AEROC^2: TOWARDS EFFICIENT CLOSED-LOOP AERIAL VISION-LANGUAGE NAVIGATION VIA VISUAL COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Aerial vision-language-action (VLA) models incur substantial computational costs from visual feature extraction and downstream processing of dense visual token sequences, limiting navigation decision rates on resource-constrained unmanned aerial vehicles (UAVs). Existing approaches reduce these costs through encoder compression or token pruning, but optimizing either stage alone leaves the other source of computation largely unchanged. Moreover, token selection based solely on current-frame information can overlook cues needed for subsequent navigation, as flight actions change the viewpoint and available observations. We introduce Aerial Closed-Loop Navigation via Visual Compression (AeroC^2), a framework that addresses both computational costs while using navigation context to guide visual information retention. Complementary-teacher distillation transfers spatial and semantic features from two teacher encoders into a single lightweight student, which replaces both teachers during inference. A state-adaptive token allocator then uses navigation context and recent control history to select visual tokens, adjusts the token budget according to observation uncertainty, and applies role-aware retention constraints. The VLA backbone processes only the selected tokens, reducing downstream multimodal computation. We evaluate AeroC^2 on the Seen, Unseen Object, and Unseen Map splits of TravelUAV. Compared with the dense VLA baseline, AeroC^2 reduces FLOPs by 40.54% and mean end-to-end latency by 27.72% while also improving navigation performance. On the Seen split, it achieves a success rate (SR) of 51.48% and success weighted by path length (SPL) of 41.02%, representing improvements of 3.95% and 3.18%.

1 INTRODUCTION

Aerial vision-language navigation (VLN) requires unmanned aerial vehicles (UAVs) to reach language-specified goals using sequential multi-view observations (Wang et al., 2024; Xu et al., 2026; Sun et al., 2026). Recent vision-language-action (VLA) models map visual observations and language instructions directly to flight actions, integrating perception, reasoning, and action prediction within a single model (Xu et al., 2026; Sun et al., 2026). However, their visual processing incurs two major computational costs. First, visual encoders extract features from incoming observations at each decision step. In architectures that use multiple large encoders to capture complementary spatial and semantic information (Karamcheti et al., 2024), this feature extraction can be expensive. Second, the resulting dense visual token sequences increase the computation required by the downstream multimodal model (Li et al., 2025). Together, these costs limit the rate at which navigation decisions can be updated and make deployment on resource-constrained UAVs difficult (Xu et al., 2026).

Reducing these costs requires considering how visual compression affects navigation beyond the current frame. The visual evidence retained by a UAV influences its actions, which in turn change its viewpoint and subsequent observations. A region that appears unimportant in the current image may therefore matter for later decisions during the flight. For example, a subtle target cue can prevent an incorrect turn, while an obstacle boundary can guide safe motion and a region in the down view can become critical during landing. After the UAV moves, a discarded cue may no longer be visible. Visual compression for navigation should therefore reduce computation while retaining cues that support decisions as the flight progresses.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

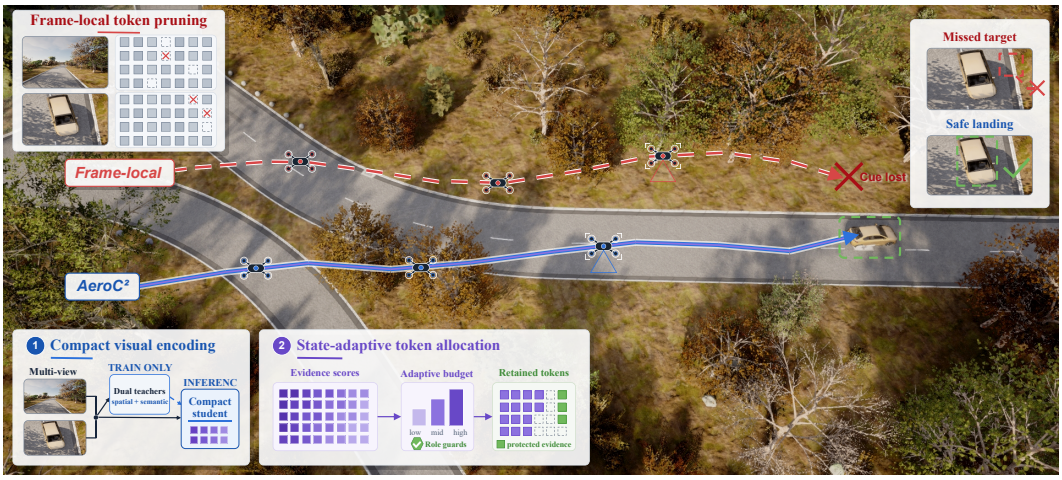

Figure 1: **Visual compression for future control.** Frame-local token pruning can discard behaviorally critical cues, whereas AeroC^2 combines compact visual encoding with state-adaptive token allocation to retain decision-relevant evidence.

Existing efficiency methods reduce visual computation by shrinking visual encoders or compressing the token sequences processed by multimodal models (Wu et al., 2023; Vasu et al., 2024; Zhang et al., 2025; Yang et al., 2025b). These two types of methods respectively target feature extraction and subsequent downstream processing. A smaller encoder can reduce feature-extraction cost but may still produce a dense token sequence, leaving substantial computational overhead for downstream processing. In contrast, pruning or compressing the token sequence after feature extraction can reduce downstream computation but cannot lower the feature-extraction cost. However, reducing computation addresses only one part of the problem. How to select and retain visual information that supports closed-loop decisions is equally important. Frame-local token compression, for example, selects tokens independently within each frame (Fang et al., 2026), based on the semantic relevance of the current frame. In closed-loop navigation, however, flight actions continuously change future viewpoints and the visual evidence required for subsequent decisions. Such frame-local selection may overlook the role of visual cues in subsequent control and consequently discard cues that are weakly semantically relevant in the current frame but critical for subsequent target recognition, obstacle avoidance, or landing. As shown in Figure 1, the loss of such critical cues ultimately leads to a missed target. Therefore, an open question remains: how can visual computation be reduced while retaining the key visual information required for closed-loop navigation?

To this end, we introduce Aerial Closed-Loop Navigation via Visual Compression (AeroC^2), a framework designed around the sensing and control demands of UAV flight. The framework has two coordinated stages. First, a lightweight visual encoder distills complementary spatial and semantic features from two teachers using paired front-view and down-view observations collected during flight, replacing both teacher encoders at inference. Second, a state-adaptive token allocator selects projected student features using instruction relevance, spatial cues, and recent flight-control history. Because each flight action changes the next viewpoint and the available visual evidence, the allocator adjusts the token budget according to observation uncertainty. Within this allocator, a role-aware retention mechanism exploits the UAV's dual-view sensing geometry by reserving front-view tokens for navigation and obstacle avoidance and down-view tokens for landing. Together, the two stages reduce feature-extraction and downstream multimodal computation while preserving visual evidence needed for subsequent flight control.

Our main contributions are as follows:

- We develop a complementary-teacher distillation module that consolidates spatial and semantic knowledge into a lightweight visual encoder, reducing feature-extraction cost while preserving navigation-relevant visual information.
- We design a state-adaptive token allocator that retains decision-relevant visual evidence under a variable token budget, reducing downstream multimodal computation while adapting

token retention to changing navigation states and the complementary roles of front-view and down-view observations.

- We establish an end-to-end efficiency evaluation protocol that jointly assesses navigation performance and efficiency. Extensive experiments show that AeroC^2 achieves strong navigation performance across seen and unseen environments with lower computational cost and latency, while ablations validate its core components.

2 Related Work

Vision-language-action models for aerial navigation. Recent vision-language-action (VLA) models unify visual perception, language understanding, and action generation within a single policy framework. PaLM-E and RT-2 explore transferring large-scale multimodal representations to robot control, demonstrating the potential of vision-language models for embodied decision making (Driess et al., 2023; Zitkovich et al., 2023). Octo and OpenVLA advance generalist policies through large-scale imitation learning and open training frameworks (Ghosh et al., 2024; Kim et al., 2025). In aerial navigation, TravelUAV establishes a realistic UAV navigation benchmark with diverse trajectories and simulation environments (Wang et al., 2024), while OpenUAV and OpenFly further expand aerial vision-language navigation datasets and evaluation platforms (Wang et al., 2024; Gao et al., 2026). Recent UAV VLA models, including AeroVLA, AutoFly, and VLA-AN, further investigate continuous action prediction, spatial reasoning, and safety-aware execution for autonomous flight (Xu et al., 2026; Sun et al., 2026; Wu et al., 2025b). However, existing works mainly focus on improving navigation capability, data scaling, and control reliability, while efficient VLA inference on resource-constrained aerial platforms remains insufficiently studied.

Efficient visual compression. Efficient visual compression encompasses compact visual representation learning and visual token compression (Ryoo et al., 2021). It reduces the visual computation of large vision-language models while retaining task-relevant information. Representation-focused methods such as TinyCLIP and MobileCLIP learn compact vision-language representations, while ToVE transfers complementary knowledge from multiple visual experts into efficient visual encoders (Wu et al., 2023; Vasu et al., 2024; Wu et al., 2025a). Token-level methods such as ToMe, FastV, and VisionZip reduce inference cost by merging or selecting visual tokens (Bolya et al., 2023; Chen et al., 2024; Yang et al., 2025a). For embodied VLA models, LightVLA and VLA-Pruner incorporate semantic or action-related signals to guide token selection (Jiang et al., 2025a; Liu et al., 2025). These approaches target different stages of visual processing, from encoder construction to sequence reduction, but most estimate visual importance from information available at the current decision step. Their objectives therefore emphasize immediate perception or action relevance rather than the future observability induced by each action. They do not explicitly model how an action changes the next viewpoint and the visual evidence available for subsequent decisions.

