width for resource-constrained adaptation [61–63]. UAV-specific quantization [64] targets local resource use; long-context models such as Gemini 1.5 [65] motivate selective context and cache management but do not by themselves demonstrate efficient onboard UAV inference; and spatial–semantic scene graphs [66] reduce repeated dense processing by retaining task-relevant structure. However, evaluations must report target-hardware accuracy, latency distributions, memory, board power, thermal behavior, and behavior when the large model is unavailable. Ground-robot or generic edge-model results should be treated as transferable techniques, not as UAV flight evidence.

Static model size is therefore an incomplete lightweighting metric. A small model that is invoked continuously, transmits raw images, or triggers thermal throttling may consume more mission energy than a larger model called selectively. Useful reporting includes cold/warm latency, percentile rather than mean timing, peak memory, accepted-action energy, sensor and serialization cost, throttling onset, and recovery after overload. The minimum safe configuration should remain operational when edge/cloud services fail; optional semantic capability may degrade, but stabilization, geofencing, obstacle avoidance, and safe termination must not depend on a successful remote response.

5.3. Communication-Aware Offloading, Cybersecurity, and Privacy

A cloud result should be a timestamped advisory, with action age $A = t_{\mathrm{exec}} - t_{\mathrm{obs}}$ rejected when it exceeds a validated deadline or the underlying state changes materially. A practical mode machine includes connected, degraded, partitioned, and recovery states; evaluation should report payload/update rate, delay and jitter, loss, outage duration, topology, retransmission, and stale-state behavior [59,67].

Communication-aware planning should jointly select motion and information actions. A vehicle may climb or reposition to restore a relay, postpone a low-priority semantic query, transmit a compact hazard update before

redundant imagery, or allocate a task to a neighbor with fresher state. In degraded mode, event-triggered summaries and local intent are preferable to continuous video; in partitioned mode, each UAV requires a bounded local objective, independent collision avoidance, and a termination condition. Recovery is not an immediate return to cloud authority: maps, task ownership, command sequence, and safety-relevant state must first be reconciled to prevent duplicated tasks or conflicting trajectories.

For swarm claims, logical decentralization should be distinguished from experimental infrastructure. A distributed optimizer or policy can still rely on centralized motion capture, synchronization, task generation, or trajectory broadcast. Reports should state which decisions are computed onboard, which global information is assumed, how neighbor loss is handled, and whether the demonstrated scale is simulation or physical. Performance-versus-bandwidth and performance-versus-outage curves are more informative than a single nominal-link result.

The threat surface spans prompts, multimodal observations, tool interfaces, radio links, cloud services, and collaborative learning. SafeEmbodAI motivates typed outputs, state checks, retry limits, and mission termination under malicious commands [68]; physical VLA evaluations expose OOD, typography, and adversarial-patch sensitivity [69]. Net-GPT demonstrates protocol-aware UAV-link attacks [70]. MAVLink 2 signing authenticates source and freshness but does not encrypt payloads [71], and GPS timestamp manipulation can undermine its assumptions [72]. Federated learning reduces raw-data transfer but remains vulnerable to gradient reconstruction [73] and poisoned updates; TASER provides one lightweight simulated defense direction [74]. Table 5 maps these failures to layered controls and residual risk.

Because hardware, workloads, measurement boundaries, and communication conditions differ, the literature does not support a universal Pareto frontier over model size, latency, power, OOD accuracy, and link reliability. Matched workloads or an explicit normalization protocol are required.

5.4. Benchmarks and Process-Oriented Metrics

Endpoint metrics such as NE, SR, OSR, and SPL do not reveal whether an agent reproduced a required multi-stage trajectory, collided, missed a deadline, or depended on offboard compute. The verified benchmark facts, summarized in Table 6, are as follows: OpenFly contains 100K trajectories in 18 scenes; its 16.6%–34.3% change is a test-seen ablation, not an unseen-scene gain [75]. AirNav currently contains 137K samples and uses fixed-altitude Tello experiments with offboard inference [76]. OccuFly provides nine scenes and more than 20K samples, with $<10\%$ referring to manually annotated source images [77]. SkyFind contains 1,015,638 target–expression pairs and no verified audio modality [78].

A unified evaluation record should group metrics into five families. Task metrics include SR and task-specific accuracy; efficiency metrics include SPL, path length, time, and energy; process metrics include TCR, subgoal completion, intervention, and replanning; safety metrics include collision rate, minimum separation, constraint violation, and fallback success; deployment metrics include latency distributions, frequency, memory, power, communication load, and thermal behavior. Sim2Real evidence should additionally state simulator, randomization, calibration, vehicle, payload, environment, wind/lighting, localization infrastructure, number of trials, and whether inference is onboard.

