AeroEVA: Unified World Modeling and VLM Policy Learning for Embodied Aerial Search

Anonymous authors
Paper under double-blind review

Abstract

Efficient aerial search for human-specified goals requires anticipating what each move will reveal. Content next to the current view is often predictable even when distant regions are not, yet existing agents remain limited in turning such local foresight into better actions. We introduce **AeroEVA** (*Envision, Verify, and Act*), a vision-language model (VLM) agent that uses transition prediction in two roles: as a training signal, and as evidence checked before each move. To *envision*, its vision tower learns to predict features of the nearest patches in the neighboring region under each action, and textual transition targets supervise the decoder alongside actions. To *verify*, AeroEVA reuses this predictor to score each candidate move against an aerial-image goal. To *act*, the policy judges when these one-step scores are relevant and combines them with observations and history. Imitation learning alone surpasses prior search agents, and multi-turn reinforcement learning widens the margin to nearly 20 points in success rate. Without retraining, AeroEVA also leads prior methods in unseen regions, with ground-photo and text goals, under disaster-induced appearance change, and on finer grids. We further introduce AeroTask, 696 search tasks across 13 application scenarios whose language goals state task requirements rather than target appearance. Code, model weights, and the AeroTask dataset will be released upon acceptance.

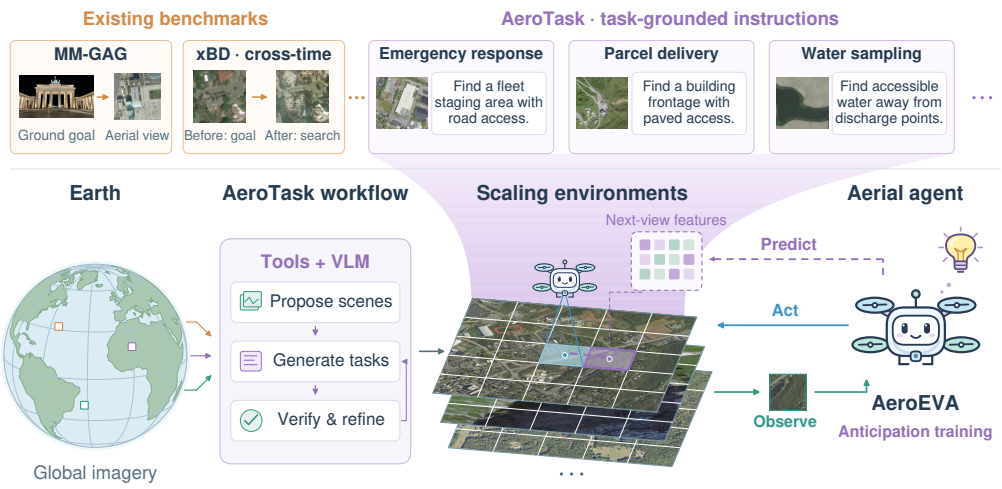


Figure 1: ***Can an aerial agent actively navigate and search for goals specified through images or human instructions?*** Our AeroTask workflow turns sampled Earth imagery into task-grounded environments at scale (purple gradient), complementing image-based benchmarks (orange) with application-grounded instructions (purple). AeroEVA learns to anticipate movement consequences through transition prediction.

1 Introduction

Embodied aerial search is a fundamental capability for aerial agents: actively navigating a partially observed area to locate and reach a human-specified goal. We study its *active geo-localization* (AGL) formulation, in which an agent uses successive local aerial views to reach an image- or language-specified target within a limited movement budget (Pirinen et al., 2022; Sarkar et al., 2024; Mi et al., 2025). Prior AGL work highlights its importance for search-and-rescue, where limited field

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.

of view, battery life, and urgency make efficient exploration essential (Pirinen et al., 2022; Sarkar et al., 2024; Mi et al., 2025). The same capability can support infrastructure inspection, delivery-site selection, and emergency response (Figure 1).

Efficient search requires anticipating what a movement will reveal and using that anticipation to select actions (Figure 2). In our image-based simulator built from real aerial and satellite imagery, agent trajectories provide observed transitions as direct supervision for learning movement consequences. Inspired by work that derives auxiliary supervision from environmental transitions (Wang et al., 2026; Liu et al., 2026), we use visual and semantic changes to learn how actions affect local aerial views. Local spatial dependence supports predicting nearby content even when entire unseen regions are difficult to infer. Learning to predict, however, does not by itself ensure that a language policy can use predictive representations or determine which movement best serves the search goal. We therefore connect visual prediction to language reasoning through textual transition supervision, and turn predicted outcomes into explicit goal-compatibility evidence for action selection. This second role draws on goal-directed evaluation of latent predictions, as in DINO-WM (Zhou et al., 2025). Together, these two roles connect learning from observed transitions with using predicted transitions to assess candidate actions before acting.

Our agent, **AeroEVA**, combines *Aero* (aerial) with *EVA* (Envision, Verify, and Act). Conditioned on an action, it predicts DINOv3 features (Siméoni et al., 2025) of patches in the next region closest to the current view. This visual objective updates the VLM's vision tower, while *textual environmental feedback* converts observed transitions into structured language targets. Predicting these semantic changes from visual features, actions, and interaction context trains the decoder to use the tower's predictive representations. A *goal-compatibility verifier tool* compares predicted features for candidate movements with an aerial-image goal before a candidate movement is executed. The policy uses its goal-adjacency judgment to assess when this local evidence is relevant, then combines the scores with observations and history to select actions.

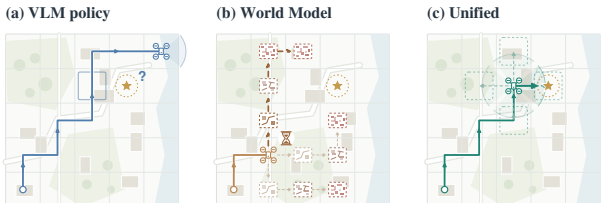


Figure 2: **AeroEVA combines latent foresight with VLM reasoning for effective search.** (a) Reward-driven VLM policies can miss nearby goals due to overfitting. (b) World models enable counterfactual evaluation of candidate actions, but latent rollouts are costly and can accumulate prediction errors. (c) Joint training teaches the agent the consequences of movement; the VLM integrates one-step predictions with goals and history to select actions, addressing missed nearby goals without costly rollouts. Solid and dashed paths denote movement and prediction, respectively.

Our contributions are threefold:

- **World modeling for aerial foresight.** We develop *Envision* through action-conditioned prediction of neighboring patch tokens and textual transitions, connecting spatial prediction in the vision tower to semantic reasoning in the decoder.
- **Foresight-guided VLM policy learning.** We turn predicted features into one-step goal-compatibility evidence (*Verify*) and train a policy to use it (*Act*) through joint action–transition SFT followed by multi-turn RL.
- **AeroTask and empirical evaluation.** We evaluate search, geographic transfer, and goal-modality generalization, and introduce AeroTask: 696 tasks across 13 application scenarios, constructed through an agentic workflow for task-grounded language goals. The workflow scales environment construction for aerial embodied agents.

2 Related Work

Active geo-localization and embodied aerial agents. Aerial embodied agents can support urban life through language-guided navigation (Liu et al., 2023; Gao et al., 2026; Lee et al., 2025) and autonomous object search (Ji et al., 2026; Xiao et al., 2025). Active geo-localization (AGL) is an embodied aerial search task in which an agent navigates through partially observed overhead imagery to reach a target specified by an aerial image, a ground-level image, or text within a limited movement budget (Sarkar et al., 2024). Goals specify what to find, and the agent chooses movements to acquire observations and locate the hidden target without route instructions. The grid of real

overhead imagery supports geographic expansion without scene-specific 3D reconstruction, while abstracting flight dynamics.

Within AGL, AiRLoc learns aerial search policies with RL (Pirinen et al., 2022). GOMAA-Geo combines cross-modal alignment, goal-aware action pretraining, and RL (Sarkar et al., 2024). GeoExplorer jointly models actions and state transitions, using state-prediction error as an intrinsic reward for exploration during RL (Mi et al., 2025). For aerial VLN, WorldVLN combines latent world prediction with waypoint decoding (Zhao et al., 2026). Our method predicts neighboring-view patch features and computes goal-compatibility scores for candidate grid moves, providing explicit evidence to the VLM policy during action selection.

Predictive visual representations. I-JEPA and V-JEPA learn predictive representations through masked image and video feature prediction (Assran et al., 2023; Bardes et al., 2024). Self-predictive representations add action-conditioned latent prediction for model-free RL (Schwarzer et al., 2020), while DINO-WM and V-JEPA 2 learn visual dynamics for planning and robotic control (Zhou et al., 2025; Assran et al., 2025). Recent work transfers predictive knowledge to VLMs: World2VLM derives spatial-reasoning supervision from camera-conditioned future views (Zhang et al., 2026b), and GeoWorld-VLM distills geometric structure from video world models into the VLM's visual pathway (Gu et al., 2026). Beyond video prediction and 3D spatial understanding, we study which prediction targets yield action-sensitive geospatial representations across non-overlapping aerial views.

World modeling for policy learning. SPA initializes textual state and transition modeling through SFT before RL (Chen et al., 2025), while VAGEN rewards explicit state estimation and transition reasoning during multi-turn RL (Wang* et al., 2025). EnvRL, PaW, and TAPO incorporate textual environment-transition supervision into policy optimization (Wang et al., 2026; Lu et al., 2026; Li et al., 2026a). Recent work extends predictive supervision to visual representations: GLANCE aligns linguistic future-state predictions with visual latents and uses prediction error to drive curiosity during RL (Li et al., 2026b); FutureNav jointly learns navigation actions, dynamics, and future spatial states (Zhang et al., 2026a). We jointly learn actions, multimodal transitions during SFT, choosing neighboring-patch prediction targets through analysis of aerial spatial dynamics.

World-model–agent interaction. Navigation World Models simulates action-conditioned visual futures to plan trajectories or rank candidates proposed by an external policy (Bar et al., 2025). Astra supplies imagined views from a camera-conditioned simulator (Zhu et al., 2026), while CoMAP uses textual world-model feedback for policy reflection (Liu et al., 2026). Simulation access alone does not ensure effective use of predicted outcomes (Qian et al., 2026). Latent world models offer interfaces beyond generated images: FOREWARN turns predicted latent trajectories into behavior narrations for VLM-based selection of candidate plans (Wu et al., 2025), and ω-EVA uses imagined latent consequences to directly refine proposed action chunks (Sun et al., 2026). Our VLM selects moves by combining goal-compatibility scores from one-step local latent predictions with the aerial-image goal, current observation, and history.