3 Method

3.1 Overview: Visual Compression for Closed-Loop Navigation

Aerial VLA models require visual encoding and multimodal reasoning at every control step, imposing a substantial computational burden on resource-constrained UAVs. Each action changes the next viewpoint and the visual information available for later decisions, making navigation a closed-loop process. Therefore, compression must reduce computation while preserving the spatial and semantic information required for closed-loop navigation. We consider language-guided aerial navigation from sequential front-view and down-view camera observations. At time t, the UAV receives the composed observation I_t and a language prompt Q_t containing the navigation instruction. As shown in Figure 2, AeroC^2 reduces visual encoding and multimodal processing costs through two coordinated stages. Stage I distills complementary knowledge from two visual teachers into a lightweight student. Stage II selects the student's visual tokens using multiple navigation cues and adjusts the token budget to observation uncertainty.

Let $\widehat{E}_t \in \mathbb{R}^{N\times d}$ denote the projected student features, containing N visual tokens of embedding dimension d. The allocator retains K_t of these N tokens to form the visual token sequence $\widetilde{E}_t \in$

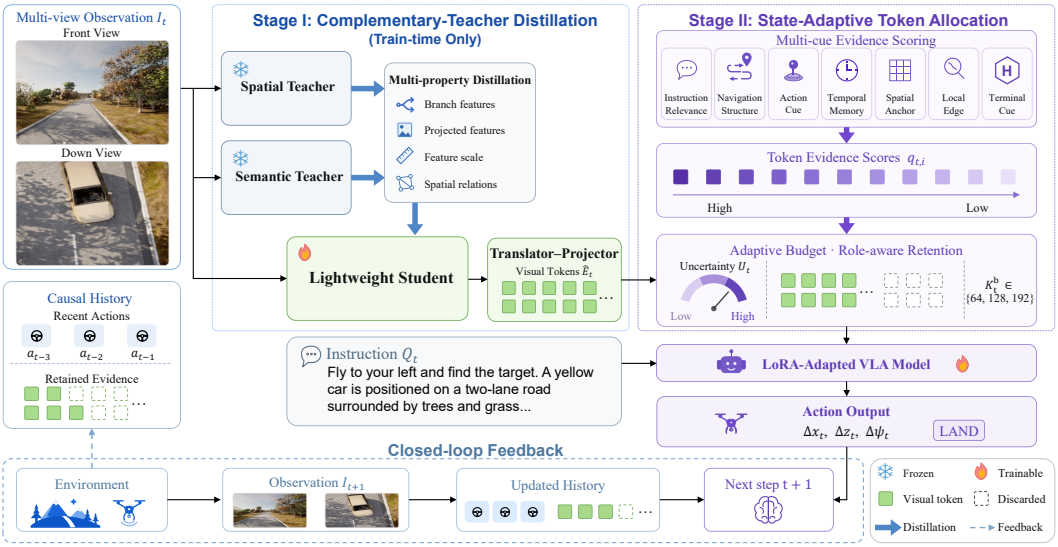

Figure 2: **Overview of AeroC^2.** Stage I distills complementary spatial and semantic knowledge into a lightweight student. Stage II combines multi-cue evidence scoring, adaptive budgeting, and role-aware retention. The VLA model processes the selected tokens to generate an action. The action changes the next viewpoint and observation, and the resulting information guides token allocation at the next control step, forming a closed-loop process.

$\mathbb{R}^{K_t \times d}$ for multimodal reasoning. AeroC^2 selects tokens relevant to the current navigation decision and adjusts K_t according to the uncertainty and navigation state of the current observation.

3.2 Complementary-Teacher Distillation

Lightweight student encoder and feature translators. Stage I distills complementary visual knowledge into a single lightweight encoder. We use DINOv2 ViT-L/14 (Oquab et al., 2024) and SigLIP ViT-So400M/14 (Zhai et al., 2023) as teachers, with DINOv3 ViT-B/16 (Siméoni et al., 2025) as the student. For observation I_t, we compute the teacher features:

$$F_t^b = T_b(I_t), \qquad F_t = \mathrm{Concat}_{b\in\mathcal{B}}(F_t^b), \qquad E_t = P(F_t), \tag{1}$$

where T_b denotes teacher encoder b, $\mathcal{B}$ is the set of teacher branches, and F_t^b is the corresponding teacher output features. $\mathrm{Concat}(\cdot)$ concatenates the complementary teacher features along the channel dimension, and the multimodal projector P maps the concatenated features to E_t, the projected teacher features used for distillation. The student encoder S_ϕ processes the same observation once and produces shared visual features Z_t. Lightweight branch-specific feature translators map Z_t into the DINOv2 and SigLIP feature spaces:

$$Z_t = S_\phi(I_t), \qquad \widehat{F}_t^b = \mathrm{MLP}_b(\mathrm{LN}(Z_t)), \quad b \in \mathcal{B}, \tag{2}$$

where LN denotes token-wise LayerNorm, and each MLP_b is a two-layer MLP with GELU activation. The translated feature $\widehat{F}_t^b$ denotes the student features mapped into the feature space of teacher branch b. The translated features are then concatenated and projected into the multimodal embedding space:

$$\widehat{F}_t = \mathrm{Concat}_{b\in\mathcal{B}}\left(\widehat{F}_t^b\right), \qquad \widehat{E}_t = P(\widehat{F}_t), \tag{3}$$

where $\widehat{F}_t$ denotes the fused student features and $\widehat{E}_t$ denotes the projected output. Thus, inference uses only a single lightweight student encoder, while the lightweight feature translators map the shared student features into the corresponding teacher feature spaces to preserve complementary visual information from both teachers.

Multi-property visual distillation. Direct feature matching alone may not sufficiently preserve the complementary information contained in the teacher features. We therefore supervise the student

from four aspects: teacher-branch features, projected features, feature scale, and spatial relations. The Stage I objective is:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{branch}} + \lambda_{\text{proj}}\mathcal{L}_{\text{proj}} + \lambda_{\text{scale}}\mathcal{L}_{\text{scale}} + \lambda_{\text{rel}}\mathcal{L}_{\text{rel}}, \tag{4}$$

where $\mathcal{L}_{\text{branch}}$ supervises each translated student feature using its corresponding teacher feature, and $\mathcal{L}_{\text{proj}}$ supervises the projected student feature using the projected teacher feature. $\mathcal{L}_{\text{scale}}$ constrains feature magnitude, and $\mathcal{L}_{\text{rel}}$ preserves spatial relations among visual tokens. The coefficients λ_{proj}, λ_{scale}, and λ_{rel} control the relative contributions of the corresponding terms.

To supervise the student with teacher-branch and projected features, we apply cosine distance to token features and Smooth L1 loss to token-wise layer-normalized features. The cosine term aligns feature directions, whereas Smooth L1 constrains differences between feature components. Because directional matching does not preserve feature magnitude, the feature-scale loss further constrains the norm of each token feature to match the teacher feature norm. For spatial-relation supervision, we first adaptively pool the native 16×16 visual token grid to 8×8 and then compare pairwise similarities between teacher and student tokens, helping the student preserve the teacher's spatial layout (Tung & Mori, 2019).

During Stage I, we train the student visual encoder and feature translators while keeping the teacher encoders and multimodal projector frozen. After distillation, we remove the teacher encoders and freeze the student visual encoder and feature translators for Stage II. Detailed definitions of the individual loss terms and their coefficients are provided in Appendix A.

3.3 State-Adaptive Token Allocation

The distilled student visual encoder and feature translators supply features to the multimodal projector, which produces $\widehat{E}_t = \{\widehat{e}_{t,i}\}_{i=1}^{N}$, where $\widehat{e}_{t,i}$ denotes the i-th visual token at time t. Allocation operates on these projected tokens to compare them with language features and control how many visual tokens enter the VLA model. The allocator has no trainable parameters. It consists of three components: multi-cue evidence scoring, adaptive budgeting, and role-aware retention.

Multi-cue evidence scoring. Aerial navigation requires different visual evidence across flight states. Instruction relevance alone may identify regions associated with the language goal but underemphasize navigation structure, boundaries, motion-related regions, or down-view evidence near landing. We therefore estimate token importance using seven complementary cues: instruction relevance s, navigation-structure evidence n, action-conditioned evidence a, temporal memory m, spatial-anchor evidence r, local-edge evidence g, and terminal evidence ℓ. Let the cue set be:

$$\mathcal{C} = \{s, n, a, m, r, g, \ell\}. \tag{5}$$

For token i at time t, the overall evidence score is:

$$q_{t,i} = \text{Norm}\left(\sum_{c \in \mathcal{C}\setminus\{\ell\}} w_c\, \bar{c}_{t,i} + w_\ell(t)\bar{\ell}_{t,i}\right), \tag{6}$$

where $\bar{c}_{t,i} \in [0,1]$ denotes the per-frame normalized response of cue c, w_c is its fixed coefficient, $w_\ell(t)$ is the state-dependent terminal coefficient, and $\text{Norm}(\cdot)$ denotes per-frame min–max normalization.