No single composite score should silently combine these families. For example, CSPL makes collision-free path efficiency visible but does not capture missed inspection coverage, communication dependence, or compute overruns. Conversely, a high TCR can coexist with terminal failure. Papers should publish the component metrics and confidence intervals where repeated trials are available, together with failure categories rather than only mean success.

HUGE-Bench defines trajectory coverage from $d_i = \min_{\mathbf{q}\in T^{\mathrm{pred}}} \|\mathbf{p}_i - \mathbf{q}\|_2$ as

$$\mathrm{TCR@}\tau = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(d_i \le \tau), \quad \tau \in \{1, 2, 5\}\,\mathrm{m}, \tag{3}$$

and collision-aware path efficiency as

$$\mathrm{CSPL} = S(1 - C)\frac{\ell}{\max(\ell, p)}, \tag{4}$$

where S is terminal success, C indicates any collision, and ℓ, p are reference and executed path lengths [79]. These metrics expose process failures hidden by SR. PiLoT reports $>$25 FPS on Jetson Orin and about 1.37 m mean error over a field sequence exceeding 10 km, but these are localization-pipeline results [80]. Bearing-UAV addresses vision-only position/heading estimation [81]; Isaac Lab (NVIDIA) supports scalable simulation [82], yet Sim2Real claims still require real-hardware disturbances and failure statistics.

Table 5. Threat–failure–countermeasure matrix for LLM-driven UAV deployment.

Source	Physical Failure Chain	Detection Evidence	Layered Controls	Residual Limitation
Hallucination/tool error/misgrounding [68]	False object/coordinate → invalid waypoint/mode → collision or mission loss	Schema, map/depth disagreement, reachability, confidence, repeated repair	Typed actions; coordinate, reachability, geofence, and CBF/MPC checks; retry cap; fallback	Plausible but wrong grounding may pass; fallback conditions also require validation.
Prompt/visual injection, patch, OOD [69]	Observation changes goal/action distribution	Cross-sensor/action inconsistency; OOD or patch indicators	Trusted instruction channel; provenance; redundant perception; adversarial tests; bounded semantic authority	Detectors can be evaded or reject rare valid scenes.
MITM, spoofing, replay, clock attack [70–72]	Corrupted telemetry/command reaches planner or actuator	Signature, age, or sequence failure; impossible kinematics; clock disagreement	Signing plus encrypted tunnel; secure keys; independent time; allowlists; state consistency	Signing is not encryption; compromised keys/endpoints remain.
Loss, jitter, or partition [59,67]	Stale cloud/neighbor state causes delayed or conflicting motion	Action age; heartbeat, queue, topology, or map-age indicators	Deadline-aware offload; compressed events; stale rejection; local avoidance; partition/recovery modes	Local safety may sacrifice global coordination or mission success.
Gradient leakage/poisoning [73,74]	Private data reconstructed or backdoor propagated	Update anomaly, validation drift, trigger behavior, provenance gap	Secure aggregation; clipping/DP; robust aggregation; signed versions; tests and rollback	Privacy–utility trade-off and adaptive/colluding attackers.

Table 6. Representative UAV datasets, benchmarks, and evaluation protocols.

Resource	Task	Scale/Env.	Metrics	Transfer Evidence	Interpretation Boundary
OpenFly [75]	Aerial VLN	100K trajectories; 18 scenes; UE, GTA V, Google Earth, 3DGS	NE, SR, OSR, SPL	23 Q250 outdoor tasks; external-PC inference	Separate test-seen ablation from test-unseen; not fully onboard.
AirNav [76]	Real-data UAV VLN	137K samples; urban data; 10 personas	NE, SR, OSR, SPL	200 fixed-altitude Tello tasks; offboard inference	Discrete planar actions and limited geography.
OccuFly [77]	Aerial SSC/depth	9 real scenes; >20K samples; seasons/altitudes	IoU/mIoU; depth errors	Camera reconstruction and cross-season data	<10% is manually annotated image fraction, not error.
SkyFind [78]	Referring expressions	1,015,638 pairs; 35,599 images; 352,910 objects	Box localization	Real imagery; maritime shift test	Grounding only; not navigation, speech, or control.
HUGE-Bench [79]	High-level VLA	4 real-scene digital twins; 8 tasks; 2.56M m	TCR, SR, CR, CSPL	Collision-aware simulation	Static; no link impairment or physical-flight validation.
PiLoT [80]	Geo-localization	Pixel-to-3D registration; synthetic + real	Pose/target error, recall, FPS	>25 FPS on Orin; >10 km sequence	Localization-pipeline, not end-to-end navigation.

5.5. Cross-Scenario Deployment Synthesis

The appropriate architecture depends on mission criticality, horizon, connectivity, and the cost of a wrong semantic decision. Infrastructure inspection and corridor missions favor auditable work-order interpretation while metric mapping, trajectory optimization, geofencing, and continuous control remain deterministic. Evidence should emphasize coverage, revisit avoidance, grounding accuracy, localization dependence, timing, energy, and safe recovery. In these settings, large models are better reserved for exception handling and mission replanning than direct continuous control.