3 Problem Formulation

An active geo-localization episode $e = (I, p_0, g, p_g)$ specifies overhead imagery I, start p_0, goal description g, and hidden target p_g (Sarkar et al., 2024; Mi et al., 2025). On an $H \times W$ grid of non-overlapping patches, the agent observes local patch o_t at p_t and chooses $a_t \in \mathcal{A} = \{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}\}$:

$$p_{t+1} = \mathcal{T}(p_t, a_t), \qquad o_{t+1} = \mathcal{O}(I, p_{t+1}). \tag{1}$$

The grid abstracts flight dynamics; off-grid moves consume one step without displacement.

The VLM policy $\pi_\theta(a_t \mid h_t, g)$ parses actions from generated responses. History h_t includes local observations, actions, responses, current position, step count, and valid directions; I and p_g remain hidden. Search ends on goal entry or after B moves, without a stop action. We maximize success over episodes $e \sim \mathcal{D}$ and policy trajectories:

$$J(\theta) = \mathbb{E}_{e\sim\mathcal{D},\,\pi_\theta}\left[\mathbf{1}\{\exists\, t \in \{0,\dots,B\} :\; p_t = p_g\}\right]. \tag{2}$$

4 Method

AeroEVA unifies policy learning and world modeling through *Envision, Verify, and Act* (EVA; Figure 3). Visual and textual transition supervision develop anticipation; a frozen verifier compares

predicted outcomes with aerial-image goals, and the policy combines this evidence with observations and history. Joint action–transition SFT trains these capabilities before multi-turn RL refines decision making; Appendix C gives implementation details.

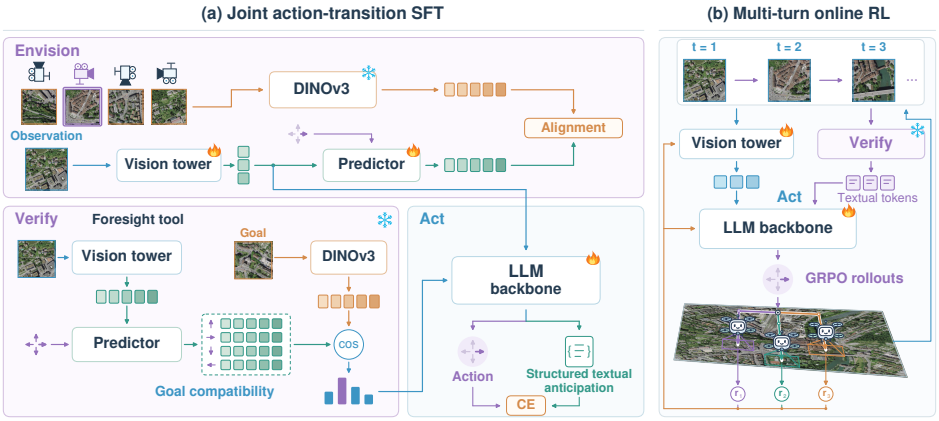


Figure 3: **AeroEVA: Envision, Verify, and Act.** (a) Joint action–transition SFT. *Envision* aligns predicted neighboring patch tokens with frozen DINOv3 targets from direction-matched GT views; the purple-outlined view matches the illustrated rightward action. *Verify* compares four directional latent predictions with goal features via cosine similarity to obtain goal-compatibility evidence. *Act* combines visual tokens and textual evidence in the LLM to generate actions and structured textual anticipation under shared cross-entropy loss (CE, $\mathcal{L}_{\text{tok}}$). (b) Multi-turn online GRPO retains frozen *Verify*, which supplies textual evidence for *Act*. Flames and snowflakes mark trainable and frozen modules, respectively.

4.1 Envision: Multimodal World Modeling for VLM Agents

Offline random walks through training grids supply recorded transitions (o_t, a_t, o_{t+1}): an observation, executed action, and resulting view. These provide visual targets and semantic transition labels for *Envision*.

Visual latent transition supervision. We predict *neighboring patch tokens*: DINOv3 (Siméoni et al., 2025) features of a single neighboring row (or column, depending on the action) in the next cell, using nearby current-view tokens. This local target follows the distance decay of spatial dependence.

Let E_θ be the vision encoder within the VLM policy π_θ, where θ denotes all VLM parameters, and let f_ϕ be the visual predictor with parameters ϕ. The post-merger tokens $E_\theta(o_t) \in \mathbb{R}^{n\times n\times d_v}$ of our Qwen3-VL backbone (Bai et al., 2025) are taken before they enter the language decoder. We use a patch-selection operator $S_a(X, w)$ to extract a band of width w from the side of a token grid X in direction a. The predictor maps this band to neighboring patch tokens in the next observation:

$$X_t^a = S_a(E_\theta(o_t), w), \qquad \widehat{Z}_{t+1}^a = f_\phi(X_t^a, a). \tag{3}$$

Here $\widehat{Z}_{t+1}^a \in \mathbb{R}^{K\times d_z}$ contains the predicted neighboring patch tokens. The frozen target encoder E_{DINO} is a DINOv3 ViT-L/16 (Siméoni et al., 2025) that supplies a $K \times K$ grid of d_z-dimensional tokens. Write $\bar{a}$ for the direction opposite a. For a recorded transition (o_t, a_t, o_{t+1}), the target is $Z_{t+1} = S_{\bar{a}_t}(E_{\text{DINO}}(o_{t+1}), 1)$. We minimize the mean squared error between token-normalized predictions and targets:

$$\mathcal{L}_{\text{vis}}(\theta, \phi) = \frac{1}{K}\,\mathbb{E}_{(o_t,a_t,o_{t+1})\sim\mathcal{D}_{\text{tr}}}\left[\left\|\mathcal{N}(\widehat{Z}_{t+1}^{a_t}) - \mathcal{N}(Z_{t+1})\right\|_F^2\right], \tag{4}$$

where $\mathcal{D}_{\text{tr}}$ is the recorded transition set, $\mathcal{N}(X)_j = X_j/\|X_j\|_2$ normalizes each token, and $\|\cdot\|_F$ denotes the Frobenius norm. This equals the mean tokenwise loss $2(1-\cos)$, with fixed, detached target features. It directly trains the vision tower and predictor to model post-navigation observation changes, motivated by weaker representations of environment dynamics in deeper language layers (Figure 4). The predictor is goal-independent; goal conditioning enters the tool comparison in Section 4.2.

Predictor design. The predictor applies local convolutions along the selected token row or column. Action conditioning enters through token selection and a learned action embedding. Appendix C.1 gives the layer dimensions, feature alignment, and token ordering.

Textual transitions for language reasoning. Textual transition supervision trains the language decoder to use the tower's predictive features: given current visual tokens, interaction context, and action, it predicts next-view land cover, entry content, coverage changes, and salient objects. Offline semantic maps supply these labels; neither maps nor labels enter observation prompts. Generated after the action, this text shapes the policy through joint SFT (Section 4.3). Appendix C.2 details the schema and label rules.

4.2 Verify and Act: Foresight-Guided Decision Making

Verify turns predicted movement consequences into goal-compatibility evidence and pairs it with the policy's goal-adjacency judgment. *Act* uses this evidence together with observations and interaction history to choose the next move.

Candidate outcome prediction. We reuse the visual predictor from Section 4.1 as a foresight tool, available only for aerial-image goals g. For each legal action a, it evaluates Eq. 3 to predict neighboring patch tokens. The tool uses a matched, frozen vision encoder E_{θ_0} and predictor f_{ϕ_0}; the initialization of this snapshot is described in Section 4.3.

Verify: score compatibility and assess proximity. Let $Z_g^a = S_{\bar{a}}(E_{\mathrm{DINO}}(g), 1)$ denote the goal patch tokens at the positions selected for action a. We define goal compatibility through the normalized latent distance:

$$c_t(a) = 1 - \frac{1}{2K} \left\| \mathcal{N}(\widehat{Z}_{t+1}^a) - \mathcal{N}(Z_g^a) \right\|_F^2 . \tag{5}$$

This score equals the mean cosine similarity of aligned patch tokens, so larger values indicate greater goal compatibility. It uses the same normalized feature distance as Eq. 4, with the selected goal patch tokens as the target. It is computed from the current observation and the goal alone, without access to the unseen neighboring view. The goal need not share acquisition time with the search imagery; Section 5.5 pairs pre-disaster goals with post-disaster observations.

Alongside the verifier scores, the policy generates its own goal-adjacency belief $\widehat{b}_t$ before the action. Its target is $b_t = \mathbf{1}\{\|p_t - p_g\|_1 = 1\}$: whether the goal lies within the verifier's one-step horizon. When the goal is adjacent, the comparison directly tests a candidate goal location, and the belief tells the policy when this holds. The scores are input evidence, while the belief is a policy output that assesses proximity from observations and history. Training goal positions supply belief labels without entering observation prompts; Section 4.3 describes the supervision and reward.

Act: select and execute a move. The four directional scores are inserted into the observation prompt, with unavailable entries represented by `null`. The policy weighs them against its observations and history to choose a move. This is explicit, world-model-based counterfactual lookahead over one-step actions. After executing a move, the agent receives a real observation and repeats the process rather than recursively rolling out predicted latents.

4.3 Unified Policy Learning and World Modeling

Joint action–transition SFT. For each state in the trajectories from Section 4.1, we retain the recorded behavior action a_t and assign an optimal action label $a_t^\star$ that reduces Manhattan distance to the goal. Action supervision uses $a_t^\star$, whereas visual and semantic transition targets describe the observed successor under a_t. A single weighted token loss $\mathcal{L}_{\mathrm{tok}}$ supervises the action, goal-adjacency belief, next-view description, and response format. We combine it with visual prediction:

$$\mathcal{L}_{\mathrm{SFT}} = \mathcal{L}_{\mathrm{tok}} + 0.5\mathcal{L}_{\mathrm{vis}}. \tag{6}$$

We first initialize the VLM and predictor through action imitation and visual transition learning. The resulting snapshot (θ_0, ϕ_0) supplies the frozen verifier in Section 4.2. We then train the full VLM and predictor with Eq. 6, while this independent scoring branch remains fixed.

During full joint SFT and subsequent GRPO (Shao et al., 2024), we drop tool evidence with probability $p = 0.5$ per trajectory or episode, keeping its availability fixed across turns. The policy thus learns goal-adjacency judgments and action selection both with and without foresight evidence. Appendices C.2 and C.3 provide supervision and training details; Section 5.4 defines the component ablations.

Multi-turn RL refinement. Starting from the joint-SFT checkpoint, multi-turn GRPO (Shao et al., 2024; Wang* et al., 2025) refines *Verify* and *Act* through interaction. It samples $G = 8$ trajectories sharing the goal, start state, and tool-availability mask. Rewards score goal reaching,

response-format compliance, and belief accuracy, without distance-progress shaping. Group-relative trajectory advantages supervise all generated assistant tokens with entropy regularization.