The seven cues capture complementary information for closed-loop navigation. Instruction relevance s measures semantic similarity between projected visual tokens and the navigation prompt. Navigation-structure evidence n combines token magnitude, local feature discontinuity, and a spatial prior over the composed front and down views. Action-conditioned evidence a adjusts spatial importance using recent forward, vertical, and yaw commands. Temporal memory m prioritizes regions that remain informative across sequential observations. Spatial-anchor evidence r favors distributed coverage across the visual token grid. Local-edge evidence g highlights feature boundaries as cues for navigation and obstacle avoidance. Terminal evidence ℓ emphasizes down-view cues and motion states associated with terminal approach and landing.

During sequential inference, the action-conditioned and temporal-memory cues use recent control states and previously retained visual evidence. During Stage-II training, where samples are shuffled frames rather than complete trajectories, we disable only the history-dependent components to prevent cross-trajectory state leakage, while retaining the static spatial priors. Detailed cue definitions, normalization rules, weighting coefficients, and other scoring constants are provided in Appendix A.

Adaptive budgeting. We adapt the token budget to the uncertainty of the current observation. We combine four signals into an observation uncertainty score:

$$U_t = \beta_H H_t + \beta_\Delta (1 - \Delta_t) + \beta_C C_t + \beta_M M_t, \tag{7}$$

where H_t denotes the normalized entropy of the instruction-relevance distribution, Δ_t measures the score separation among highly relevant visual tokens, C_t measures disagreement between the semantic responses of the front and down views, and M_t characterizes recent vertical or yaw motion. The coefficients β_H, β_Δ, β_C, and β_M control the relative contributions of the four signals. The uncertainty score determines the base token budget:

$$K_t^{\mathrm{b}} = \begin{cases} K_{\mathrm{lo}}, & U_t < \tau_{\mathrm{lo}}, \\ K_{\mathrm{mid}}, & \tau_{\mathrm{lo}} \le U_t < \tau_{\mathrm{hi}}, \\ K_{\mathrm{hi}}, & U_t \ge \tau_{\mathrm{hi}}, \end{cases} \tag{8}$$

where K_{lo}, K_{mid}, and K_{hi} denote the low, medium, and high token budgets, respectively, while τ_{lo} and τ_{hi} are the corresponding uncertainty thresholds. Accordingly, low-uncertainty observations use a smaller token budget, whereas high-uncertainty observations retain more visual evidence. Terminal states additionally impose a minimum retention budget:

$$K_t = \max\left(K_t^{\mathrm{b}}, K_t^{\mathrm{term}}\right), \tag{9}$$

where K_t^{term} denotes the retention floor activated by the current terminal state and is zero when no terminal constraint is active. Operating budgets, uncertainty coefficients, thresholds, and terminal-state criteria are reported in Appendix A.

Role-aware retention. Given the adaptive budget K_t and evidence scores $q_{t,i}$, we construct the retained set Ω_t through a deterministic staged selection procedure. The projected visual grid is partitioned into front-view and down-view regions so that evidence from both sensing roles is maintained. The allocator first reserves spatial-anchor tokens distributed over the visual grid. During sequential inference, when uncertainty is medium or high and nonzero temporal memory is available, high-memory tokens are then selected from the remaining candidates according to their temporal-memory responses, subject to a predefined memory reserve. Next, we require at least ρ_f tokens from the front view and ρ_d tokens from the down view. If either view falls below its minimum, the allocator adds the highest-scoring unselected tokens from that view until the required count is reached.

After spatial anchoring, temporal recovery, and view-coverage constraints are satisfied, the unfilled portion of K_t is used for three state-dependent safeguards. Terminal protection increases down-view retention when the UAV approaches landing and activates the corresponding terminal budget constraint. Edge rescue selects unretained tokens with strong local-edge responses, while motion rescue selects unretained tokens associated with the current motion state. Each safeguard only adds previously unselected tokens and is bounded by the available token capacity. After the three safeguards are applied, if fewer than K_t tokens have been selected, the remaining unselected tokens are added in descending order of $q_{t,i}$ until the budget is reached. Let $\Omega_t \subseteq \{1, \ldots, N\}$ denote the resulting retained token indices. The compressed visual sequence passed to the multimodal policy is:

$$\widetilde{E}_t = \widehat{E}_t[\Omega_t], \qquad |\Omega_t| = K_t, \tag{10}$$

where $\widehat{E}_t[\Omega_t]$ denotes indexing the projected visual features by the retained set Ω_t. The allocator therefore combines multi-cue evidence scoring with deterministic staged selection without introducing an additional trainable token-selection network. All view quotas, reserve sizes, activation thresholds, recovery ratios, and other allocator constants are provided in Appendix A.

3.4 VLA WITH DYNAMIC VISUAL TOKEN INPUTS

The retained visual tokens $\widetilde{E}_t$ are combined with the language prompt Q_t and passed to the VLA model. During Stage-II training, different samples may retain different numbers of visual tokens according to their sample-dependent budgets K_t. To support batch training, the retained visual tokens are padded to the largest retained token count within each batch, with an attention mask preventing padded positions from contributing to attention. During closed-loop inference, the model directly processes the visual tokens retained at the current control step. During Stage II, the student encoder



Figure 3: **Qualitative closed-loop rollouts of AeroC^2.** Each row shows eight uniformly sampled timesteps from one complete rollout. The right panel shows unmasked front-view and down-view observations stacked vertically, together with the full instruction. The left plots show the full-route token budget schedule K_t. Orange and blue indicate search and target-approach phases, green dots mark sampled steps, and red boxes identify visible targets.

and feature translators obtained from Stage I remain frozen, whereas the multimodal projector and LoRA adapters (Hu et al., 2022) in the VLA backbone are trained. This allows the VLA model to adapt to dynamically compressed visual inputs while preserving the visual features learned during Stage I. Detailed training settings are provided in Appendix B.

At time t, the VLA model generates tokens encoding a three-degree-of-freedom flight command. We represent the continuous flight command as a three-component action vector:

$$A_t = (\Delta x_t, \Delta z_t, \Delta \psi_t)\,, \tag{11}$$

where Δx_t, Δz_t, and $\Delta \psi_t$ denote forward displacement, vertical displacement, and yaw change, respectively. We discretize and serialize these values as numerical action tokens, together with an optional `LAND` command and an end-of-sequence token. Let $\mathcal{J}_t$ denote the set of supervised positions in the serialized action sequence, and let $y^*_{t,j}$ denote the ground-truth action token at position j. We train the model to predict the action tokens using the following loss:

$$\mathcal{L}_{\text{act}} = -\mathbb{E}_t \left[\frac{1}{|\mathcal{J}_t|} \sum_{j \in \mathcal{J}_t} \log p_\theta \left(y^*_{t,j} \mid \widetilde{E}_t, Q_t, y^*_{t,<j} \right) \right], \tag{12}$$

where p_θ denotes the conditional probability of the action token at the current output position, Q_t denotes the navigation prompt, and $y^*_{t,<j}$ denotes the preceding ground-truth action tokens used for autoregressive supervision. $\mathcal{J}_t$ contains the supervised positions corresponding to the action-output sequence, so the training objective is applied to the complete serialized action output.

4 Experiments

4.1 Experimental Setup

Dataset and evaluation protocol. We evaluate AeroC^2 on the TravelUAV UAV-Need-Help benchmark (Wang et al., 2024; Xu et al., 2026), following the public data split and evaluation protocol of AeroVLA (Xu et al., 2026). The training set contains 7,922 expert trajectories. Evaluation is conducted under three splits: Seen (1,418 routes), Unseen Object (629 routes), and Unseen Map (958 routes). Navigation metrics are reported on the full set of routes in all three splits. Efficiency is evaluated on a stratified subset of 2,400 decision steps, with 800 steps sampled from each split. We report Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OSR), and Success weighted by Path Length (SPL) (Anderson et al., 2018a), together with total parameters, profiled FLOPs per

Table 1: **Navigation performance and efficiency on TravelUAV.** Navigation metrics cover the Seen, Unseen Object, and Unseen Map subsets. Efficiency is measured on 2,400 decision steps selected by stratified sampling. FLOPs are reported per decision. Bold and underlined values indicate the best and second-best results per column. TravelUAV denotes the data-aggregation (DA) variant on Seen.