Search-and-rescue and disaster response require open-vocabulary grounding and dynamic task reprioritization, but they also expose the largest semantic and communication uncertainty. Multimodal reasoning can identify victims, hazards, and access routes, while local navigation must remain capable of operating under smoke, low visibility, GNSS loss, or link partition. Evaluation should include false hazard/victim detections, delayed updates, contradictory observations, sensor dropout, and whether a safe local behavior remains available when the semantic service fails. A high endpoint success rate under stable connectivity is insufficient for this scenario.

Short-horizon agile flight can benefit most from compact end-to-end or hybrid VLA policies because visual–motor correlations and fast reaction dominate long-form reasoning. Success should be reported with action rate, percentile latency, tracking error, minimum separation, OOD conditions, and safety-filter intervention. For swarm missions, semantic task allocation may be global while collision avoidance and partition behavior remain local. Claims of scalability should separate simulated agent count, physically flown vehicles, centralized infrastructure, and onboard decision authority.

Across scenarios, industrial practicability can be summarized by three gates. The functional gate asks whether the architecture completes the mission on representative hardware and environments. The resilience gate asks whether safety and bounded task behavior persist under sensor, compute, link, and model failures. The assurance gate asks whether requirements, data/model versions, interfaces, monitors, tests, and operational changes are traceable. An LLM–UAV system should not be described as deployment-ready when it passes only the first gate.

6. Research Roadmap and Industrialization Path

The evidence points to four coupled transitions summarized in Table 7.

Table 7. Research roadmap and deployment gates for LLM–UAV systems.

Direction	Technical Target	Required Evidence	Deployment Gate
Hierarchical lightweight reasoning	Deterministic flight kernel, compact onboard policy, event-triggered escalation to larger semantic models	Branch latency distributions, memory, board power/energy, thermal throttling, escalation/rejection rate, compression loss, recovery	Stable collision-aware flight when the largest model is late, unavailable, or rejected.
Bounded semantic adaptation	Ground typed semantics into pre-verified CBF/MPC/STL parameter families; never remove baseline constraints	Feasibility/intervention rate, realized margin, false-negative/false-positive behavior, solver delay, stale/contradictory-language tests	Uncertain semantics only tighten or retain the safe envelope; deterministic fallback handles infeasibility.
Communication-aware autonomy	Include link state, information age, relay availability, and communication energy in planning	Performance versus bandwidth/outage, payload/update rate, loss/jitter, action age, partition/recovery success	Local safety and bounded task behavior survive network partition without stale remote authority.
Assurance and secure lifecycle	Trace operational hazards to requirements, data/model versions, runtime containment, cybersecurity, and change control	Coverage/provenance, independent verification, adversarial/security tests, rollback, monitoring, authority-agreed compliance evidence	Safety-critical authority expands only after containment and degraded modes are demonstrated.

A deployable hierarchy should keep state estimation, stabilization, actuator limits, emergency braking, and geofencing in a deterministic onboard kernel; use a compact perception–action model for reactive flight; and invoke larger semantic models only at keyframes, uncertainty triggers, or mission transitions. Quantization, pruning, distillation, MoE routing, token/keyframe reduction, cache compression, and early exit should be evaluated by mission-level energy, thermal behavior, accepted-action latency, and graceful degradation—not parameter count alone.

Semantically adaptive safety certificates should be bounded, timestamped instances of pre-verified CBF/MPC/STL or reachability constraints. Grounded labels may tighten margins or select a constraint family, but a generative model must not delete baseline constraints or expand a certified envelope. Evaluation should include infeasibility, intervention, minimum separation, fallback success, and counterfactual perturbations of language, coordinates, labels, confidence, and timestamps. Communication likewise belongs in the planning state: degraded links require event-triggered summaries; partitions require local bounded objectives; reconnection requires state/intent reconciliation before remote authority returns.

Industrial deployment should be framed as a lifecycle assurance case. ARP4754B/ED-79B structure aircraft/system development [83, 84], while the active FAA AC 20-174 recognizes ARP4754A as an acceptable method, so the certification basis must be agreed with the authority [85]. AC 20-115D recognizes DO-178C/ED-12C for airborne software assurance [86]; these objectives suit deterministic kernels, monitors, interfaces, tools, and configuration control but do not automatically certify an adaptive generative controller. EASA's AI Roadmap 2.0 and AI Concept Paper Issue 2 provide evolving learning-assurance and human–AI guidance [87, 88]. DO-326B/ED-202B integrate intentional electronic threats into the airworthiness-security process [89, 90]. A credible near-term path therefore confines LLM/VLM functions to advisory or bounded high-level roles and expands authority only with verified degraded modes, cybersecurity, controlled updates, rollback, and authority-approved evidence.