RL updates the full VLM, including its vision tower, while the matched tool encoder and predictor remain frozen. Direct visual prediction and next-view semantic reconstruction losses are disabled. Appendix C.4 gives the advantage normalization, trajectory objective, reward weights and gates, and optimization settings.

5 Experiments

5.1 Experimental Setup

Evaluation protocol. We follow the search setting and established benchmark protocols of GOMAA-Geo and GeoExplorer (Sarkar et al., 2024; Mi et al., 2025): 5×5 grids, four movement actions, and a budget of $B = 10$ moves. Success means reaching the goal cell within the budget. We denote the initial Manhattan distance by $C = \|p_0 - p_g\|_1$. Dataset-specific sampling details are in Appendix A.2.

Distance range and reporting. On 5×5 grids, $C7$ and $C8$ have only 16 and four ordered start–goal geometries, respectively, so success can reflect geometry-specific strategies. We therefore restrict training initializations to $C3$–$C6$ and average results over this range for the core ablation (Table 2) and AeroTask. Following prior benchmark protocols, we report $C4$–$C8$ results and their Avg on established benchmarks, alongside the $C4$–$C6$ mean. We select checkpoints on in-distribution validation episodes and use greedy decoding.

Benchmark datasets. MASA (Pirinen et al., 2022) supplies training and in-domain evaluation scenes. We test transfer across time on xBD-disaster (Gupta et al., 2019); across regions on SwissView100 (Mi et al., 2025), MM-GAG (Sarkar et al., 2024), and xBD-pre; across grid resolution on MASA 10×10; and across goal modalities on MM-GAG and SwissViewMonuments (Mi et al., 2025). AeroTask evaluates task-grounded language goals without tool evidence. Appendix A.2 details each protocol.

Baselines and policy backbones. We compare against Random, PPO (Schulman et al., 2017), AiRLoc (Pirinen et al., 2022), Decision Transformer (DiT) (Chen et al., 2021), GOMAA-Geo (Sarkar et al., 2024), GeoExplorer (Mi et al., 2025), and DINO-WM (Zhou et al., 2025). With Qwen3-VL-4B (Bai et al., 2025), we include zero-shot inference, cold-start GRPO, Action SFT + GRPO, and a GeoExplorer-Qwen reimplementation, alongside AeroEVA (SFT) and AeroEVA (SFT + GRPO). We also include GPT-5.4-mini (OpenAI, 2026) as a zero-shot baseline. Implementation details and result sources are in Appendix A.1.

5.2 What visual representations do we need for policy learning?

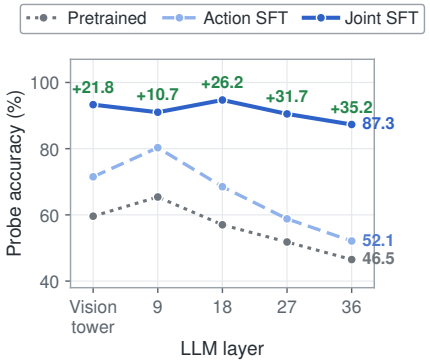


Figure 4: **Do VLM representations capture environment dynamics?** We probe inverse dynamics by predicting actions from consecutive views. Green: Joint SFT minus Action SFT.

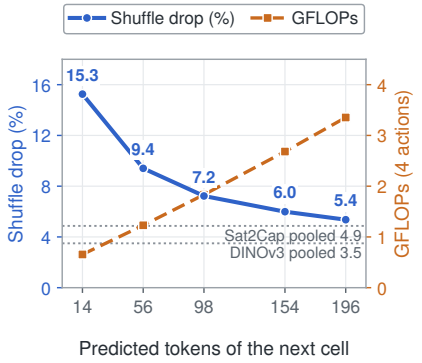


Figure 5: **How far into the next cell should prediction reach?** Cosine drop under shuffled actions (left) and head cost (right) as the target grows from the nearest row (14 tokens) to the whole next cell (196).

We examine how visual representations encode action-dependent dynamics and which prediction targets preserve action sensitivity. An inverse-dynamics probe recovers the intervening action from

adjacent views (four classes; chance 25%). Readouts share the architecture and MASA training/evaluation splits (Figure 4). Action SFT and Joint action–transition SFT use matched data, training budgets, schedules, and trainable parameters. Appendix B.1 provides protocol details and uncertainty estimates.

Environment dynamics across representation depth. Action SFT improves dynamics decodability at the vision tower, but decodability declines in deeper decoder layers (Figure 4). This motivates supervising predictions of post-navigation features at the vision tower (Eq. 4). Joint visual and textual supervision improves decodability at both the tower and deeper layers.

Prediction targets and action sensitivity. Predicting the nearest row of the next cell best preserves action sensitivity at the lowest patch-level cost (Figure 5). Extending coverage makes prediction harder and less action-dependent, while pooled features can achieve high similarity with weak action sensitivity. These findings support local patch targets for learning action-dependent dynamics.

5.3 Main Results

Strong SFT initialization and further gains from RL. AeroEVA (SFT) reaches 71.8% Avg on MASA test (Table 1), above every baseline evaluated on our episodes: released GOMAA-Geo (64.6%), GeoExplorer under its original protocol (66.4%), the $C3$–$C6$ retrained baselines, and the same-backbone GeoExplorer-Qwen (47.9%). Multi-turn GRPO raises success to 86.1%, a 14.3-point gain and the highest Avg and $C4$–$C6$ mean in Table 1. Section 5.4 examines the contributions of transition supervision and verifier evidence.

Table 1: **Comparing policy learning and latent world-model planning on MASA.** Test success (%) with $B = 10$; Avg: mean over $C4$–$C8$. $*$: results excerpted from GeoExplorer (Mi et al., 2025); remaining rows: our evaluation.

Method	$C4$	$C5$	$C6$	$C7$	$C8$	Avg	$C4$–$C6$
Original training protocols							
Random*	14.1	5.8	6.4	2.5	2.4	6.2	8.8
PPO*	14.3	17.8	19.2	22.7	26.0	20.0	17.1
AiRLoc*	17.9	15.6	21.3	24.2	23.9	20.6	18.3
DiT*	20.1	29.6	35.7	42.2	45.6	34.6	28.4
GOMAA-Geo	41.8	50.1	71.7	80.7	78.9	64.6	54.5
GeoExplorer	43.0	50.9	76.8	80.0	81.5	66.4	56.9
Zero-shot							
Qwen3-VL-4B	13.2	9.2	5.1	2.1	1.6	6.2	9.2
GPT-5.4-mini	33.8	26.4	27.9	19.1	18.8	25.2	29.4
Our training protocol ($C3$–$C6$ only)							
GOMAA-Geo	40.1	50.4	72.5	81.1	80.7	65.0	54.3
GeoExplorer	41.9	45.6	54.3	59.4	52.8	50.8	47.3
GeoExplorer-Qwen	40.8	46.5	60.3	45.5	46.4	47.9	49.2
DINO-WM	63.5	49.2	50.5	36.1	36.1	47.1	54.4
AeroEVA (SFT)	54.3	64.0	78.3	86.4	76.0	71.8	65.5
AeroEVA (SFT + GRPO)	75.9	78.7	92.7	92.0	91.1	86.1	82.4

5.4 Ablation Study

Table 2 compares action-supervised variants and RL baselines. Visual latent adds visual prediction; +text adds next-view text targets, +tool adds verifier inputs and goal-adjacency targets, and full combines both. GRPO starts from the pretrained policy or the named SFT checkpoint; AeroEVA's scoring branch stays frozen.

Connecting foresight to action. Visual latent supervision alone falls below Action SFT (29.7% versus 33.3%), so predictive features must be connected to action. Adding text or the tool configuration raises success to 36.3% or 37.8%; combining both reaches 59.8%.

Imitation learning and RL refinement. Full SFT exceeds cold-start GRPO (52.2%) and Action SFT + GRPO (56.4%) on the same backbone. Further GRPO raises its validation success from 59.8% to 77.0%, a 17.2-point gain, leading at every reported distance (Table 2).

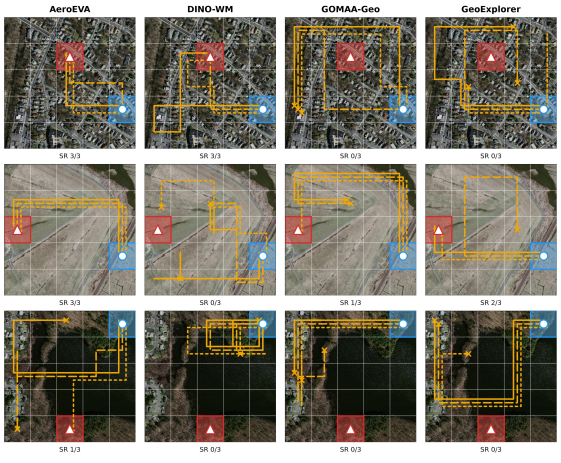


Figure 6: **Qualitative comparison on MASA.** Rows show fixed validation episodes at $C4$, $C5$, and $C6$. Circles and triangles mark starts and goals. Line styles distinguish three sampled roll-outs per method; crosses mark failed endpoints, and SR counts successful roll-outs.

Qualitative analysis. In Figure 6, GOMAA-Geo and GeoExplorer repeatedly pass through goal-adjacent cells without stepping onto the goal: all of their $C4$ roll-outs fail, and at $C5$ they succeed in one and two of three. AeroEVA follows the road network without detours at $C4$ and reaches the $C5$ goal in every roll-out. DINO-WM's three-step planner succeeds at $C4$ but struggles at $C5$, illustrating the limits of short-horizon planning. The $C6$ lake scene exposes a shared failure mode: homogeneous views provide few cues for crossing to the far shore, and AeroEVA succeeds in only one of three roll-outs. Actions are sampled to show behavioral variability; DINO-WM samples only its first action from the planner's per-action scores. Sampling details and further examples appear in Appendix A.5.

Table 2: **Ablating transition supervision, verifier evidence, and RL.** MASA validation success (%); Avg.: mean over $C3$–$C6$. IL: imitation learning. Bold: best in each block.

Stage	Method	$C3$	$C4$	$C5$	$C6$	Avg.
	Action SFT	19.8	23.5	30.7	59.0	33.3
	Joint action–transition SFT	**44.0**	**53.8**	**63.6**	**77.8**	**59.8**
IL	Visual latent + tool only	27.4	35.6	37.9	50.2	37.8
	Visual latent + text only	23.8	31.2	35.5	54.6	36.3
	Visual latent only	20.3	26.1	29.4	43.0	29.7
	GRPO (cold start)	31.6	42.8	55.8	78.7	52.2
RL	Action SFT + GRPO	27.6	47.1	60.6	90.4	56.4
	Our SFT + GRPO	**62.5**	**75.2**	**77.8**	**92.4**	**77.0**

Table 3: **Measuring how RL changes goal judgment and verifier use.** Acc./Bal.: per-turn / balanced accuracy of $\widehat{b}_t$. Alt./Foll.: defined in the text. Δ: + GRPO minus SFT (points).