	Seen				Unseen Object				Unseen Map				Efficiency		
Method	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑	Params (B)↓	FLOPs (T)↓	Latency (ms)↓
Random Action	222.20	0.14	0.21	0.07	260.14	0.16	0.16	0.16	202.98	0.00	0.00	0.00	–	–	–
Fixed Action	188.61	2.27	8.16	1.40	212.84	3.66	9.54	2.16	180.47	0.52	2.61	0.39	–	–	–
CMA (Anderson et al., 2018b)	135.73	8.37	18.72	7.90	155.79	9.06	16.06	8.68	141.68	2.30	10.02	2.16	–	–	–
TravelUAV (Wang et al., 2024)	98.66	17.45	48.87	15.76	118.11	22.42	46.90	20.51	138.80	4.18	20.77	3.84	9.28	8.47	449.18
NavFoM (Zhang et al., 2026)	93.05	29.17	49.24	25.03	108.04	29.83	47.99	27.20	125.10	6.30	18.95	5.68	–	–	–
LongFly (Jiang et al., 2025b)	**60.02**	36.39	**65.87**	31.07	66.74	43.87	64.56	38.39	108.32	11.27	30.27	9.32	–	–	–
AeroVLA (Xu et al., 2026)	65.88	47.96	57.69	38.54	61.45	56.60	64.86	46.61	67.42	37.58	52.92	28.22	7.77	5.18	472.94
AeroC^2 (ours)	64.93	**51.48**	59.94	**41.02**	**53.77**	**59.78**	**68.36**	**49.61**	**59.79**	**42.38**	**57.62**	**31.82**	**7.13**	**3.08**	**341.86**

Table 2: **Contributions of visual distillation and token allocation.** Navigation metrics cover Seen, Unseen Object, and Unseen Map subsets. Checkmarks indicate active components, and bold values mark the best results in each column. All configurations use the same training protocol as AeroC^2.

	Components		Seen		Unseen Object		Unseen Map		Efficiency		
Configuration	Distill.	Alloc.	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑	Params (B)↓	FLOPs (T)↓	Latency (ms)↓
Dense VLA	✗	✗	47.53	37.84	56.12	45.97	37.16	27.64	7.77	5.18	472.94
Visual distillation only	✓	✗	48.73	38.42	56.76	46.58	**42.69**	31.75	**7.13**	4.86	410.93
Token allocation only	✗	✓	48.45	38.05	57.87	47.31	37.47	27.92	7.77	3.48	354.37
AeroC^2 (ours)	✓	✓	**51.48**	**41.02**	**59.78**	**49.61**	42.38	**31.82**	**7.13**	**3.08**	**341.86**

decision, and mean end-to-end latency. All navigation metrics are computed from closed-loop rollouts in the TravelUAV AirSim simulator (Shah et al., 2017). Efficiency measurements use batch size 1 and BF16 on an NVIDIA RTX 4090, timing from image preprocessing to action decoding after warm-up.

Model and training details. We use OpenVLA-7B (Kim et al., 2025) as the VLA backbone and use the same language prompts, observation format, action representation, and closed-loop navigation protocol as AeroVLA (Xu et al., 2026). Both stages use AdamW with a batch size of 64 (Loshchilov & Hutter, 2019). Stage I runs for five epochs, using learning rates of 2×10^{-5} for the student encoder and 2×10^{-4} for the feature translators. Stage II trains the multimodal projector and backbone LoRA adapters for 46,000 steps. The LoRA adapters use rank 64 and a learning rate of 2×10^{-4}.

4.2 Main Results

Quantitative comparison. We compare navigation performance on the Seen, Unseen Object, and Unseen Map subsets of TravelUAV and report efficiency for the measured VLA configurations in Table 1. Compared with AeroVLA, AeroC^2 improves SR by 3.52%, 3.18%, and 4.80% on Seen, Unseen Object, and Unseen Map, respectively. It achieves the lowest parameter count, FLOPs, and latency among the measured VLA configurations. FLOPs decrease by 40.54% and mean end-to-end latency decreases by 27.72% relative to AeroVLA, combining higher navigation success with lower inference cost.

Qualitative visualization. Figure 3 visualizes how AeroC^2 adapts its token budget during two closed-loop rollouts. In the first rollout, the budget alternates between 64 and 128 tokens during search and increases to 192 near the target. In the second rollout, the budget rises to 192 early in search, drops briefly to 64, and remains at 128 through target approach, with terminal protection inactive during this phase. These schedules illustrate how allocation changes along a route rather than using a fixed budget throughout the flight.

4.3 Ablation Studies

Component contributions. We evaluate the two compression stages in Table 2. The Visual distillation only configuration uses the distilled student pathway without token allocation. For the Token allocation only configuration, visual features are extracted by both teacher encoders. Each component improves navigation over the dense configuration, and their combination achieves the highest mean SR and SPL across the three splits. Visual distillation reduces the cost of extracting visual features, while token allocation reduces the number of visual tokens processed by the VLA

Table 3: **Visual distillation ablations.** We report SR and SPL on Seen, Unseen Object, and Unseen Map using the same token allocator. The No distillation configuration uses a single DINOv3 student without teacher supervision.

Configuration	Seen		Unseen Object		Unseen Map	
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
No distillation	37.87	29.64	34.02	28.05	20.98	16.35
Spatial teacher (DINOv2)	41.82	33.10	49.28	40.35	38.94	27.65
Semantic teacher (SigLIP)	33.07	25.60	39.27	31.73	20.25	12.49
Dual teachers (ours)	**51.48**	**41.02**	**59.78**	**49.61**	**42.38**	**31.82**

Table 4: **Token allocation ablations.** Navigation metrics are reported on all routes in each split. K denotes the retained visual token budget, and the adaptive setting reports its mean budget.

Configuration	K	Seen		Unseen Object		Unseen Map	
		SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
Random selection	126	36.25	28.44	46.26	37.85	37.16	26.82
Fixed budget	64	46.40	36.58	52.94	42.46	36.01	26.20
Fixed budget	126	47.60	37.64	54.37	44.15	38.10	27.35
Fixed budget	192	47.95	38.02	55.64	45.42	39.14	28.12
w/o role-aware retention	126	47.88	37.92	55.48	45.31	36.22	26.45
AeroC^2 (ours)	126	**51.48**	**41.02**	**59.78**	**49.61**	**42.38**	**31.82**

backbone at each control step. The complete model reduces FLOPs by 40.54% and mean end-to-end latency by 27.72%. The two components act at different stages, so they reduce computation and latency while maintaining strong navigation performance.

Visual distillation ablation. We evaluate the contribution of the visual teachers while keeping the token allocator fixed. The No distillation configuration uses a single DINOv3 student without teacher supervision. Table 3 compares the student without teacher supervision, single-teacher distillation using the spatial DINOv2 or semantic SigLIP teacher, and the complete dual-teacher objective. The dual-teacher model outperforms both single-teacher variants across all three splits, supporting the use of complementary spatial and semantic supervision for closed-loop navigation.

Token allocation ablations. We compare random selection with selection based on our multi-cue evidence scoring and examine fixed budgets of 64, 126, and 192 tokens in Table 4. The model retains $K = 126$ tokens on average, so we use a fixed budget of $K = 126$ for this comparison. At this fixed budget, selection based on our multi-cue evidence scoring substantially outperforms random selection, increasing SR by 11.35 percentage points on Seen and 8.11 percentage points on Unseen Object. Increasing the budget from 64 to 192 tokens further improves SR and SPL across all three splits by retaining additional complementary evidence. Compared with a fixed budget of 126 tokens, our adaptive budgeting also achieves higher SR and SPL on all three splits. We also compare the full allocator with the allocator without role-aware retention at a reported mean budget of 126 tokens. Compared with the allocator without role-aware retention, the full allocator achieves higher SR and SPL across all three splits, with the largest gains on Unseen Map, confirming the benefit of role-aware retention. Overall, our multi-cue evidence scoring, adaptive budgeting, and role-aware retention help retain visual evidence useful for navigation.

5 Conclusion

We introduce AeroC^2, a visual compression framework for closed-loop aerial vision-language navigation. Complementary-teacher distillation and state-adaptive token allocation reduce computation while retaining evidence relevant to subsequent control. Our framework achieves strong navigation performance across seen and unseen environments. It reduces FLOPs by 40.54% and mean end-to-end latency by 27.72% relative to the dense VLA baseline. These results show AeroC^2 enables efficient aerial VLA inference by reducing visual computation while improving navigation performance.

AI USE STATEMENT

In this work, we used generative AI tools to assist with literature searches and the identification of related work. The authors individually reviewed and verified all references identified through this process. The authors take full responsibility for the references and final content of the paper.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018a. URL https://arxiv.org/abs/1807.06757.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018b.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024. doi: 10.1007/978-3-031-73004-7_2. URL https://arxiv.org/abs/2403.06764.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. URL https://arxiv.org/abs/1604.06174.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/0b408293619f725fd30162af057e531a-Paper-Conference.pdf.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in VLMs via synergistic importance-diversity. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=i36E5Ezm0H.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, Yiwen Tang, Yuhang Tang, Shuai Liang, Songyi Zhu, Ziqin Xiong, Yifei Su, Xinyi Ye, Jianan Li, Yan Ding, Dong Wang, Xuelong Li, Zhigang Wang, and Bin Zhao. OpenFly: A comprehensive platform for aerial vision-language navigation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2502.18041.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. URL https://www.roboticsproceedings.org/rss20/p090.html.

Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1921–1930, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Heo_A_Comprehensive_Overhaul_of_Feature_Distillation_ICCV_2019_paper.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Titong Jiang, Xuefeng Jiang, Yuan Ma, Xin Wen, Bailin Li, Kun Zhan, Peng Jia, Yahui Liu, Sheng Sun, and Xianpeng Lang. The better you learn, the smarter you prune: Towards efficient vision-language-action models via differentiable token pruning. *arXiv preprint arXiv:2509.12594*, 2025a. URL https://arxiv.org/abs/2509.12594.