A staged path makes this principle operational. Stage 1 permits language explanation, mission drafting, and operator support while deterministic software retains all flight authority. Stage 2 permits typed targets, task allocation, or waypoint proposals after independent grounding, freshness, feasibility, and safety checks. Stage 3 may permit bounded autonomous replanning only after evidence covers OOD conditions, compute/link loss, monitor independence, fallback transitions, update security, and configuration change. Every model weight, adapter, prompt template, quantization setting, dataset revision, and safety-monitor version must be configuration-controlled; retraining or compression is a behavioral change that requires impact analysis rather than a routine software update.

The roadmap also implies a minimum reporting package for future prototypes: a functional architecture and trust boundaries; branch-specific compute and communication measurements; explicit safety assumptions; a failure-injection matrix; simulation, HIL, and real-flight separation; data/model provenance; and a statement of which functions remain available after model, sensor, or link degradation. This evidence will make future surveys and certification discussions more comparable than isolated endpoint scores.

7. Conclusions

This survey connects LLM-enabled UAV cognition with planning, control, swarm coordination, deployment, safety, and evaluation. Its quantitative audit shows that semantic latency, control timing, frequency, board power, task time, and Sim2Real evidence must remain separate; heterogeneous studies do not support universal thresholds or a common Pareto ranking. Learned policies offer adaptation but remain vulnerable to misspecification and distribution shift, whereas CBF/MPC/STL guarantees remain conditional on perception, dynamics, timing, input, and solver assumptions.

We define a semantically adaptive safety certificate as a bounded, timestamped, runtime-verifiable constraint parameterized by grounded semantics but enforced outside the generative model. The practical architecture combines typed actions, deterministic validation, stale-command rejection, local fallback, and explicit connected/degraded/partitioned/recovery modes. Industrialization requires lifecycle evidence linking hazards to data/model provenance, configuration control, independent verification, cybersecurity, rollback, and in-service monitoring. Because many 2024–2026 systems remain preprints, publication status and numerical claims must be rechecked immediately before submission.

Author Contributions

H.W.: Conceptualization, methodology, investigation, and original draft; M.Z.: Supervision, validation, and review/editing. All authors have read and agreed to the published version of the manuscript.

Funding

This paper was jointly supported by the National Natural Science Foundation of China under Grant Nos. 62373262 and 62403336.