Metric	SFT	+ GRPO	Δ
Acc.	67.2	82.4	+15.2
Bal.	78.4	85.7	+7.3
Alt.	70.9	74.7	+3.8
Foll.	47.4	22.7	−24.7

Goal judgment and verifier use. After GRPO, goal-adjacency accuracy improves and the policy follows misleading verifier scores less often when the goal is adjacent (Table 3). Alt. measures moves toward the goal when the top-scoring action does not; Foll. measures following that top score on such turns with the goal adjacent.

5.5 Temporal, Spatial, and Goal Generalization

We evaluate temporal, spatial, and goal generalization with the same MASA-trained checkpoint, without adaptation, following established protocols where available (Sarkar et al., 2024; Mi et al., 2025). Foresight is enabled only for aerial-image goals. Sampling details appear in Appendix A.2; additional per-distance results are in Appendix A.4.

Temporal generalization under appearance change. AeroEVA's latent prediction abstracts away pixel-level details to learn transferable predictive representations. On xBD-disaster (Gupta et al., 2019), the agent searches post-disaster imagery using pre-disaster goal views. The results (Table 4) support temporal generalization despite disaster-induced appearance changes.

Spatial transfer. Across SwissView100, MM-GAG, and xBD-pre, AeroEVA exceeds reproduced GOMAA-Geo by 17.0–19.6 points (Table 5). On finer MASA grids (Table 6), AeroEVA transferred from 5×5 reaches 60.9% over $C14$–$C18$, compared with 12.9% and 10.5% for GOMAA-Geo and GeoExplorer under the same transfer, and exceeds both methods trained directly on 10×10 grids at every distance. Appendix A.2 discusses the restricted long-distance geometries.

Goal generalization. Despite training only with aerial-image goals, AeroEVA reaches 73.4–74.8% with ground-photo goals and 69.6% with MM-GAG text goals, exceeding GOMAA-Geo by 19.0–19.7 and 9.2 points, respectively (Table 5).

Table 4: **Cross-time search on xBD-disaster.** Success (%) with post-disaster observations and pre-disaster goals.

Method	$C4$	$C5$	$C6$	$C7$	$C8$
GOMAA-Geo	37.0	47.4	67.3	75.8	**77.9**
GeoExplorer	40.9	41.0	50.2	49.5	38.8
DINO-WM	31.2	21.2	19.2	11.8	10.2
AeroEVA	**57.4**	**66.4**	**82.5**	**83.7**	73.2

Table 5: **Region and goal-modality transfer.** Mean success (%) over $C4$–$C8$.

Dataset	GOMAA-Geo	Geo Explorer	DINO-WM	AeroEVA (ours)
Aerial-image goals				
SwissView100	57.3	42.0	10.8	**75.2**
MM-GAG	61.9	40.7	27.3	**81.5**
xBD-pre	60.3	43.0	17.9	**77.3**
Ground-level-image goals				
MM-GAG	55.8	40.2	—	**74.8**
SwissViewMon.	53.7	37.6	—	**73.4**
Text goals				
MM-GAG	60.4	45.0	—	**69.6**

Table 6: **Transfer to** 10×10 **MASA grids.** Success (%); $B = 20$ for transfer. Retrained rows follow Mi et al. (2025).

Method	$C14$	$C15$	$C16$	$C17$	$C18$
Retrained: $10 \times 10 \rightarrow 10 \times 10$					
GOMAA-Geo	26.0	27.0	29.2	24.1	22.0
GeoExplorer	28.8	31.2	33.5	30.7	31.5
Transfer: $5 \times 5 \rightarrow 10 \times 10$					
GOMAA-Geo	24.4	18.8	15.8	4.7	0.9
GeoExplorer	22.9	19.7	8.5	1.2	0.0
DINO-WM	4.6	3.5	3.4	2.2	3.5
AeroEVA	**59.6**	**57.5**	**69.0**	**61.1**	**57.5**

Table 7: **Task-grounded search on AeroTask.** Success (%) with language goals and no tool evidence; Avg: $C3$–$C6$ mean.

Method	$C3$	$C4$	$C5$	$C6$	Avg
GOMAA-Geo	26.9	35.8	45.6	67.0	43.8
GeoExplorer	**32.1**	39.2	41.9	50.7	41.0
AeroEVA	29.1	**51.6**	**57.8**	**83.0**	**55.4**

5.6 AeroTask: Toward More Demanding Goal Reasoning

Task-grounded goals. AeroTask asks agents to infer suitable locations from operational requirements, such as water access for a fire engine or a helicopter landing site. It contains 696 tasks over 651 overhead-image windows, spanning 13 application categories and eight target land-cover families. Imagery comes primarily from Chesapeake NAIP surveys, supplemented by MM-GAG (Sarkar et al., 2024) and SwissView (Mi et al., 2025); evaluation uses the same 5×5, $B = 10$ setting.

Agentic dataset construction. Our agentic workflow combines geospatial products, external visual tools, and VLM generation and verification. In the main NAIP branch, expert knowledge encoded in land-cover mappings and sampling rules guides screening with NLCD maps (Dewitz, 2023), avoiding costly segmentation and VLM inference on rejected regions. Because map labels can differ from observable image content, SegEarth-OV (Li et al., 2025) measures target coverage on the acquired imagery, and Qwen3-VL-32B-Instruct-AWQ (Bai et al., 2025) confirms visibility. The same VLM generates purpose instructions and iteratively refines them through rule-based and visual checks, retaining tasks that pass blind localization and selected-rival ambiguity checks. Figure 7 in Appendix A.3 summarizes the construction workflow, application distribution, and text-goal record counts relative to MM-GAG.

Evaluation and results. With task-hard language goals and the tool off, AeroEVA reaches 55.4% Avg over $C3$–$C6$, above GOMAA-Geo (43.8%) and GeoExplorer (41.0%; Table 7). It leads at $C4$–$C6$, while GeoExplorer leads at $C3$. Appendices A.2–A.3 give sampling, acceptance checks, and generator and judge settings; Table 12 reports scenario results.

6 Conclusion

We introduced AeroEVA, a VLM agent for embodied aerial search. Three findings stand out. First, the nearest patches of the next view are the most action-sensitive prediction target, and joint supervision keeps action information decodable at the final decoder layer, unlike action SFT. Second, visual prediction alone does not improve search; it pays off when combined with textual transition targets and verifier evidence. Third, RL improves goal-adjacency judgment and makes the policy follow misleading verifier scores less often. Without retraining, the policy transfers across regions, goal modalities including AeroTask's task-grounded language goals, appearance changes, and finer grids. Together, these results support unifying world modeling and VLM policy learning: a single spatial predictor can serve as both a training signal for the policy and evidence for its decisions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing parts of the method and data-processing pipeline, interpreting experimental results, and preparing the manuscript. We also used these tools for feedback on methodology and experimental design. For AeroTask, we used Qwen3-VL-32B-Instruct-AWQ to generate, refine, and screen synthetic task instructions; the same model served as both generator and judge, as described in Appendix A.3. Additional uses included writing and debugging code, formatting tables, and revising text for clarity and conciseness. We reviewed all AI-assisted work. AI-assisted implementations were run and tested by the authors, and suggested interpretations were checked against the experimental results. The authors take responsibility for the final content of this work, including its text, claims, code, and other artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

The model architecture, visual prediction and textual transition objectives, and policy learning procedure are described in Section 4. The construction of AeroTask, including scenario definitions, is detailed in Section 5.6 and Appendix A.3. Evaluation protocols, including the observation and action spaces, search budgets, and success criteria, are specified in Section 5. Additional implementation details, training hyperparameters, and predictor-tool configurations are provided in Appendix C. Code, model weights, and the AeroTask dataset will be released upon acceptance.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801. IEEE, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Shiqi Chen, Tongyao Zhu, Zian Wang, Jinghan Zhang, Kangrui Wang, Siyang Gao, Teng Xiao, Yee Whye Teh, Junxian He, and Manling Li. Internalizing world models via self-play finetuning for agentic rl. *arXiv preprint arXiv:2510.15047*, 2025.

Jon Dewitz. National Land Cover Database (NLCD) 2021 products. U.S. Geological Survey data release, 2023. URL https://doi.org/10.5066/P9JZ7AO3.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Li Zhen, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Yiwen Tang, et al. Openfly: A comprehensive platform for aerial vision-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 101858–101871, 2026.

Renjie Gu, Kaichen Zhou, Yan Luo, and Mengyu Wang. Geoworld-vlm: Geometry from world models for vision-language models. *arXiv preprint arXiv:2605.16713*, 2026.

Ritwik Gupta, Richard Hosfelt, Sandra Sajeev, Nirav Patel, Bryce Goodman, Jigar Doshi, Eric Heim, Howie Choset, and Matthew Gaston. xbd: A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019.

Yatai Ji, Zhengqiu Zhu, Yong Zhao, Beidan Liu, Chen Gao, Yihao Zhao, Sihang Qiu, Yue Hu, and Quanjun Yin. Towards autonomous uav visual object search in city space: Benchmark and agentic methodology. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18342–18350, 2026.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. Citynav: A large-scale dataset for real-world aerial navigation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5912–5922. IEEE, 2025.

Cong Li, Peixi Peng, Yisen Zhao, Xinyu Hu, Shudong Liu, Zhan Su, and Zhuojian Li. Tapo: Transition-aware policy optimization for llm agents. *arXiv preprint arXiv:2607.27973*, 2026a.

Haoxi Li, Qinglin Hou, Jianfei Ma, Jinxiang Lai, Tao Han, Sikai Bai, Jingcai Guo, Jie Zhang, and Song Guo. What you think is what you see: Driving exploration in vlm agents via visual-linguistic curiosity. *arXiv preprint arXiv:2605.03782*, 2026b.

Kaiyu Li, Ruixun Liu, Xiangyong Cao, Xueru Bai, Feng Zhou, Deyu Meng, and Zhi Wang. Segearth-ov: Towards training-free open-vocabulary segmentation for remote sensing images. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10545–10556. IEEE, 2025.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15338–15348. IEEE, 2023.

Youwei Liu, Jian Wang, Hanlin Wang, and Wenjie Li. Comap: Co-evolving world models and agent policies for llm agents. *arXiv preprint arXiv:2606.02372*, 2026.

Ning Lu, Baijiong Lin, Shengcai Liu, Jiahao Wu, Haoze Lv, Yanbin Wei, Lingting Zhu, Shengju Qian, Xin Wang, Ying-Cong Chen, et al. Policy and world modeling co-training for language agents. *arXiv preprint arXiv:2606.02388*, 2026.