Wen Jiang, Li Wang, Kangyao Huang, Wei Fan, Jinyuan Liu, Shaoyu Liu, Hongwei Duan, Bin Xu, and Xiangyang Ji. LongFly: Long-horizon UAV vision-and-language navigation with spatiotemporal context integration. *arXiv preprint arXiv:2512.22010*, 2025b. URL https://arxiv.org/abs/2512.22010.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23123–23144. PMLR, 2024. URL https://proceedings.mlr.press/v235/karamcheti24a.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Kevin Y. Li, Sachin Goyal, João D. Semedo, and J. Zico Kolter. Inference optimal VLMs need fewer visual tokens and more parameters. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ef748b4544e8f12d608fd28bbf04177f-Paper-Conference.pdf.

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Bridging the semantic-action gap in visual token pruning for efficient VLA inference. *arXiv preprint arXiv:2511.16449*, 2025. URL https://arxiv.org/abs/2511.16449.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2304.07193.

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: What can 8 learned tokens do for images and videos? In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.11297.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pp. 621–635. Springer, 2017. URL https://arxiv.org/abs/1705.05065.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Xiaolou Sun, Wufei Si, Wenhui Ni, Yuntian Li, Dongming Wu, Fei Xie, Runwei Guan, Heyang Xu, Henghui Ding, Yuan Wu, Yutao Yue, Yongming Huang, and Hui Xiong. AutoFly: Vision-language-action model for UAV autonomous navigation in the wild. In *International Conference on Learning Representations*, 2026.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1365–1374, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Tung_Similarity-Preserving_Knowledge_Distillation_ICCV_2019_paper.html.

Pavan Kumar Anasosalu Vasu, Hadi Pour Ansari, Fartash Faghri, Raviteja Vemulapalli, and Oncel Tuzel. MobileCLIP: Fast image-text models through multi-modal reinforced training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15963–15974, 2024. doi: 10.1109/CVPR52733.2024.01511. URL https://openaccess.thecvf.com/content/CVPR2024/html/Vasu_MobileCLIP_Fast_Image-Text_Models_through_Multi-Modal_Reinforced_Training_CVPR_2024_paper.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Wenjun Wu, Hongsheng Li, Yue Liao, and Si Liu. Towards realistic UAV vision-language navigation: Platform, benchmark, and methodology. *arXiv preprint arXiv:2410.07087*, 2024. URL https://arxiv.org/abs/2410.07087.

Kan Wu, Houwen Peng, Zhenghong Zhou, Bin Xiao, Mengchen Liu, Lu Yuan, Hong Xuan, Michael Valenzuela, Xi (Stephen) Chen, Xinggang Wang, Hongyang Chao, and Han Hu. TinyCLIP: CLIP distillation via affinity mimicking and weight inheritance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21913–21923, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wu_TinyCLIP_CLIP_Distillation_via_Affinity_Mimicking_and_Weight_Inheritance_ICCV_2023_paper.html.

Yuanchen Wu, Junlong Du, Ke Yan, Shouhong Ding, and Xiaoqiang Li. ToVE: Efficient vision-language learning via knowledge transfer from vision experts. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2504.00691.

Yuze Wu, Mo Zhu, Xingxing Li, Yuheng Du, Yuxin Fan, Wenjun Li, Zhichao Han, Xin Zhou, and Fei Gao. VLA-AN: An efficient and onboard vision-language-action framework for aerial navigation in complex environments. *arXiv preprint arXiv:2512.15258*, 2025b. URL https://arxiv.org/abs/2512.15258.

Peng Xu, Zhengnan Deng, Jiayan Deng, Zonghua Gu, and Shaohua Wan. AerialVLA: A vision-language-action model for UAV navigation via minimalist end-to-end control. *arXiv preprint arXiv:2603.14363*, 2026. URL https://arxiv.org/abs/2603.14363.

Chuanguang Yang, Zhulin An, Libo Huang, Junyu Bi, Xinqiang Yu, Han Yang, Boyu Diao, and Yongjun Xu. CLIP-KD: An empirical study of CLIP model distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15952–15962, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_CLIP-KD_An_Empirical_Study_of_CLIP_Model_Distillation_CVPR_2024_paper.html.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_VisionZip_Longer_is_Better_but_Not_Necessary_in_Vision_Language_CVPR_2025_paper.html.

Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. EfficientVLA: Training-free acceleration and compression for vision-language-action models. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-1365. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3a2ef31a1e45908901adc0ca853a8faf-Paper-Conference.pdf.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhai_Sigmoid_Loss_for_Language_Image_Pre-Training_ICCV_2023_paper.html.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and He Wang. Embodied navigation foundation model. In *International Conference on Learning Representations*, 2026.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25s.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Additional Method Details

Throughout this appendix, t indexes control steps and $i \in \{1, \dots, N\}$ indexes visual tokens, with $N = 256$ on the native 16×16 grid. The teacher-branch set is $\mathcal{B} = \{d, s\}$, where d and s label DINOv2 and SigLIP, respectively. The student encoder S_ϕ has parameters ϕ. Feature notation follows Equations 1–3. We use $\cos(\cdot, \cdot)$ for cosine similarity, $\|\cdot\|_2$ for the Euclidean norm, $\|\cdot\|_F$ for the Frobenius norm, and $\log$ for the natural logarithm.

A.1 Complementary-Teacher Distillation Losses

Let $z_{t,i} \in \mathbb{R}^{d_z}$ be token i of the student features Z_t in Equation 2, where d_z is the student feature dimension and d_b is the output dimension of teacher branch b. The feature translator is:

$$G_b(z_{t,i}) = W_{b,2}\, \sigma(W_{b,1}\, \mathrm{LN}_b(z_{t,i}) + b_{b,1}) + b_{b,2}, \tag{13}$$

where $W_{b,1} \in \mathbb{R}^{r_b \times d_z}$ and $W_{b,2} \in \mathbb{R}^{d_b \times r_b}$ are weights, $b_{b,1} \in \mathbb{R}^{r_b}$ and $b_{b,2} \in \mathbb{R}^{d_b}$ are biases, σ is GELU, and the hidden width is $r_b = d_b$. The branch-specific LN_b has learned affine parameters.

Let $X, Y \in \mathbb{R}^{N \times d'}$ contain predicted and target token features of dimension d', with rows x_i and y_i. In the feature-matching loss, $j \in \{1, \dots, d'\}$ indexes feature components, and α_{dir} and α_{shape} weight cosine distance and Smooth L1 loss, respectively. The operator LN normalizes each token across its feature components without learned affine parameters:

$$\begin{aligned}\ell_{\mathrm{match}}(X, Y) = \alpha_{\mathrm{dir}} \frac{1}{N} \sum_{i=1}^{N} (1 - \cos(x_i, y_i)) \\ + \alpha_{\mathrm{shape}} \frac{1}{N d'} \sum_{i=1}^{N} \sum_{j=1}^{d'} \rho_{\delta_H}(\mathrm{LN}(x_i)_j - \mathrm{LN}(y_i)_j)\,.\end{aligned} \tag{14}$$

For a scalar residual z and transition threshold $\delta_H = 1$, Smooth L1 equals the Huber penalty:

$$\rho_{\delta_H}(z) = \begin{cases} \frac{1}{2} z^2, & |z| \le \delta_H, \\ \delta_H(|z| - \frac{1}{2}\delta_H), & |z| > \delta_H. \end{cases} \tag{15}$$

For teacher branches $b \in \mathcal{B}$ with weights π_b, branch and projected-feature supervision are:

$$\mathcal{L}_{\mathrm{branch}} = \sum_{b \in \mathcal{B}} \pi_b \ell_{\mathrm{match}}(\widehat{F}_t^b, F_t^b), \qquad \mathcal{L}_{\mathrm{proj}} = \ell_{\mathrm{match}}(\widehat{E}_t, E_t). \tag{16}$$

Hatted features are student outputs, and unhatted features are teacher targets. Lowercase f and e denote token rows of the branch and projected features, and ϵ provides a positive norm floor. Superscripts B and E identify the branch and projected-feature scale losses. Feature-scale supervision matches the unnormalized token norms:

$$\mathcal{L}_{\mathrm{scale}} = \tfrac{1}{2}\mathcal{L}^B_{\mathrm{scale}} + \tfrac{1}{2}\mathcal{L}^E_{\mathrm{scale}}, \tag{17}$$

$$\mathcal{L}^B_{\mathrm{scale}} = \sum_{b \in \mathcal{B}} \frac{\pi_b}{N} \sum_i \rho_{\delta_H}\left(\log \frac{\max(\epsilon, \|\widehat{f}^b_{t,i}\|_2)}{\max(\epsilon, \|f^b_{t,i}\|_2)}\right), \tag{18}$$

$$\mathcal{L}^E_{\mathrm{scale}} = \frac{1}{N} \sum_i \rho_{\delta_H}\left(\log \frac{\max(\epsilon, \|\widehat{e}_{t,i}\|_2)}{\max(\epsilon, \|e_{t,i}\|_2)}\right). \tag{19}$$

For spatial-relation supervision, $A_{m_{\mathrm{rel}}}(X)$ applies adaptive average pooling to the native 16×16 grid and L_2-normalizes each pooled token. The output grid has size $m_{\mathrm{rel}} \times m_{\mathrm{rel}}$, with $m_{\mathrm{rel}} = 8$, and is flattened into m^2_{rel} token rows. Let $\Gamma_{m_{\mathrm{rel}}}(X) = A_{m_{\mathrm{rel}}}(X) A_{m_{\mathrm{rel}}}(X)^\top$ be the $m^2_{\mathrm{rel}} \times m^2_{\mathrm{rel}}$ matrix of pairwise token similarities. The spatial-relation loss compares these matrices for the concatenated branch features:

$$\mathcal{L}_{\mathrm{rel}} = \frac{1}{m^4_{\mathrm{rel}}} \left\| \Gamma_{m_{\mathrm{rel}}}(\widehat{F}_t) - \Gamma_{m_{\mathrm{rel}}}(F_t) \right\|^2_F. \tag{20}$$

Table 5: Coefficients and architecture settings for complementary-teacher distillation.