Data Availability Statement

All datasets, benchmarks, and source studies analyzed in this survey are publicly available and cited in the reference list. No new datasets were generated during this study.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used OpenAI to assist manuscript proofreading and text checking. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; et al. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. https://doi.org/10.1109/access.2019.2909530.
2. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. https://doi.org/10.1177/0278364911406761.
3. Loquercio, A.; Kaufmann, E.; Ranftl, R.; et al. Learning high-speed flight in the wild. *Sci. Robot.* **2021**, *6*, eabg5810. https://doi.org/10.1126/scirobotics.abg5810.
4. Kaufmann, E.; Bauersfeld, L.; Loquercio, A.; et al. Champion-level drone racing using deep reinforcement learning. *Nature* **2023**, *620*, 982–987. https://doi.org/10.1038/s41586-023-06419-4.
5. Lykov, A.; Serpiva, V.; Khan, M.H.; et al. CognitiveDrone: A VLA Model and Evaluation Benchmark for Real-Time Cognitive Task Solving and Reasoning in UAVs. *arXiv* **2025**, arXiv:2503.01378.
6. Zhang, Y.; Yu, H.; Xiao, J.; et al. Grounded Vision-Language Navigation for UAVs with Open-Vocabulary Goal Understanding. *arXiv* **2025**, arXiv:2506.10756.
7. Serpiva, V.; Lykov, A.; Myshlyaev, A.; et al. RaceVLA: VLA-based Racing Drone Navigation with Human-like Behaviour. *arXiv* **2025**, arXiv:2503.02572.
8. Driess, D.; Xia, F.; Sajjadi, M.S.M.; et al. PaLM-E: An Embodied Multimodal Language Model. In Proceedings of the 40th International Conference on Machine Learning (ICML), Honolulu, HI, USA, 23–29 July 2023.
9. Zitkovich, B.; Yu, T.; Xu, S.; et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In Proceedings of the 7th Conference on Robot Learning (CoRL), Atlanta, GA, USA, 6–9 November 2023; pp. 2165–2183.
10. Ahn, M.; Brohan, A.; Brown, N.; et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In Proceedings of the Conference on Robot Learning (CoRL), Auckland, New Zealand, 14–18 December 2022.
11. Peng, C.J.; Liu, C.T.; Liao, I.B.; et al. A Survey of Deep Learning for Unmanned Aerial Vehicle. In Proceedings of the 2025 International Automatic Control Conference, CACS 2025, Hsinchu, Taiwan, 5–8 November 2025; pp. 1–6.
12. Bian, S.; Zhang, Y.; Tian, G.; et al. Large language model-based task planning for service robots: A review. *arXiv* **2025**, arXiv:2510.23357.
13. Javaid, S.; Fahim, H.; He, B.; et al. Large Language Models for UAVs: Current State and Pathways to the Future. *IEEE Open J. Veh. Technol.* **2024**, *5*, 1166–1192. https://doi.org/10.1109/ojvt.2024.3446799.
14. Tian, Y.; Lin, F.; Li, Y.; et al. UAVs meet LLMs: Overviews and perspectives towards agentic low-altitude mobility. *Inf. Fusion* **2025**, *122*, 103158. https://doi.org/10.1016/j.inffus.2025.103158.
15. Kheddar, H.; Habchi, Y.; Ghanem, M.C.; et al. Recent Advances in Transformer and Large Language Models for UAV Applications. *arXiv* **2025**, arXiv:2508.11834.
16. Chen, Y.; Que, X.; Zhang, J.; et al. When Large Language Models Meet UAV Projects: An Empirical Study from Developers' Perspective. *arXiv* **2025**, arXiv:2509.12795.
17. Emami, Y.; Zhou, H.; Reddy, R.; et al. Large Language Model-Assisted UAV Operations and Communications: A Multifaceted Survey and Tutorial. *arXiv* **2026**, arXiv:2602.19534.
18. Chen, H.; Zheng, J.; Yang, S.; et al. Vision-and-Language Navigation for UAVs: Progress, Challenges, and a Research Roadmap. *arXiv* **2026**, arXiv:2604.13654.
19. Xia, X.; Zhou, L.; Tang, Y.; et al. Vision-Language Navigation for Aerial Robots: Towards the Era of Large Language Models. *arXiv* **2026**, arXiv:2604.07705.
20. Kumar, V.; Michael, N. Opportunities and challenges with autonomous micro aerial vehicles. *Int. J. Robot. Res.* **2012**, *31*, 1279–1291. https://doi.org/10.1177/0278364912455954.
21. Saxena, P.; Raghuvanshi, N.; Goveas, N. UAV-VLN: End-to-End Vision Language guided Navigation for UAVs. In Proceedings of the 2025 European Conference on Mobile Robots (ECMR), Padua, Italy, 2–5 September 2025; pp. 1–6. https://doi.org/10.1109/ecmr65884.2025.11163198.
22. Chen, G.; Yu, X.; Ling, N.; et al. TypeFly: Flying Drones with Large Language Model. *arXiv* **2023**, arXiv:2312.14950.
23. Dominguez-Dager, B.; Suescun-Ferrandiz, S.; Escalona, F.; et al. VLN-Pilot: Large Vision-Language Model as an Autonomous Indoor Drone Operator. *arXiv* **2026**, arXiv:2602.05552.
24. Zhang, D.; Chen, P.; Xia, X.; et al. APEX: A Decoupled Memory-based Explorer for Asynchronous Aerial Object Goal Navigation. *arXiv* **2026**, arXiv:2602.00551.