Li Mi, Manon Béchaz, Zeming Chen, Antoine Bosselut, and Devis Tuia. Geoexplorer: Active geo-localization with curiosity-driven exploration. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6122–6131. IEEE, 2025.

OpenAI. Introducing GPT-5.4 mini and nano, March 2026. URL `https://openai.com/index/introducing-gpt-5-4-mini-and-nano/`.

Aleksis Pirinen, Anton Samuelsson, John Backsund, and Kalle Åström. Aerial view localization with reinforcement learning: Towards emulating search-and-rescue. *arXiv preprint arXiv:2209.03694*, 2022.

Cheng Qian, Emre Can Acikgoz, Bingxuan Li, Xiusi Chen, Yuji Zhang, Bingxiang He, Qinyu Luo, Gokhan Tur, Dilek Hakkani-Tur, Yunzhu Li, et al. Current agents fail to leverage world model as tool for foresight. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13686–13723, 2026.

Anindya Sarkar, Srikumar Sastry, Aleksis Pirinen, Chongjie Zhang, Nathan Jacobs, and Yevgeniy Vorobeychik. Gomaa-geo: Goal modality agnostic active geo-localization. *Advances in Neural Information Processing Systems*, 37:104934–104964, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

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

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Zhenguo Sun, Yu Sun, Hande Huang, and Alois Knoll. ω-eva: Envision, verify, and act with latent interactive world models. *arXiv preprint arXiv:2606.09457*, 2026.

Kangrui Wang*, Pingyue Zhang*, Zihan Wang*, Yaning Gao*, Linjie Li*, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://vagen-ai.github.io/`.

Zhitong Wang, Songze Li, Hao Peng, Shuzheng Si, Yi Wang, Maosong Sun, and Juanzi Li. Envrl: Learn from environment dynamics in agentic reinforcement learning. *arXiv preprint arXiv:2606.17680*, 2026.

Yilin Wu, Ran Tian, Gokul Swamy, and Andrea Bajcsy. From foresight to forethought: Vlm-in-the-loop policy steering via latent alignment. *arXiv preprint arXiv:2502.01828*, 2025.

Jianqiang Xiao, Yuexuan Sun, Yixin Shao, Boxi Gan, Rongqiang Liu, Yanjin Wu, Weili Guan, and Xiang Deng. Uav-on: A benchmark for open-world object goal navigation with aerial agents. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 13023–13029, 2025.

Lingfeng Zhang, Zeying Gong, Xiaoshuai Hao, Haoxiang Fu, Qiang Zhang, Mingliang Zhou, Hangjun Ye, Xiaojun Liang, Junwei Liang, and Wenbo Ding. Futurenav: Unified world-action modeling for vision-and-language navigation. *arXiv preprint arXiv:2606.30367*, 2026a.

Wanyue Zhang, Wenxiang Wu, Wang Xu, Jiaxin Luo, Helu Zhi, Yibin Huang, Shuo Ren, Zitao Liu, and Jiajun Zhang. World2vlm: Distilling world model imagination into vlms for dynamic spatial reasoning. *arXiv preprint arXiv:2604.26934*, 2026b.

Baining Zhao, Jiacheng Xu, Weicheng Feng, Xin Zhang, Zhaolu Wang, Haoyang Wang, Shilong Ji, Ziyou Wang, Jianjie Fang, Zhiheng Zheng, et al. Worldvln: Autoregressive world action model for aerial vision-language navigation. *arXiv preprint arXiv:2605.15964*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: world models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 79115–79135, 2025.

Chenming Zhu, Jingli Lin, Yilin Long, Peizhou Cao, Tai Wang, Jiangmiao Pang, and Xihui Liu. Thinking with imagination: Agentic visual spatial reasoning with world simulators. *arXiv preprint arXiv:2606.06476*, 2026.

A Additional Results

A.1 Baseline Implementations and Result Sources

Unless otherwise indicated, baseline results throughout the main text and appendix use our reimplementations trained on MASA with $C3$–$C6$ initializations. Tables 1 and 6 explicitly distinguish retrained runs from published results or released checkpoints.

Architectures and checkpoints. GOMAA-Geo and GeoExplorer combine Sat2Cap features with Falcon-7B actor–critic policies (Sarkar et al., 2024; Mi et al., 2025). We use our retrained checkpoints for all transfer experiments. On MASA, we also evaluate the released GOMAA-Geo checkpoint and our reproduction of GeoExplorer under its original training protocol. DINO-WM (Zhou et al., 2025) learns dynamics over frozen DINOv3 patch features on the same scenes and plans exhaustively over three steps. GeoExplorer-Qwen reimplements GeoExplorer's method with Qwen3-VL-4B (Bai et al., 2025), following the original paper. AeroEVA (SFT) jointly learns actions and visual/textual transitions; AeroEVA (SFT + GRPO) adds RL. Both use prediction-tool evidence during search when available; training details are in Appendices C.3 and C.4.

Published MASA results. Rows marked $*$ in Table 1 report Random, PPO (Schulman et al., 2017), AiRLoc (Pirinen et al., 2022), and Decision Transformer (DiT) (Chen et al., 2021) from GeoExplorer's published MASA test comparison, which attributes these baselines to GOMAA-Geo; we recompute averages from the per-distance values. These scores use the same test scenes with different sampled episodes. All unmarked rows report our MASA test evaluations, including GOMAA-Geo and GeoExplorer under both the original and $C3$–$C6$ training protocols. The zero-shot block includes Qwen3-VL-4B and GPT-5.4-mini.

A.2 Evaluation Protocols and Distance Coverage

MASA splits and episode sampling. We follow the distance-stratified evaluation of GOMAA-Geo and GeoExplorer (Sarkar et al., 2024; Mi et al., 2025) on non-overlapping 5×5 grids, with four cardinal actions and a ten-move budget. The training pool comprises 831 scenes, disjoint from the 178 validation and 179 test scenes. Training episodes are initialized with $C \sim \mathcal{U}\{3, \ldots, 6\}$; this restriction applies to initial distance, not every intermediate state visited in a rollout. We select policy checkpoints on in-distribution validation episodes. Thus, $C7$–$C8$ initializations are held out from training and checkpoint selection. Our $C4$–$C8$ validation analyses use five start–goal pairs per scene at each distance: 890 episodes per distance and 4,450 in total. Our trained-policy evaluations in Table 1 use 4,475 fixed test episodes, with five pairs per scene at each of the same distances.

Geometric coverage. A 5×5 grid admits 136, 120, 80, 40, 16, and four ordered start–goal pairs at $C3$–$C8$, respectively. The limited geometric coverage at $C7$–$C8$ motivates reporting the $C4$–$C6$ mean alongside the full-range Avg.

Transfer with aerial-image goals. All policies evaluated in our transfer runs are trained on MASA with aerial-image goals and evaluated without adaptation. Unless otherwise stated, transfer uses 5×5 grids and $B = 10$. SwissView100 contains 100 scenes and 500 episodes per distance; MM-GAG contains 65 areas and 325 episodes per distance on stitched aerial grids; and the xBD test set contains 800 areas and 4,000 episodes per distance. The xBD pre-disaster setting uses pre-disaster observations and goals; xBD-disaster (cross-time search) pairs post-disaster observations with a pre-disaster goal. AeroEVA uses the same checkpoint throughout, with foresight evidence for aerial-image goals and the tool off for ground-level-image and text goals. Table 9 gives per-distance aerial-image results. The cross-time comparison evaluates the full policy; it does not isolate robustness gains from latent transition prediction or tool use.

MM-GAG cross-modal goals. Following GOMAA-Geo (Sarkar et al., 2024), we extract a 5×5 window of individual 256×256-pixel tiles from each area's 11×11 tile grid, placing the tile associated with the ground photograph at the sampled goal cell. This tile is central in the original grid, but its position varies across episode windows. The start–goal pairs match the aerial-image evaluation. AeroEVA receives the photograph in place of the aerial goal patch; GOMAA-Geo and GeoExplorer

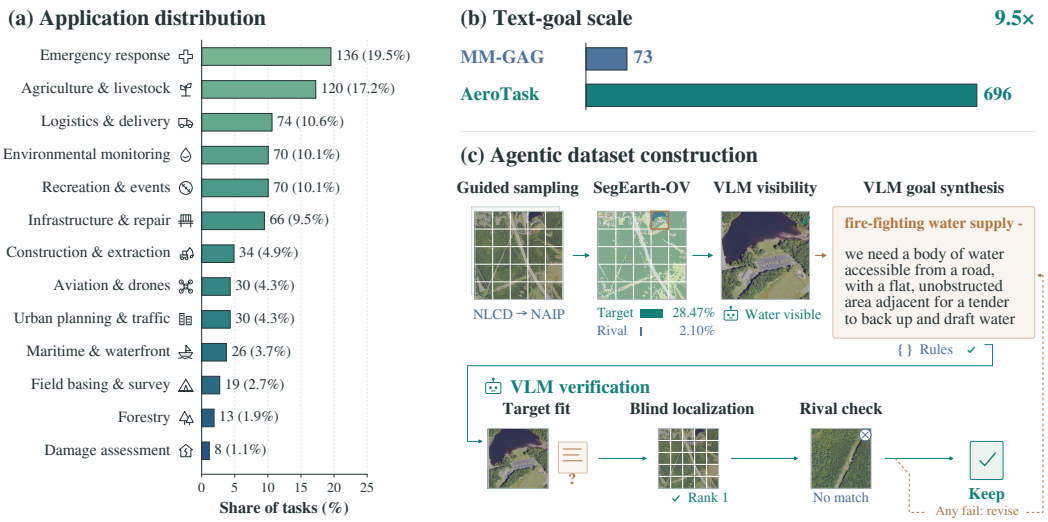


Figure 7: **AeroTask spans 13 applications with 696 task-grounded goals.** (a,b) The full release contains 9.5× as many text-goal records as reported for MM-GAG (73) (Sarkar et al., 2024). (c) Geospatial sampling, segmentation, and VLM verification connect operational requirements to visible target features, illustrated by firefighting water supply.

receive its CLIP image embedding in place of the goal's Sat2Cap embedding. Text-goal evaluation uses the same window construction with the caption replacing the photograph. Table 10 also reports aerial-image goals under this window construction.

SwissViewMonuments. Following GeoExplorer (Mi et al., 2025), we use 15 landmarks, each with one ground-level photograph and 25 shifted 1500×1500-pixel aerial views that place the landmark in every grid cell. The landmark's cell is the goal. One start is sampled per view at each distance; views whose goal admits no start at that distance are excluded, leaving 375, 360, 300, 180, and 60 episodes for $C4$–$C8$. The goal interfaces are the same as for MM-GAG ground-level photographs. Ground photographs are rotated according to EXIF orientation and downscaled to at most 1,008 pixels on the long side; GeoExplorer's pipeline passes the original files to CLIP. Table 11 reports the per-distance results.