Quantity	Value	Role
π_d, π_s	$0.5, 0.5$	Equal DINOv2 and SigLIP branch weights
$\alpha_{\text{dir}}, \alpha_{\text{shape}}$	$0.9, 0.1$	Cosine and Smooth L1 weights in both feature-matching terms
δ_H, ϵ	$1, 10^{-6}$	Penalty transition and token-norm floor
Branch coefficient	1	Unit coefficient of $\mathcal{L}_{\text{branch}}$
λ_{proj}	1	Projected-feature supervision
λ_{scale}	0.2	Feature-scale supervision
λ_{rel}	0.1	Spatial-relation supervision
m_{rel}	8	Relation-grid side after pooling from 16×16
r_b	d_b	Translator hidden width

A.2 Distillation Coefficients

Table 5 lists the coefficients of the distillation objective in Equation 4, the feature-matching weights, and the translator settings. The objectives combine feature-level supervision (Heo et al., 2019; Yang et al., 2024) with similarity-preserving supervision (Tung & Mori, 2019).

A.3 Cue Normalization and Temporal Memory

The per-frame min–max normalization operator Norm used in Equation 6 acts on a vector $x = (x_1, \ldots, x_N)$ of scalar scores over the visual token grid. Its output at token i is:

$$\mathrm{Norm}_i(x) = \begin{cases} 0, & \max_j x_j - \min_j x_j < 10^{-8}, \\ \dfrac{x_i - \min_j x_j}{\max_j x_j - \min_j x_j}, & \text{otherwise}, \end{cases} \tag{21}$$

where N is the number of visual tokens in the current frame, x_i is the input score of token i, and the minimum and maximum are taken over all token indices $j \in \{1, \ldots, N\}$. The 10^{-8} threshold maps nearly constant score vectors to zero. For the token-indexed expressions below, normalization is likewise performed over the full token grid.

Let $\bar{q}_t$ be the mean of the language model's input embeddings for non-padding prompt tokens, excluding the beginning-of-sequence token. Instruction relevance is:

$$s_{t,i} = \mathrm{Norm}_i(\cos(\widehat{e}_{t,i}, \bar{q}_t)) \,. \tag{22}$$

Let $u_{t,i} = \mathrm{Norm}_i(\|\widehat{e}_{t,i}\|_2)$ be the normalized projected-token norm. Navigation-structure evidence combines this norm with local-edge evidence $g_{t,i}$ and the spatial prior p_i. Their respective weights α_u, α_g, and α_p are listed in Table 6:

$$n_{t,i} = \mathrm{Norm}_i(\alpha_u u_{t,i} + \alpha_g g_{t,i} + \alpha_p p_i) \,, \quad \alpha_u + \alpha_g + \alpha_p = 1. \tag{23}$$

Let $q_{t,i}$ be the combined evidence score and Ω_t the retained index set. The update target $\zeta_{t,i}$ mixes the score and selection indicator with weights ν and $1 - \nu$. Here, $\mathbf{1}[i \in \Omega_t]$ is one for a retained token and zero otherwise. The stored temporal memory $m_{t,i}$ then uses an exponential update with decay μ:

$$\zeta_{t,i} = \mathrm{Norm}_i(\nu q_{t,i} + (1 - \nu)\mathbf{1}[i \in \Omega_t]) \,, \tag{24}$$

$$m_{t+1,i} = \begin{cases} \zeta_{t,i}, & \text{after the first decision}, \\ \mu m_{t,i} + (1 - \mu)\zeta_{t,i}, & \text{otherwise}. \end{cases} \tag{25}$$

Memory is zero at the start of each episode and is updated after each decision. Before scoring, the stored memory is min-max normalized over the token grid. Section A.8 defines spatial-anchor evidence and its 16 reserved indices. Table 6 lists the scoring weights and retention settings.

A.4 Spatial Priors and Action-Conditioned Evidence

Let $(x_i, y_i) \in [0, 1]^2$ denote normalized coordinates in the vertically composed grid. The indicators $\mathbf{1}_f(i)$ and $\mathbf{1}_d(i)$ equal one when token i belongs to the front-view and down-view regions, respectively,

and zero otherwise. The unnormalized view priors $\widetilde{p}_i^f$ and $\widetilde{p}_i^d$, their combined prior p_i, and the separately normalized priors p_i^f and p_i^d are:

$$\widetilde{p}_i^f = \exp\left[-\left(\frac{|x_i - 0.50|}{0.28}\right)^2 - \left(\frac{|y_i - 0.30|}{0.22}\right)^2\right] \mathbf{1}_f(i), \tag{26}$$

$$\widetilde{p}_i^d = \exp\left[-\left(\frac{|x_i - 0.50|}{0.28}\right)^2 - \left(\frac{|y_i - 0.76|}{0.20}\right)^2\right] \mathbf{1}_d(i), \tag{27}$$

$$p_i = \mathrm{Norm}_i(0.55\widetilde{p}_i^f + 0.45\widetilde{p}_i^d), \tag{28}$$

$$p_i^f = \mathrm{Norm}_i(\widetilde{p}_i^f), \qquad p_i^d = \mathrm{Norm}_i(\widetilde{p}_i^d). \tag{29}$$

For zero-based row u and column v in the 16×16 grid, $x_i = v/15$ and $y_i = u/15$. Rows 0 to 7 form the front-view region and rows 8 to 15 form the down-view region. The constants in the exponentials specify spatial centers and widths, and 0.55 and 0.45 are the view-mixture weights. Local-edge evidence $g_{t,i}$ is the min–max-normalized maximum absolute difference between the token magnitude $u_{t,i}$ and its horizontal or vertical neighbors.

For previous forward and vertical displacements (f, z) in meters and yaw ψ in radians, let $[x]_0^1 = \min(1, \max(0, x))$ denote clipping to $[0, 1]$. The operator $\mathrm{clip}(\cdot, -0.28, 0.28)$ clips its argument to $[-0.28, 0.28]$. The yaw-dependent horizontal center $x_t^\star$ and directional responses $b_{t,i}^f$ and $b_{t,i}^d$ for the two views are:

$$x_t^\star = 0.5 + \mathrm{clip}(0.28\psi/1.1, -0.28, 0.28), \tag{30}$$

$$b_{t,i}^f = \exp\left[-\left(\frac{x_i - x_t^\star}{0.24}\right)^2 - \left(\frac{y_i - 0.32}{0.24}\right)^2\right] \mathbf{1}_f(i), \tag{31}$$

$$b_{t,i}^d = \exp\left[-\left(\frac{x_i - x_t^\star}{0.24}\right)^2 - \left(\frac{y_i - 0.76}{0.24}\right)^2\right] \mathbf{1}_d(i). \tag{32}$$

The normalized forward, vertical, and yaw strengths are $\alpha_f = [f/5]_0^1$, $\alpha_z = [|z|/5]_0^1$, and $\alpha_\psi = [|\psi|/1.1]_0^1$, respectively. Using displacement scales of 5 m and a yaw scale of 1.1 rad, action-conditioned evidence is:

$$\begin{aligned} a_{t,i} = \mathrm{Norm}_i\Big(&0.45p_i^f + 0.35p_i^d + 0.20p_i + \alpha_f b_{t,i}^f + \alpha_z b_{t,i}^d \\ &+ \alpha_\psi(0.70b_{t,i}^f + 0.30b_{t,i}^d)\Big). \end{aligned} \tag{33}$$

Without a previous action, only the first three spatial terms are used. If the previous action has a LAND flag, or satisfies $f < 0.05$, $|z| < 0.05$, and $|\psi| < 0.05$, the score becomes $\mathrm{Norm}_i(0.25p_i^f + 0.75p_i^d)$.