25. Zheng, G.; Ban, Y.; Zhang, M.; et al. OnFly: Onboard Zero-Shot Aerial Vision-Language Navigation toward Safety and Efficiency. *arXiv* **2026**, arXiv:2603.10682.
26. Shafiullah, N.M.; Paxton, C.; Pinto, L.; et al. CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory. In Proceedings of the Robotics: Science and Systems (RSS), Daegu, Korea, 10–14 July 2023. https://doi.org/10.15607/rss.2023.xix.074.
27. Jatavallabhula, K.M.; Kuwajerwala, A.; Gu, Q.; et al. ConceptFusion: Open-set Multimodal 3D Mapping. In Proceedings of the Robotics: Science and Systems (RSS), Daegu, Korea, 10–14 July 2023. https://doi.org/10.15607/rss.2023.xix.066.
28. Xu, H.; Hu, Y.; Gao, C.; et al. GeoNav: Empowering MLLMs with Dual-Scale Geospatial Reasoning for Language-Goal Aerial Navigation. *Pattern Recognit.* **2026**, *177*, 113365. https://doi.org/10.1016/j.patcog.2026.113365.
29. Jiang, W.; Huang, K.; Wang, L.; et al. SpatialFly: Implicit 3D Prior-Guided Visual Reparameterization for Continuous UAV Vision-and-Language Navigation. *arXiv* **2026**, arXiv:2603.21046.
30. Jiang, W.; Wang, L.; Huang, K.; et al. LongFly: Long-Horizon UAV Vision-and-Language Navigation with Spatiotemporal Context Integration. *arXiv* **2025**, arXiv:2512.22010.
31. Ding, X.; Gao, J.; Pan, C.; et al. History-Enhanced Two-Stage Transformer for Aerial Vision-and-Language Navigation. In Proceedings of the AAAI Conference on Artificial Intelligence, Singapore, 20–27 January 2026; pp. 18225–18233. https://doi.org/10.1609/aaai.v40i22.38885.
32. van de Laar, T.; Zhang, Z.; Qi, S.; et al. VernaCopter: Disambiguated Natural-Language-Driven Robot via Formal Specifications. *arXiv* **2024**, arXiv:2409.09536.
33. Councilman, A.; Fu, D.J.; Gupta, A.; et al. Towards Formal Verification of LLM-Generated Code from Natural Language Prompts. *arXiv* **2025**, arXiv:2507.13290.
34. OpenAI. *GPT-4V(ision) System Card*; Technical Report; OpenAI: San Francisco, CA, USA, 2023.
35. Ping, Y.; Ding, H.; Liang, T.; et al. LLM-Enabled Low-Altitude UAV Natural Language Navigation via Signal Temporal Logic Specification Translation and Repair. *arXiv* **2026**, arXiv:2603.27583.
36. Xu, Z.; Lu, Y.; Bao, W.; et al. Fly0: Persistent Metric Anchoring for Zero-Shot Aerial Vision-Language Navigation. *arXiv* **2026**, arXiv:2602.15875.
37. Zhong, J.; Li, M.; Chen, Y.; et al. A Safer Vision-Based Autonomous Planning System for Quadrotor UAVs with Dynamic Obstacle Trajectory Prediction and Its Application with LLMs. In Proceedings of the 2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), Waikoloa, HI, USA, 1–6 January 2024; pp. 920–929. https://doi.org/10.1109/wacvw60836.2024.00131.
38. Song, H.; Yadav, R.D.; Guo, C.; et al. SoraNav: Adaptive UAV Task-Centric Navigation via Zeroshot VLM Reasoning. *arXiv* **2025**, arXiv:2510.25191.
39. Tong, H.; Dong, X.; Ma, X.; et al. ViSA-Enhanced Aerial VLN: A Visual-Spatial Reasoning Enhanced Framework for Aerial Vision-Language Navigation. *arXiv* **2026**, arXiv:2603.08007.
40. Xu, P.; Deng, Z.; Deng, J.; et al. AerialVLA: A Vision-Language-Action Model for UAV Navigation via Minimalist End-to-End Control. *arXiv* **2026**, arXiv:2603.14363.
41. Sun, X.; Si, W.; Ni, W.; et al. AutoFly: Vision-Language-Action Model for UAV Autonomous Navigation in the Wild. *arXiv* **2026**, arXiv:2602.09657.
42. Chi, C.; Xu, Z.; Feng, S.; et al. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *Int. J. Robot. Res.* **2025**, *44*, 1684–1704. https://doi.org/10.1177/02783649241273668.
43. Brohan, A.; Brown, N.; Carbajal, J.; et al. RT-1: Robotics Transformer for Real-World Control at Scale. In Proceedings of the Robotics: Science and Systems (RSS), Daegu, Korea, 10–14 July 2023. https://doi.org/10.15607/rss.2023.xix.025.
44. Chen, Q.; Gao, N.; Huang, S.; et al. GRaD-Nav++: Vision-Language Model Enabled Visual Drone Navigation with Gaussian Radiance Fields and Differentiable Dynamics. *IEEE Robot. Autom. Lett.* **2026**, *11*, 1418–1425. https://doi.org/10.1109/lra.2025.3643290.
45. Wu, Y.; Zhu, M.; Li, X.; et al. VLA-AN: An Efficient and Onboard Vision-Language-Action Framework for Aerial Navigation in Complex Environments. *arXiv* **2025**, arXiv:2512.15258.
46. Ames, A.D.; Xu, X.; Grizzle, J.W.; et al. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Autom. Control.* **2017**, *62*, 3861–3876. https://doi.org/10.1109/tac.2016.2638961.
47. Sanyal, S.; Roy, K. ASMA: An Adaptive Safety Margin Algorithm for Vision-Language Drone Navigation via Scene-Aware Control Barrier Functions. *IEEE Robot. Autom. Lett.* **2025**, *10*, 9232–9239. https://doi.org/10.1109/lra.2025.3592138.
48. Zhao, W.; Zhao, Y.; Liu, G.; et al. SAGE-LLM: Towards Safe and Generalizable LLM Controller with Fuzzy-CBF Verification and Graph-Structured Knowledge Retrieval for UAV Decision. *arXiv* **2026**, arXiv:2602.23719.
49. Shao, S.; Xu, Z. Autonomous UAV Path Planning in Complex Environments Using Deep Reinforcement Learning. In Proceedings of the 2025 9th International Workshop on Control Engineering and Advanced Algorithms, Xi'an, China, 31 October–2 November 2025; pp. 69–72. https://doi.org/10.1109/iwceaa68605.2025.11352075.
50. Dutta, S.; Gupta, A.; Saran, V.; et al. Vision-Guided Outdoor Flight and Obstacle Evasion via Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2026**, *11*, 1202–1209. https://doi.org/10.1109/lra.2025.3641120.
51. Achiam, J.; Held, D.; Tamar, A.; et al. Constrained Policy Optimization. In Proceedings of the 34th International Conference

on Machine Learning (ICML), Sydney, NSW, Australia, 6–11 August 2017; pp. 22–31.