Larger MASA grids. For Table 6, we re-tile MASA validation scenes into 10×10 grids with 150-pixel cells and evaluate 890 fixed episodes per distance at $C14$–$C18$, with $B = 20$. These distances admit 140, 80, 40, 16, and four ordered start–goal configurations, respectively; AeroEVA's $C14$–$C16$ mean is 62.0%. The published retrained rows come from GeoExplorer's supplementary Table S6 (Mi et al., 2025); our budget and episode specification apply only to the transfer rows.

AeroTask. The main NAIP acquisition branch uses $1500 \times 1500\,\mathrm{m}$ windows at $1\,\mathrm{m}$ resolution, yielding $300\,\mathrm{m}$ cells. Evaluation uses 4,136 instances with equally represented start–goal geometries across the $C3$–$C6$ range, task-hard language goals, and AeroEVA's tool off. Each instance uses a 5×5 grid and $B = 10$. The reported Avg is the mean of the four per-distance success rates. Table 12 additionally breaks down application scenarios.

A.3 AeroTask Construction Details

Counting conventions. The 696-task release contains 428 training and 268 evaluation records. Figure 7 uses the reported MM-GAG count of 73; the available caption release contains 65 records. Counts refer to goal records rather than resampled navigation instances. The figure groups four structural and four storm-related timber assessments under damage assessment based on instruction review; Table 12 uses the original released category labels.

Scene proposal. The agentic workflow combines geospatial products, an external segmentation tool, and VLM reasoning. In the main NAIP acquisition branch, expert knowledge is encoded as manually specified mappings from NLCD land-cover classes (Dewitz, 2023) to target families and class-specific sampling rules. For example, forest classes guide tree-cover sampling, while medium- and high-intensity developed classes propose building scenes. The sampler uses coverage floors and relative dominance to identify candidate windows; developed-area fractions additionally support sampling built-up scenes. These inexpensive operations on cached, $30\,\mathrm{m}$ land-cover maps precede high-resolution image acquisition, segmentation, and VLM inference, avoiding those downstream costs for rejected regions.

Target verification. Map resolution and acquisition date can differ from those of the search imagery, so map labels serve as proposal priors rather than target annotations. Targets are instead derived from the acquired imagery using SegEarth-OV (Li et al., 2025), an external open-vocabulary segmentation tool with a fixed nine-class vocabulary. A candidate's target-class coverage must meet a class-specific floor and exceed twice the coverage of every competing cell. Qwen3-VL-32B-Instruct-AWQ then confirms that the target class is visible in the imagery used for search.

Instruction generation and filtering. For each retained target, the VLM generates a purpose instruction from a task domain associated with its land-cover class. Programmatic rules enforce text constraints; VLM checks assess target fit, blind localization among all 25 cells, and ambiguity against selected rival cells. This feedback guides instruction revision and filtering. The released subset retains tasks whose purpose instruction ranks the intended target first and passes the selected-rival ambiguity check. Qwen3-VL-32B-Instruct-AWQ both generates and judges the texts, so these model-based quality criteria do not constitute independent-model validation.

Aligned instruction variants. Each task retains *natural*, *task*, and *task-hard* texts: an appearance description, a job with an appearance description, and a job with site requirements, respectively. Evaluation uses *task-hard*, which asks the searcher to infer a suitable place from its purpose and requirements. The quantitative *spec* records the segmentation-derived predicate for analysis and is never provided as an agent instruction.

Task examples. Figure 8 expands the application examples in Figure 1 with nine tasks from the released corpus, including emergency staging, delivery, field basing, land reclamation, event planning, crop treatment, water sampling, damage assessment, and parking surveys. Each example shows the complete *task-hard* instruction together with the corresponding scene and target crop.

A.4 Additional Navigation Results

Unless marked as published references, the navigation tables below use single checkpoints and frozen evaluation episodes, with 890 episodes per distance for MASA validation. Avg covers $C4$–$C8$ unless otherwise specified; AeroTask uses $C3$–$C6$. The $C4$–$C6$ mean excludes the two held-out initial-distance buckets with the least geometric coverage.

Table 8 repeats the MASA validation evaluation with sampled actions. AeroEVA samples at temperature 1.0 with top-p 1; GOMAA-Geo and GeoExplorer draw each move from their wall-masked action distributions. All runs use the same 4,450 frozen episodes, so the spread reflects decoding alone. Sampling lowers Avg by 1.4 points for AeroEVA and by 2.8 and 1.7 points for GOMAA-Geo and GeoExplorer (differences of unrounded means), and the standard deviation of Avg across runs is at most 0.6 points. Paired per episode over $C4$–$C8$, AeroEVA with the tool exceeds GOMAA-Geo by 21.6 ± 0.6 points and GeoExplorer by 33.6 ± 0.6 points under sampling (mean $\pm$ standard error over episodes; sampled success counted as the frequency over the four runs), compared with 20.3 ± 0.8 and 33.2 ± 0.9 under greedy decoding. The Qwen3-VL GeoExplorer and DINO-WM are not included in this evaluation.

Table 9 expands Tables 5 and 4 under the protocols in Appendix A.2. The published MM-GAG rows ($*$) come from Table 2 of GeoExplorer (Mi et al., 2025), which attributes them to GOMAA-Geo (Sarkar et al., 2024); they use the source's episodes, with percentages and averages computed from its reported scores. The released GOMAA-Geo checkpoint and RL policy without spatial prediction are evaluated on SwissView100 only.

(a) Emergency response



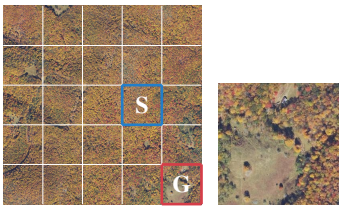
paved area large enough for a fleet to assemble with direct road access for vehicles to enter and exit safely and adjacent to a large building with parking spaces

(b) Logistics and delivery



deliver packages to a building - we need a paved area for vehicle access and a building frontage for drop-off

(c) Field basing and survey



temporary camp - we need dry level grass, vehicle access

(d) Construction and extraction



land reclamation planning - disturbed ground to be brought back - we need exposed bare earth with visible grading and access tracks, a paved surface with parking and buildings nearby

(e) Recreation and events



crowd-management planning - we need a paved surface, a baseball field adjacent to it

(f) Agriculture and livestock



crop treatment - we need open cultivated land, access path along the edge

(g) Environmental monitoring



collect water samples - we need open water accessible by land, away from any discharge point

(h) Damage assessment



assess structural damage to buildings - we need accessible buildings, street-level approach, no obstructions, and presence of multiple buildings in close proximity

(i) Urban planning and traffic



count parking bay usage - we need paved lot with marked spaces, accessible from road, surrounded by buildings or trees

Figure 8: **AeroTask task examples.** Nine tasks illustrate application-oriented language goals. Each panel pairs the complete *task-hard* instruction with its search scene (left) and target-cell crop (right). Blue S and red G mark the recorded start and goal in the 5×5 grid. Full scenes, location markers, and target crops are shown for the reader; the agent receives local observations and the language goal. The damage-assessment instruction specifies a task, not a confirmed damage label.

Table 8: **Testing sensitivity to greedy versus sampled decoding.** MASA validation success (%); sampled rows report mean ± standard deviation over four runs on the same episodes.

Method	Decoding	$C4$	$C5$	$C6$	$C7$	$C8$	Avg	$C4$–6
AeroEVA	greedy	75.2	77.8	92.4	90.9	88.3	84.9	81.8
	sampled	73.8 ± 0.7	75.4 ± 1.1	91.5 ± 1.5	89.6 ± 0.7	87.3 ± 0.4	83.5 ± 0.5	80.2 ± 0.7
GOMAA-Geo	greedy	40.6	49.6	73.5	80.6	79.0	64.6	54.5
	sampled	38.6 ± 1.0	47.6 ± 1.3	69.2 ± 1.4	76.4 ± 0.9	77.5 ± 1.1	61.9 ± 0.6	51.8 ± 0.7
GeoExplorer	greedy	41.6	46.2	54.3	61.6	54.7	51.7	47.3
	sampled	41.5 ± 1.7	45.4 ± 1.1	54.8 ± 1.1	58.6 ± 0.6	49.4 ± 1.1	49.9 ± 0.4	47.2 ± 0.4

Table 9: **Testing geographic and temporal transfer with aerial-image goals.** Per-distance success (%) with aerial-image goals. $*$: published result.

Protocol	Method	$C4$	$C5$	$C6$	$C7$	$C8$	Avg	$C4$–6
SwissView100	GOMAA-Geo (released)	42.0	50.2	67.8	74.6	68.8	60.7	53.3
	GOMAA-Geo	38.4	46.0	64.6	74.0	63.4	57.3	49.7
	GeoExplorer	41.0	41.4	47.6	44.4	35.6	42.0	43.3
	DINO-WM	20.6	14.2	10.2	4.0	5.2	10.8	15.0
	RL w/o spatial prediction	47.0	56.2	86.6	100.0	100.0	78.0	63.3
	AeroEVA	60.4	70.0	83.8	86.6	75.4	75.2	71.4
MM-GAG aerial	Random*	14.12	5.84	6.40	2.47	2.36	6.24	8.79
	PPO*	14.89	18.54	18.79	21.76	24.32	19.66	17.41
	AiRLoc*	17.45	16.89	20.19	21.56	22.90	19.80	18.18
	DiT*	20.23	28.56	35.16	41.90	44.23	34.02	27.98
	GOMAA-Geo	39.7	46.2	68.0	76.6	79.1	61.9	51.3
	GeoExplorer	40.9	39.7	45.8	45.2	32.0	40.7	42.2
	DINO-WM	41.5	32.3	28.6	14.5	19.7	27.3	34.2
	AeroEVA	66.8	76.0	88.3	88.6	88.0	81.5	77.0
xBD pre-disaster	GOMAA-Geo	38.4	46.9	67.6	75.1	73.5	60.3	51.0
	GeoExplorer	39.5	41.4	49.9	47.2	37.0	43.0	43.6
	DINO-WM	30.1	20.2	17.5	11.5	10.2	17.9	22.6
	AeroEVA	63.6	70.0	86.0	88.5	78.4	77.3	73.2
xBD-disaster	GOMAA-Geo	37.0	47.4	67.3	75.8	77.9	61.1	50.6
	GeoExplorer	40.9	41.0	50.2	49.5	38.8	44.1	44.1
	DINO-WM	31.2	21.2	19.2	11.8	10.2	18.7	23.9
	AeroEVA	57.4	66.4	82.5	83.7	73.2	72.6	68.8

Table 10 uses the window construction in Appendix A.2; its aerial-goal results are not directly comparable with the stitched protocol in Table 9. For GOMAA-Geo and GeoExplorer, fixing the goal descriptor changes ground-view and text-goal success by at most 3 points, indicating largely goal-independent search rather than established use of the descriptor. These controls do not cover the full AeroEVA checkpoint.