A.5 Observation Uncertainty and Base Budgets

Let $\mathcal{V}_f$ and $\mathcal{V}_d$ be the index sets of front-view and down-view tokens. Set $J = \min\{N, \max(4, \lfloor N/16 \rfloor)\}$, where $\lfloor \cdot \rfloor$ denotes integer floor, and let $s_{t,(j)}$ be the jth largest instruction-relevance score. The factors $c_\Delta = c_C = 3$ calibrate the raw score gap and view difference to $[0, 1]$:

$$\Delta_t^{\mathrm{raw}} = s_{t,(1)} - s_{t,(J)}, \qquad \Delta_t = \min(1, c_\Delta \Delta_t^{\mathrm{raw}}), \tag{34}$$

$$C_t^{\mathrm{raw}} = \left|\mathrm{mean}_{i \in \mathcal{V}_f} s_{t,i} - \mathrm{mean}_{i \in \mathcal{V}_d} s_{t,i}\right|, \qquad C_t = \min(1, c_C C_t^{\mathrm{raw}}). \tag{35}$$

These are the Δ_t and C_t used in Equation 7. Let $c_{t,i} = \cos(\widehat{e}_{t,i}, \bar{q}_t)$ denote the raw cosine score and $\tau_s = 1/8$ the softmax temperature. The semantic probability $P_{t,i}$ and normalized entropy H_t are computed with a positive logarithm floor of 10^{-8}:

$$P_{t,i} = \frac{\exp(c_{t,i}/\tau_s)}{\sum_{j=1}^N \exp(c_{t,j}/\tau_s)}, \qquad H_t = -\frac{\sum_{i=1}^N P_{t,i} \log \max(P_{t,i}, 10^{-8})}{\log N}. \tag{36}$$

For the previous action (f, z, ψ), motion uncertainty is $M_t = \max\{[|\psi|/1.1]_0^1, [|z|/5]_0^1\}$. It is zero at episode start. Table 6 lists the four uncertainty weights, which sum to one.

The thresholds in Equation 8 are $(\tau_{\mathrm{lo}}, \tau_{\mathrm{hi}}) = (0.48, 0.68)$, with base budgets $(K_{\mathrm{lo}}, K_{\mathrm{mid}}, K_{\mathrm{hi}}) = (64, 128, 192)$. The measured mean final budget is 126 tokens.

A.6 Terminal Evidence and Retention Floors

Terminal detection combines normalized semantic scores with recent motion. At step t, $s_f^{\max}$ and $s_d^{\max}$ are the maxima of $s_{t,i}$ over $\mathcal{V}_f$ and $\mathcal{V}_d$, respectively, and $\bar{s}_d$ is the mean over $\mathcal{V}_d$. The center response is $s_d^c = \max_i(s_{t,i}p_i^d)$. Visual terminal evidence is:

$$v_t = \left[0.35s_d^{\max} + 0.20\bar{s}_d + 0.25s_d^c + 0.20\left[\frac{s_d^{\max} - s_f^{\max} + 0.25}{0.50}\right]_0^1\right]_0^1. \tag{37}$$

For the previous action (f, z, ψ), define the slow-forward, vertical-motion, and yaw priors as $h_t^f = [1 - f/1.2]_0^1$, $h_t^z = [|z|/2.5]_0^1$, and $h_t^\psi = [|\psi|/0.45]_0^1$, respectively. These three priors are zero when no previous action is available. The deceleration prior d_t is the fraction of adjacent pairs satisfying $f_{j+1} \le f_j + 0.05$ among up to four recent actions, where f_j is the forward displacement of the jth action in this chronological history. It is zero when fewer than three actions are available. The explicit prior a_t^{land} is one when the previous parsed action has a `LAND` flag and zero otherwise. Motion terminal evidence is:

$$a_t^{\text{term}} = \max\Big\{a_t^{\text{land}},\ \left[0.42h_t^f + 0.28d_t + 0.20h_t^z + 0.10h_t^\psi\right]_0^1\Big\}. \tag{38}$$

The scalar terminal-state score T_t and token-level terminal evidence $\ell_{t,i}$ are:

$$T_t = \max\left\{a_t^{\text{land}},\ 0.68v_t + 0.32a_t^{\text{term}}\right\}, \tag{39}$$

$$\ell_{t,i} = \text{Norm}_i\left(0.52s_{t,i}\mathbf{1}_d(i) + 0.32p_i^d + 0.16a_{t,i}\right). \tag{40}$$

Terminal protection is activated when $T_t \ge 0.72$. The terminal retention floor in Equation 9 is:

$$K_t^{\text{term}} = \begin{cases} 0.75N = 192, & a_t^{\text{land}} = 1, \\ 0.625N = 160, & a_t^{\text{land}} = 0 \text{ and } T_t \ge 0.72, \\ 0, & \text{otherwise.} \end{cases} \tag{41}$$

Terminal protection also raises the down-view minimum to $0.625|\mathcal{V}_d|$. For the adaptive configuration, combining these floors with the base budgets gives possible final retained counts in $\{64, 128, 160, 192\}$.

A.7 Role-Aware Retention and Allocation Settings

Table 6 summarizes the allocation settings. The front-view and down-view regions each contain 128 tokens. Retained indices are sorted in grid order before gathering with Equation 10. $R(x)$ rounds to the nearest integer, with ties rounded to the even integer, as in Python `round`. The motion-activity factor is $\xi_t = \max(a_t^{\text{term}}, M_t)$ and is raised to at least 0.75 when terminal protection is active.

A.8 Integer Quotas and Spatial Anchors

For the final budget K_t in Equation 9, let $\kappa_t = K_t/N$ be the final retention ratio. The implementation converts this ratio to the count $\max(1, \min(N, R(N\kappa_t)))$. The front-view quota is $\rho_f = \min(|\mathcal{V}_f|, \max(1, R(0.25|\mathcal{V}_f|)))$, and the down-view quota is $\rho_d = \min(|\mathcal{V}_d|, \max(1, R(0.20|\mathcal{V}_d|)))$. For $N = 256$, these are 32 and 26. Terminal protection raises the down-view quota to 80. If the quota sum exceeds K_t, the front quota is scaled by $K_t/(\rho_f + \rho_d)$ and truncated to an integer of at least one, and the down quota receives the remainder.

We use zero-based row-major indices $j = 16u + v$, corresponding to mathematical token indices $i = j + 1$. Anchor candidates consist of the 16 positions in the Cartesian product $\{1, 5, 10, 14\} \times \{1, 5, 10, 14\}$, together with four additional positions $(6, 8)$, $(7, 8)$, $(8, 8)$, and $(14, 8)$, for a total of 20 candidates. The spatial-anchor cue $r_{t,i}$ equals one at these 20 positions and zero elsewhere. The reserve selects the 16 largest values of $r_{t,i} + 0.05p_i$, yielding the zero-based indices:

$$\mathcal{A}_0 = \{21, 26, 81, 85, 90, 94, 104, 120, 136, 161, 165, 170, 174, 229, 232, 234\}. \tag{42}$$

Temporal-memory recovery is activated when $U_t \ge 0.48$ and positive memory exists. It selects unreserved tokens in descending memory-score order, up to $\max(1, R(0.125K_t))$ or the remaining

Table 6: Fixed scoring coefficients, budget settings, and retention ratios for state-adaptive token allocation.

Quantity	Value	Role
$(\alpha_u, \alpha_g, \alpha_p)$	$(0.45, 0.35, 0.20)$	Token magnitude, local edges, and spatial prior
(w_s, w_n, w_a)	$(0.28, 0.30, 0.20)$	Instruction, structure, and action cues
(w_m, w_r, w_g)	$(0.14, 0.08, 0.04)$	Memory, anchor, and edge cues
$w_\ell(t)$	$0.04,\ 0.10$	Normal and terminal states
Uncertainty weights	$(0.42, 0.28, 0.18, 0.12)$	$(\beta_H, \beta_\Delta, \beta_C, \beta_M)$
Range factors	$(c_\Delta, c_C) = (3, 3)$	Semantic separation and view disagreement
Budget thresholds	$(0.48, 0.68)$	$(\tau_{\mathrm{lo}}, \tau_{\mathrm{hi}})$
Base budgets	$(64, 128, 192)$	$(K_{\mathrm{lo}}, K_{\mathrm{mid}}, K_{\mathrm{hi}})$
View minima	$32,\ 26$	Rounded front-view and down-view quotas
Terminal floor	$0.625N$	Minimum budget of 160
Explicit-LAND floor	$0.75N$	Minimum budget of 192
Terminal down-view minimum	$0.625\lvert\mathcal{V}_d\rvert = 80$	Down-view quota during terminal protection
Terminal threshold	0.72	Terminal protection activation
Anchor reserve	$0.0625N = 16$	Distributed spatial anchors
Memory reserve	$\max(1, R(0.125K_t))$	Requested memory-recovery count
Memory update	$\nu = 0.75,\ \mu = 0.86$	Mixing weight and decay
Action history	16	Maximum previous actions
Edge rescue	$R(0.10K_t)$	Boundary evidence
Motion rescue	$R(0.08K_t\xi_t)$	Motion evidence

capacity. The allocator then fills the front-view quota followed by the down-view quota using the combined evidence score.

Edge rescue adds up to $R(0.10K_t)$ tokens ranked by $g_{t,i}$. Motion rescue uses the score $\mathrm{Norm}_i(0.52a_{t,i} + 0.28g_{t,i} + 0.20\ell_{t,i})$ and requests $R(0.08K_t\xi_t)$ tokens using the motion-activity factor ξ_t defined above. Both rescue stages are limited by remaining capacity. If quota filling exceeds K_t, anchor, memory, and rescue tokens receive a score bonus of 2, and the top K_t tokens are selected across the full grid. Otherwise, remaining slots are filled in descending evidence-score order.

B Implementation Details

B.1 Complementary-Teacher Distillation

The student is initialized with DINOv3 ViT-B/16 weights pretrained on LVD-1689M. For the student, the composed RGB observation is recovered from the policy processor, bicubically resized to 256×256, and normalized from pixel values in $[0, 1]$ using channel means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$ in RGB order. Prefix tokens are removed to obtain $N = 256$ tokens on a 16×16 patch grid. Both teachers receive 224×224 inputs and provide 16×16 patch-token grids. DINOv2's class token and four register tokens (Darcet et al., 2024) are excluded.