52. Yaqoot, Y.; Mustafa, M.A.; Sautenkov, O.; et al. UAV-VLRR: Vision-Language Informed NMPC for Rapid Response in UAV Search and Rescue. In Proceedings of the 2025 IEEE Intelligent Vehicles Symposium (IV), Cluj-Napoca, Romania, 22–25 June 2025; pp. 1195–1200. https://doi.org/10.1109/iv64158.2025.11097824.
53. Johnsen, T.K.; Levorato, M. CADENCE: Context-Adaptive Depth Estimation for Navigation and Computational Efficiency. *arXiv* **2026**, arXiv:2604.07286.
54. Yuan, Z.; Xie, F.; Ji, T. Patrol Agent: An Autonomous UAV Framework for Urban Patrol Using on Board Vision Language Model and on Cloud Large Language Model. In Proceedings of the 2024 6th International Conference on Robotics and Computer Vision (ICRCV), Wuxi, China, 20–22 September 2024; pp. 237–242. https://doi.org/10.1109/icrcv62709.2024.10758606.
55. Schuck, M.; Dahanaggamaarachchi, D.O.; Sprenger, B.; et al. SwarmGPT: Combining Large Language Models With Safe Motion Planning for Drone Swarm Choreography. *IEEE Robot. Autom. Lett.* **2025**, *10*, 12237–12244. https://doi.org/10.1109/lra.2025.3619745.
56. Wang, Z.; Li, R.; Li, S.; et al. RALLY: Role-Adaptive LLM-Driven Yoked Navigation for Agentic UAV Swarms. *IEEE Open J. Veh. Technol.* **2025**, *6*, 2693–2708. https://doi.org/10.1109/ojvt.2025.3610852.
57. Han, B.; Chen, Y.; Li, J.; et al. SwarmChain: Collaborative LLM Inference for UAV Swarm Control. *IEEE Internet Things Mag.* **2025**, *8*, 64–71. https://doi.org/10.1109/miot.2025.3575886.
58. Liu, S.; Yan, Q. VULCAN: Vision-Language-Model Enhanced Multi-Agent Cooperative Navigation for Indoor Fire-Disaster Response. *arXiv* **2026**, arXiv:2604.12831.
59. Xu, Y.; Zha, J.; Hong, W.; et al. Scalable UAV Multi-Hop Networking via Multi-Agent Reinforcement Learning with Large Language Models. *arXiv* **2025**, arXiv:2505.08448.
60. Chen, P.; Ouyang, T.; Luo, K.; et al. CoDrone: Autonomous Drone Navigation Assisted by Edge and Cloud Foundation Models. *IEEE Internet Things J.* **2026**, *13*, 5593–5609. https://doi.org/10.1109/jiot.2025.3648721.
61. Kim, M.J.; Pertsch, K.; Karamcheti, S.; et al. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv* **2024**, arXiv:2406.09246.
62. Wen, J.; Zhu, Y.; Li, J.; et al. TinyVLA: Towards Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation. *arXiv* **2024**, arXiv:2409.12514.
63. Yu, Z.; Wang, Z.; Li, Y.; et al. EDGE-LLM: Enabling Efficient Large Language Model Adaptation on Edge Devices via Unified Compression and Adaptive Layer Voting. In Proceedings of the 61st ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 23–27 June 2024. https://doi.org/10.1145/3649329.3658473.
64. Williams, J.; Gupta, K.D.; George, R.; et al. LiteVLA-Edge: Quantized On-Device Multimodal Control for Embedded Robotics. *arXiv* **2026**, arXiv:2603.03380.
65. Reid, M.; Savinov, N.; Teplyashin, D.; et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv* **2024**, arXiv:2403.05530.
66. Gai, W.; Gao, Y.; Zhou, Y.; et al. USS-Nav: Unified Spatio-Semantic Scene Graph for Lightweight UAV Zero-Shot Object Navigation. *arXiv* **2026**, arXiv:2602.00708.
67. Zhao, Z.; Chen, C.; Shi, H.; et al. Towards Robust Multi-UAV Collaboration: MARL with Noise-Resilient Communication and Attention Mechanisms. *arXiv* **2025**, arXiv:2503.02913.
68. Zhang, W.; Kong, X.; Braunl, T.; et al. SafeEmbodAI: A Safety Framework for Mobile Robots in Embodied AI Systems. *arXiv* **2024**, arXiv:2409.01630.
69. Cheng, H.; Xiao, E.; Wang, Y.; et al. Manipulation Facing Threats: Evaluating Physical Vulnerabilities in End-to-End Vision Language Action Models. *arXiv* **2024**, arXiv:2409.13174.
70. Piggott, B.; Patil, S.; Feng, G.; et al. Net-GPT: A LLM-Empowered Man-in-the-Middle Chatbot for Unmanned Aerial Vehicle. In Proceedings of the 2023 IEEE/ACM Symposium on Edge Computing (SEC), Wilmington, DE, USA, 6–9 December 2023; pp. 287–293. https://doi.org/10.1145/3583740.3626809.
71. MAVLink. Development Team. Message Signing (Authentication). Available online: https://mavlink.io/en/guide/message_signing.html (accessed on 25 July 2026).
72. Colton, Z.; Oracevic, A.; Dilek, S. Timestamp Manipulation-Based GPS Spoofing Attacks on the MAVLink 2.0 Protocol for UAV Communication: An Empirical Study. *IEEE Trans. Commun.* **2025**, *74*, 2564–2579. https://doi.org/10.1109/tcomm.2025.3644474.
73. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
74. Huang, S.; Yang, S. TASER: Task-Aware Spectral Energy Refine for Backdoor Suppression in UAV Swarms Decentralized Federated Learning. *arXiv* **2026**, arXiv:2603.10075.
75. Gao, Y.; Li, C.; You, Z.; et al. OpenFly: A Comprehensive Platform for Aerial Vision-Language Navigation. In Proceedings of the International Conference on Learning Representations (ICLR), Rio de Janeiro, Brazil, 23–27 April 2026.
76. Cai, H.; Rao, Y.; Huang, L.; et al. AirNav: A Large-Scale UAV Vision-and-Language Navigation Dataset with Natural and Diverse Instructions. *arXiv* **2026**, arXiv:2601.03707.