Table 11 expands the SwissViewMonuments column of Table 5 under Appendix A.2. Views of each landmark overlap and share a goal photograph, leaving 15 independent areas. Published rows ($*$) follow the same sampling rule but use different episodes; their averages are computed from the reported per-distance scores.

A.5 Additional Qualitative Examples

Figures 9 and 10 extend Figure 6 with further frozen validation episodes, drawn with the same conventions. Roll-outs are sampled rather than greedy so that the spread of each policy is vis-

Table 10: **Testing goal-modality transfer on MM-GAG.** Success (%); Avg: $C4$–$C8$ mean. Aerial/Ground/Text denote aerial-image, ground-view, and text goals. All rows use matched window episodes; the tool is off for Ground/Text.

Goal	Method	$C4$	$C5$	$C6$	$C7$	$C8$	Avg	$C4$–$C6$
Aerial	GOMAA-Geo	40.9	41.5	70.8	74.8	74.2	60.4	51.1
	GeoExplorer	40.0	41.5	48.6	48.0	42.2	44.1	43.4
	AeroEVA	72.6	74.8	92.9	94.5	92.0	85.4	80.1
Ground	GOMAA-Geo	37.8	43.7	62.5	68.9	66.2	55.8	48.0
	GeoExplorer	36.6	44.3	43.7	44.3	32.0	40.2	41.5
	AeroEVA	50.5	59.7	84.0	91.4	88.6	74.8	64.7
Text	GOMAA-Geo	40.3	43.1	67.4	80.0	71.4	60.4	50.3
	GeoExplorer	38.5	47.7	51.1	48.9	39.1	45.0	45.7
	AeroEVA	52.0	61.2	84.3	84.9	65.5	69.6	65.8

Table 11: **Testing cross-view transfer with ground-photo goals.** SwissViewMonuments success (%); 375/360/300/180/60 episodes for $C4$–$C8$. $*$: published in Table 4 of GeoExplorer (Mi et al., 2025).

Method	$C4$	$C5$	$C6$	$C7$	$C8$	Avg	$C4$–6
GOMAA-Geo*	41.1	38.3	63.7	77.2	75.0	59.1	47.7
GeoExplorer*	41.6	51.7	77.3	87.8	78.3	67.3	56.9
GOMAA-Geo	39.5	42.8	63.0	65.0	58.3	53.7	48.4
GeoExplorer	35.7	36.4	43.0	42.8	30.0	37.6	38.4
AeroEVA	**48.0**	**57.8**	**85.3**	**94.4**	**81.7**	**73.4**	**63.7**

ible: AeroEVA decodes at temperature 1.0, GOMAA-Geo and GeoExplorer sample from their wall-masked action distributions, and DINO-WM draws its first action from a softmax over the standardized per-action scores of the three-step exhaustive planner (temperature 0.5); the deterministic planner used in the tables is the arg-min of the same scores. Three roll-outs are shown per model; for GOMAA-Geo and GeoExplorer the SR over all four sampled roll-outs is reported in the accompanying index.

Cross-dataset examples. Figures 11–14 repeat the protocol on the benchmarks of Sections 5.5 and 5.6, with the same sampled roll-outs (three per model). DINO-WM is shown only where its world model applies, i.e. on aerial image goals; the ground-photo and text-goal protocols compare AeroEVA with the two reactive policies.

Table 12 reports success by application scenario on the 4,136-instance AeroTask set, pooled over $C3$–$C6$, with language goals and the tool off. GeoExplorer uses its 449k checkpoint; the 1M checkpoint scores 37.0 Avg.

B Inverse-Dynamics Probe

B.1 Probe Protocol

We use inverse dynamics to probe representations of environment dynamics, recovering the navigation action from current and subsequent observations. Figure 4 uses the same inverse-dynamics readout architecture for all features: subtraction of the training-split mean token, per-token L2 normalization, a shared 64-dimensional linear projection without spatial pooling, and an MLP on $[z_t, z_{t+1}, z_{t+1} - z_t]$ for each ordered adjacent pair, where z_t and z_{t+1} are the flattened projected token features of the current and next observations, respectively. We fit the readout on the 831 MASA training scenes, reserving every tenth scene for early stopping, and evaluate on the 178 held-out scenes. The curves compare the released Qwen3-VL-4B with two supervised configurations: Action SFT, trained with the action-prediction loss only, and Joint action–transition SFT (full), which



Figure 9: **Additional successes across scene types.** Rows: suburban street grid ($C4$), dense residential blocks ($C5$), a river bend with a riverside park ($C6$), a wooded suburban scene ($C6$), and a golf course between woods ($C5$). AeroEVA reaches the goal in every roll-out but one. The reactive policies pass through goal-adjacent cells without stopping in the first two rows; in the third row, where the goal sits on the corner of the park, GOMAA-Geo and GeoExplorer each succeed in two of three roll-outs by following the border. DINO-WM is reliable only when the goal is a few steps away.

adds $\mathcal{L}_{\text{vis}}$, goal-relative and next-view textual supervision and the prediction tool, and initializes GRPO. Both SFT checkpoints are trained for the same 1,125 updates on the same data with the same schedule and the same trainable parameters; the configurations differ in supervision and tool use. For context, frozen Sat2Cap and DINOv3 achieve 35.7% and 88.9%, respectively, under the same probe. Scene-level bootstrap 95% intervals are within ± 1.7 percentage points for every estimate.



Figure 10: **Failure cases and long-horizon planning.** *Row 1* ($C5$): repetitive salt-marsh texture; the start and goal views are nearly indistinguishable, and only one AeroEVA roll-out succeeds. *Row 2* ($C4$): an industrial yard whose goal is a parking lot next to storage tanks; every method but one AeroEVA roll-out fails. *Rows 3–4* ($C8$, corner to corner): DINO-WM's three-step lookahead never leaves the start quadrant and oscillates among a few cells, while AeroEVA and the reactive policies cover the distance; in the last row the reactive policies succeed by walking the border, a geometry-specific shortcut at $C8$.

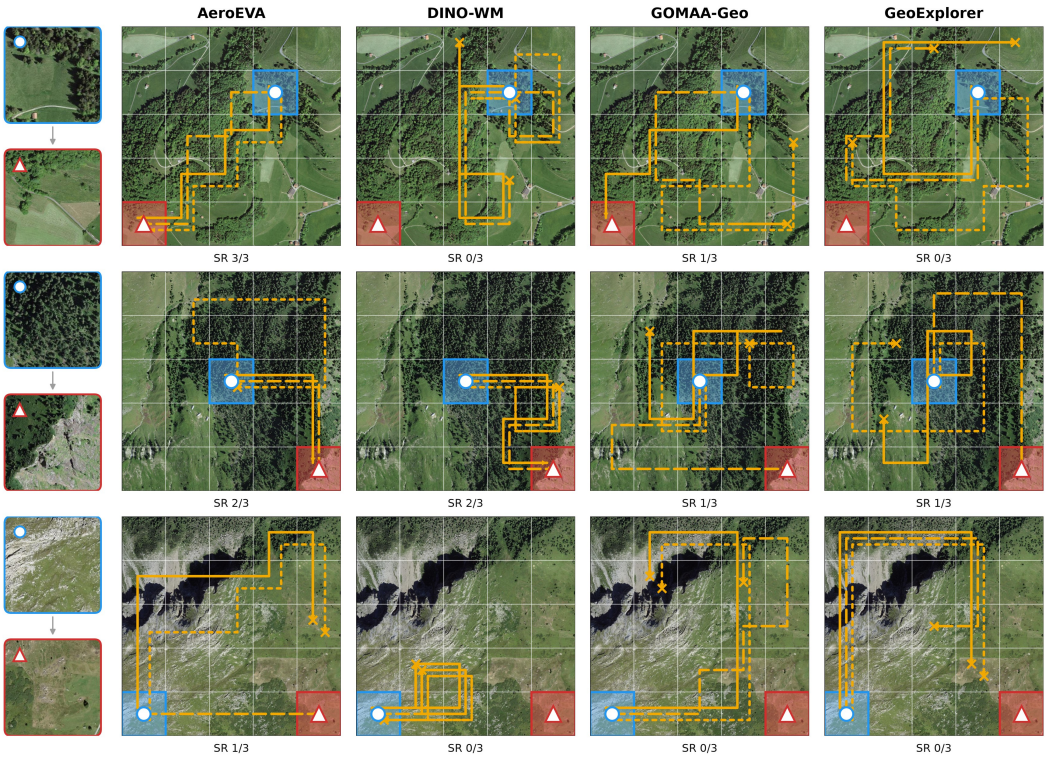


Figure 11: **SwissView100, zero-shot transfer.** Alpine scenes unseen in training. *Row 1* ($C6$, meadows and farm tracks): AeroEVA descends diagonally to the goal in every roll-out; DINO-WM circles near the start, GOMAA-Geo reaches the goal once, GeoExplorer sweeps the border. *Row 2* ($C4$, forest with a rock outcrop as goal): DINO-WM and AeroEVA each succeed in two of three roll-outs; the reactive policies succeed once each. *Row 3* ($C4$, scree slope): start and goal share the same rock texture along the bottom edge; only one AeroEVA roll-out reaches the goal.

Table 12: **Evaluating task-grounded search across application scenarios.** AeroTask success (%) pooled over $C3$–$C6$.