Stage I uses four GPUs (16 samples per GPU), seed 42, BF16, and AdamW with moment-decay coefficients $(\beta_1, \beta_2) = (0.9, 0.95)$. Weight decay is 0.05. The learning rate uses linear warmup over the first 3% of training steps followed by cosine decay, and the gradient norm is clipped at 1.0.

B.2 VLA Training with Dynamic Visual Token Inputs

The multimodal projector is initialized from the dense VLA baseline. LoRA is applied to the language backbone's `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` layers with scaling parameter $\alpha = 128$ and dropout probability 0.05. Training uses four GPUs, BF16,

and gradient checkpointing (Chen et al., 2016). We use AdamW with a weight decay of 0.03. The learning rate warms up over the first 3% of training steps before following a cosine schedule, and gradients are clipped at a norm of 1.0.

During shuffled-frame training, action-conditioned evidence uses its spatial-prior terms. Temporal-memory responses and previous-motion inputs are zero. Instruction relevance averages prompt embeddings while excluding padding, the beginning-of-sequence token, and action targets.

Within each minibatch, selected visual sequences are padded to the largest retained count, with padding excluded by the attention mask. Token indices are selected without gradient tracking. Gradients flow through the gathered features to the trainable projector.

Action-bin indices are serialized as space-separated, zero-padded two-digit integers. The training loader appends `LAND` to the final and penultimate expert steps.

C Computational Efficiency

The two stages of AeroC^2 reduce computation at complementary parts of the visual pathway.

First, complementary-teacher visual distillation reduces the cost of dense visual representation extraction. Let C_T denote the total inference cost of the teacher encoders and C_S the cost of the distilled student pathway, including the lightweight student encoder and feature translators:

$$C_T = \sum_{b \in \mathcal{B}} C_{T_b}, \qquad C_S = C_{S_\phi} + C_G, \tag{43}$$

where C_{T_b} denotes the cost of teacher encoder b, C_{S_ϕ} the cost of the student encoder, and C_G the total cost of the feature translators. Since the student encoder is substantially smaller than the teacher encoders and the translators consist only of lightweight MLP modules, we have:

$$C_S \ll C_T. \tag{44}$$

After Stage I, the teacher encoders are removed and only the lightweight student pathway is retained for inference, substantially reducing the visual encoding cost at each control step.

Second, closed-loop visual token allocation reduces the visual sequence processed by the VLA backbone from N tokens to a dynamically selected budget K_t, where $K_t < N$. Because token allocation is performed after the multimodal projector, the projector still operates on the dense N-token representation. The main downstream saving therefore arises from shortening the sequence entering the VLA backbone. Let L_t denote the number of non-visual tokens at time t. Since self-attention scales approximately quadratically with sequence length (Vaswani et al., 2017), reducing the visual-token count from N to K_t decreases the corresponding attention complexity from $\mathcal{O}((L_t + N)^2)$ to $\mathcal{O}((L_t + K_t)^2)$. Because K_t varies with the current observation, the realized computational benefit depends on the empirical distribution of token budgets during closed-loop execution.

D Additional Experimental Details and Results

D.1 Efficiency Measurement Protocol

The efficiency evaluation subset contains 150 clips sampled with seed 42, each with 16 measured decisions. Sampling is balanced between Easy and Hard routes within each split. For history-dependent methods, 16 preceding observations initialize temporal state before each measured clip. Decisions within the clip run sequentially.

We measure BF16 inference with batch size 1 on an NVIDIA GeForce RTX 4090. We use PyTorch 2.1.2 and CUDA 11.8. After 32 warm-up decisions, CUDA is synchronized before starting and stopping each decision timer. After warm-up, timing starts with preprocessing of already loaded images and ends after action decoding, covering the intervening VLA inference computation.

We count FLOPs with `torch.profiler` for supported matrix multiplications and two-dimensional convolutions. Profiling is separate from latency measurement. For adaptive allocation, we replay one representative decision per observed execution-token bucket, including terminal-adjusted counts, and weight its FLOPs by the bucket frequency across 2,400 measured decisions. Mean latency is the arithmetic mean of the 2,400 per-decision timings.

D.2 Detailed Navigation Comparison by Difficulty

Tables 7, 8, and 9 report results on the three TravelUAV splits. Easy routes have a reference-trajectory length of at most 250 m, computed by summing distances between consecutive waypoints. Hard routes exceed 250 m. Full contains both groups. NE is measured in meters, while SR, OSR, and SPL are percentages. Baseline results are taken from AeroVLA (Xu et al., 2026). We evaluate AeroC^2 using the same public splits and metric definitions. Bold and underlined values indicate the best and second-best results among autonomous methods.

Table 7: Navigation performance on the Seen split across Full, Easy, and Hard routes.

Method	Full				Easy				Hard			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Human	14.15	94.51	94.51	77.84	11.68	95.44	95.44	76.19	17.16	93.37	93.37	79.85
Random Action	222.20	0.14	0.21	0.07	142.07	0.26	0.39	0.13	320.12	0.00	0.00	0.00
Fixed Action	188.61	2.27	8.16	1.40	121.36	3.48	11.48	2.14	270.69	0.79	4.09	0.49
CMA	135.73	8.37	18.72	7.90	84.89	11.48	24.52	10.68	197.77	4.57	11.65	4.51
TravelUAV-DA	98.66	17.45	48.87	15.76	66.40	20.26	51.23	18.10	138.04	14.02	45.98	12.90
NavFoM	93.05	29.17	49.24	25.03	58.98	32.91	53.16	27.87	143.83	23.58	43.40	20.80
LongFly	**60.02**	36.39	**65.87**	31.07	**38.10**	38.52	**71.90**	31.24	**85.20**	33.94	**58.94**	30.88
AeroVLA	65.88	<u>47.96</u>	57.69	<u>38.54</u>	43.76	<u>49.30</u>	61.30	<u>37.14</u>	<u>93.16</u>	<u>46.30</u>	53.23	<u>40.26</u>
AeroC^2	<u>64.93</u>	**51.48**	<u>59.94</u>	**41.02**	<u>40.05</u>	**53.26**	<u>64.11</u>	**39.54**	95.60	**49.29**	<u>54.80</u>	**42.86**

Table 8: Navigation performance on the Unseen Object split across Full, Easy, and Hard routes.

Method	Full				Easy				Hard			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Random Action	260.14	0.16	0.16	0.16	174.10	0.48	0.48	0.48	302.96	0.00	0.00	0.00
Fixed Action	212.84	3.66	9.54	2.16	151.66	6.70	13.88	3.72	243.29	2.14	7.38	1.38
CMA	155.79	9.06	16.06	8.68	102.92	14.83	22.49	13.90	182.09	6.19	12.86	6.08
TravelUAV	118.11	22.42	46.90	20.51	86.12	24.40	49.28	22.03	134.03	21.43	45.71	19.75
NavFoM	108.04	29.83	47.99	27.20	70.51	32.54	50.72	29.54	133.01	28.03	46.18	25.64
LongFly	66.74	43.87	64.56	38.39	54.84	38.01	56.84	31.36	**57.07**	50.25	**74.16**	45.27
AeroVLA	<u>61.45</u>	<u>56.60</u>	<u>64.86</u>	<u>46.61</u>	<u>45.72</u>	<u>56.94</u>	<u>64.11</u>	<u>43.76</u>	69.27	<u>56.43</u>	65.24	<u>48.03</u>
AeroC^2	**53.77**	**59.78**	**68.36**	**49.61**	**34.18**	**65.55**	**73.68**	**49.79**	<u>63.52</u>	**56.90**	<u>65.71</u>	**49.52**

Table 9: Navigation performance on the Unseen Map split across Full, Easy, and Hard routes.

Method	Full				Easy				Hard			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Random Action	202.98	0.00	0.00	0.00	158.46	0.00	0.00	0.00	265.88	0.00	0.00	0.00
Fixed Action	180.47	0.52	2.61	0.39	132.89	0.89	4.28	0.67	247.72	0.00	0.25	0.00
CMA	141.68	2.30	10.02	2.16	102.29	3.57	14.26	3.33	197.35	0.50	4.03	0.50
TravelUAV	138.80	4.18	20.77	3.84	102.94	4.63	22.82	4.24	189.46	3.53	17.88	3.28
NavFoM	125.10	6.30	18.95	5.68	102.41	6.77	20.07	6.04	170.58	5.36	15.71	4.97
LongFly	108.32	11.27	30.27	9.32	78.56	12.96	34.31	10.32	148.10	9.02	24.88	7.98
AeroVLA	<u>67.42</u>	<u>37.58</u>	<u>52.92</u>	<u>28.22</u>	**44.99**	<u>41.89</u>	<u>58.47</u>	<u>29.72</u>	<u>99.11</u>	<u>31.49</u>	<u>45.09</u>	<u>26.11</u>
AeroC^2	**59.79**	**42.38**	**57.62**	**31.82**	<u>47.30</u>	**43.67**	**59.18**	**30.57**	**77.45**	**40.55**	**55.42**	**33.57**