77. Gross, M.; Matha, S.B.; Fahmy, A.; et al. OccuFly: A 3D Vision Benchmark for Semantic Scene Completion from the Aerial Perspective. *arXiv* **2025**, arXiv:2512.20770.
78. Wang, K.; Wu, G.; Fu, X.; et al. SkyFind: A Large-Scale Benchmark Unveiling Referring Expression Comprehension for UAV. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 9859–9875. https://doi.org/10.1109/tpami.2026.3681112.
79. Guo, J.; Chen, Z.; Li, Z.; et al. HUGE-Bench: A Benchmark for High-Level UAV Vision-Language-Action Tasks. *arXiv* **2026**, arXiv:2603.19822.
80. Cheng, X.; Wang, L.; Liu, Y.; et al. PiLoT: Neural Pixel-to-3D Registration for UAV-based Ego and Target Geo-localization. *arXiv* **2026**, arXiv:2603.20778.
81. Liu, K.; Zhou, H.; Xu, R.; et al. Beyond Matching to Tiles: Bridging Unaligned Aerial and Satellite Views for Vision-Only UAV Navigation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Denver, CO, USA, 3–7 June 2026; pp. 5359–5368.
82. Mittal, M.; Roth, P.; Tigue, J.; et al. Isaac Lab: A GPU-Accelerated Simulation Framework for Multi-Modal Robot Learning. *arXiv* **2025**, arXiv:2511.04831.
83. SAE International. *ARP4754B: Guidelines for Development of Civil Aircraft and Systems*; SAE International: Warrendale, PA, USA, 2023.
84. EUROCAE. *ED-79B: Guidelines for Development of Civil Aircraft and Systems*; EUROCAE: Saint-Denis, France, 2023.
85. Federal Aviation Administration. *AC 20-174: Development of Civil Aircraft and Systems*; United States Department of Transportation: Washington, DC, USA, 2011. Available online: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1019527 (accessed on 25 July 2026).
86. Federal Aviation Administration. *AC 20-115D: Airborne Software Development Assurance Using EUROCAE ED-12() and RTCA DO-178()*; United States Department of Transportation: Washington, DC, USA, 2017. Available online: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1032046 (accessed on 25 July 2026).
87. European Union Aviation Safety Agency. *EASA Artificial Intelligence Roadmap 2.0: A Human-Centric Approach to AI in Aviation*; European Union Aviation Safety Agency: Cologne, Germany, 2023.
88. European Union Aviation Safety Agency. *EASA Artificial Intelligence Concept Paper Issue 2: Guidance for Level 1 & 2 Machine-Learning Applications*; European Union Aviation Safety Agency: Cologne, Germany, 2024. Available online: https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-concept-paper-issue-2 (accessed on 25 July 2026).
89. RTCA. *DO-326B: Airworthiness Security Process Specification*; RTCA: Washington, DC, USA, 2024.
90. EUROCAE. *ED-202B: Airworthiness Security Process Specification*; EUROCAE: Saint-Denis, France, 2024.