Application scenario	n	GOMAA-Geo	GeoExplorer	AeroEVA
Emergency response	887	37.8	37.4	**51.2**
Agriculture and livestock	668	41.8	41.3	**47.6**
Logistics and delivery	499	36.9	36.1	**44.5**
Recreation and events	443	28.4	33.9	**47.2**
Environmental monitoring	373	45.6	44.0	**48.5**
Infrastructure inspection and repair	332	38.9	38.6	**45.8**
Construction and extraction	219	34.2	38.8	**46.1**
Urban planning and traffic	208	44.2	38.5	**49.0**
Aviation and drone operations	160	35.6	38.8	**45.0**
Maritime and waterfront	127	26.8	29.1	**45.7**
Field basing and survey	118	44.1	**50.8**	**50.8**
Forestry	88	42.0	31.8	**58.0**
Damage assessment	14	–	–	–

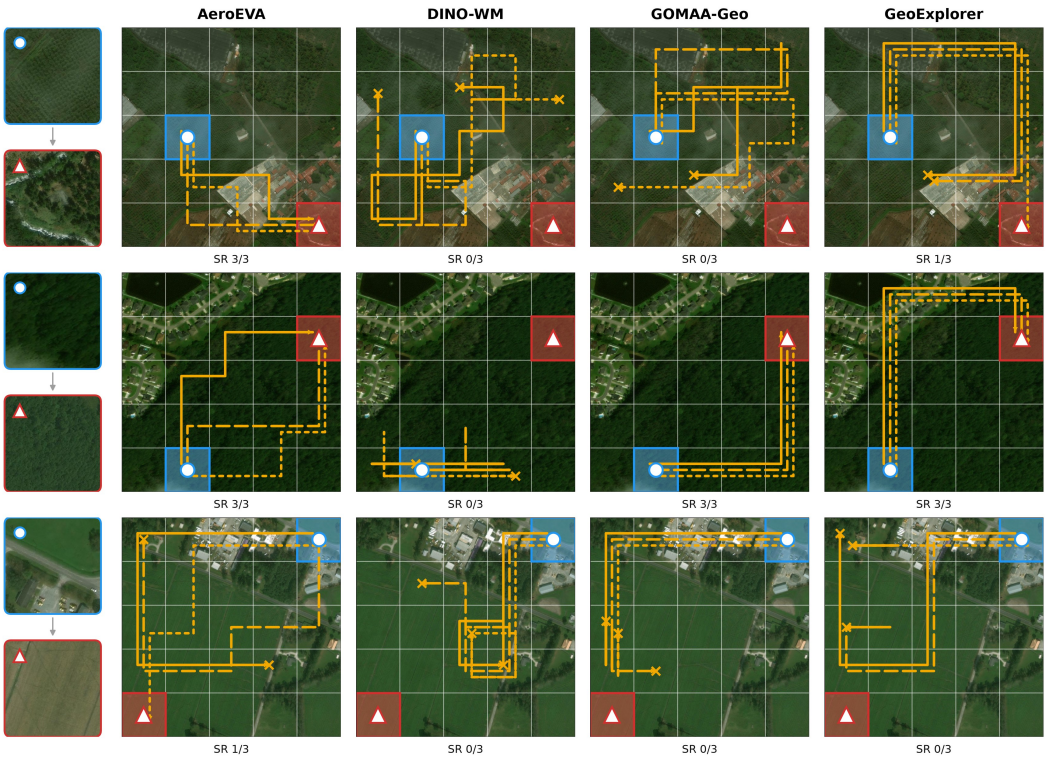


Figure 12: **xBD cross-time.** The agent searches post-disaster imagery for a goal given as its pre-disaster view (red frame). *Row 1* ($C5$): the goal region and the damaged buildings on the way look different from the goal image, yet AeroEVA reaches it in all roll-outs; GeoExplorer succeeds once, DINO-WM and GOMAA-Geo not at all. *Row 2* ($C6$, forest, right-edge goal): the reactive policies reach the goal by following the border, AeroEVA takes interior routes, and DINO-WM remains near the bottom of the scene. *Row 3* ($C8$, corner to corner): the cultivated goal field has to be found across farmland and warehouses; one AeroEVA roll-out succeeds, every baseline fails.

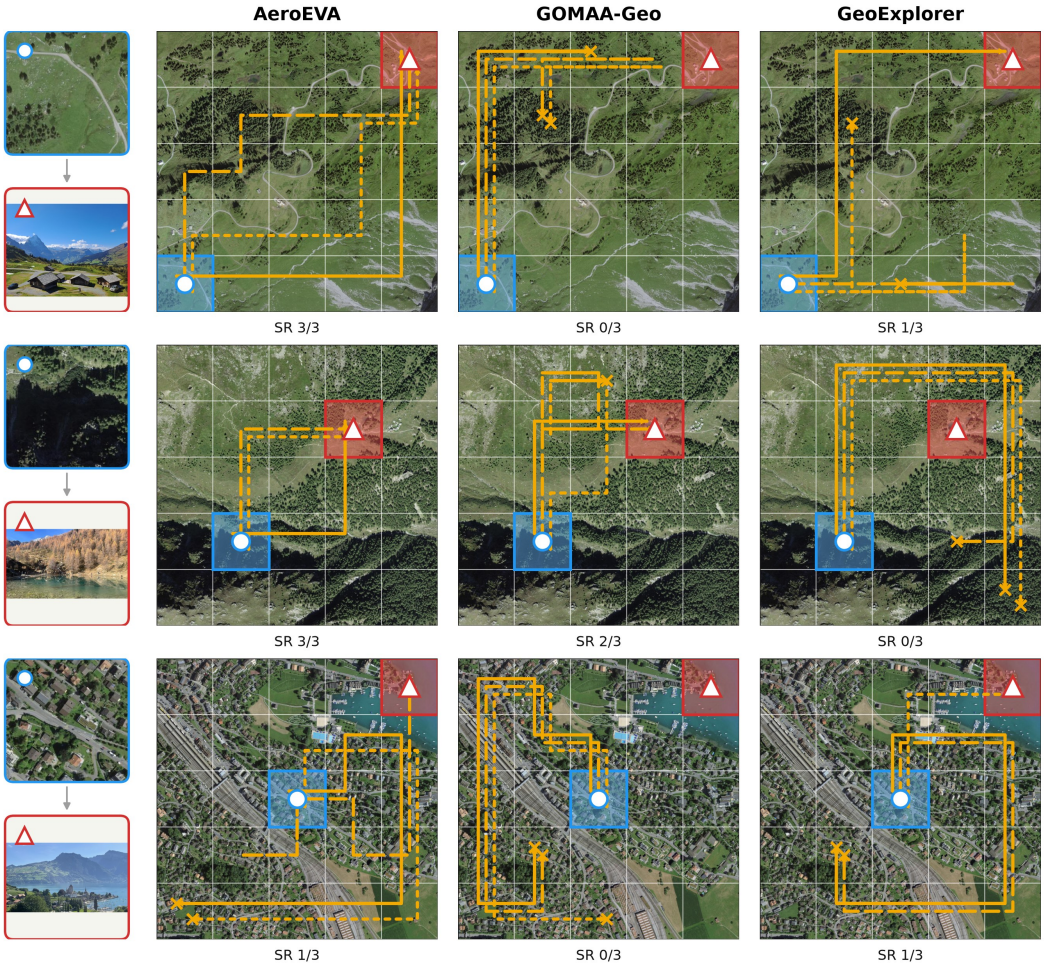


Figure 13: **SwissViewMonuments, ground-photo goal.** The goal is a street-level photograph of the monument (red frame); the agent must find the aerial cell it was taken from. *Row 1* ($C8$, alpine pass): AeroEVA reaches the cell in all three roll-outs while the reactive policies sweep the far side of the scene. *Row 2* ($C4$, mountain lake): GOMAA-Geo also succeeds twice, GeoExplorer follows the border and misses. *Row 3* ($C4$, lakeside town): a hard case in which a lake-shore photograph has to be matched to a dense urban view; every method succeeds at most once.

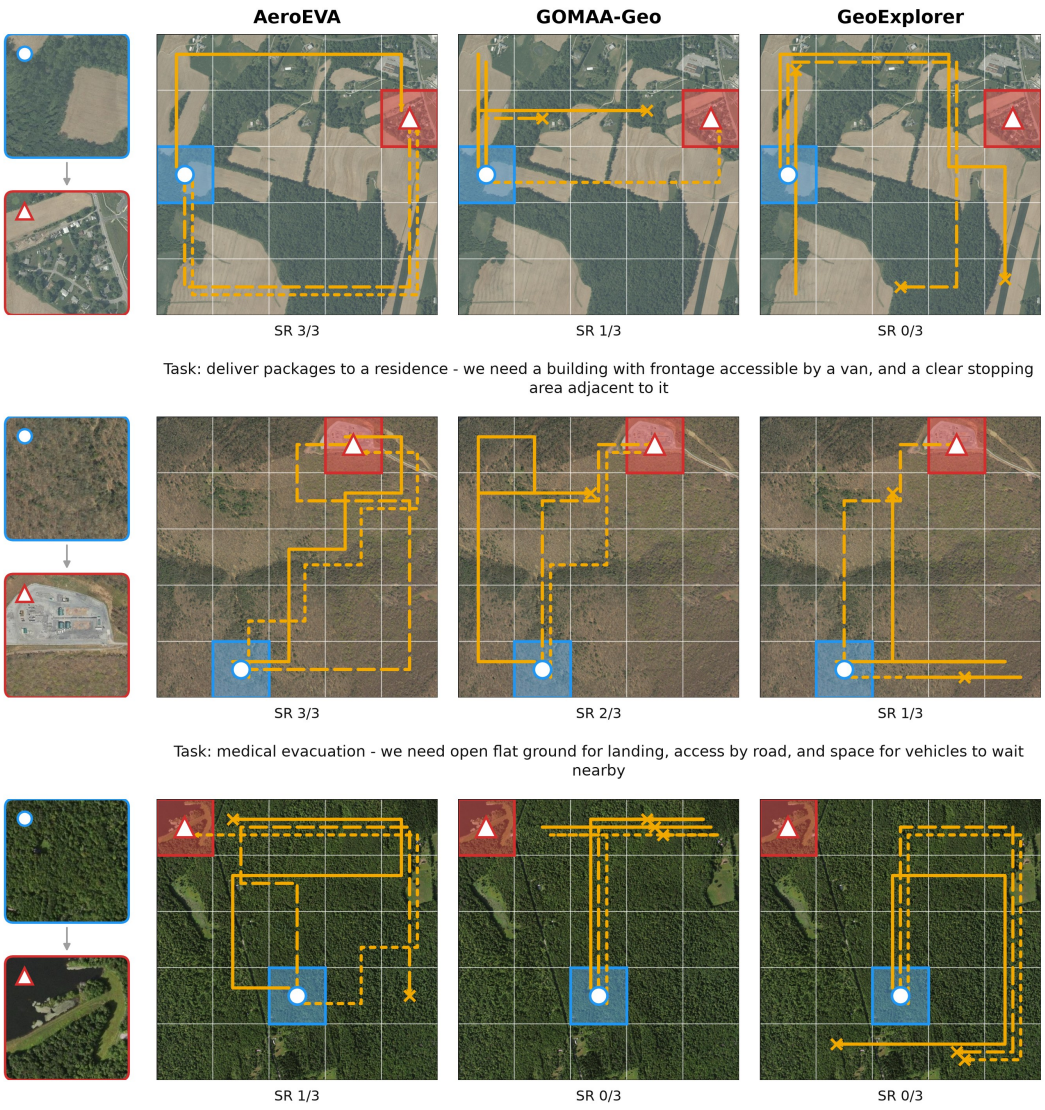


Figure 14: **AeroTask, language-specified goal.** The goal is the task description printed under each row; no goal image is given and the foresight tool is off for every method. The red frame shows the aerial view of the target cell for the reader only. *Row 1* ($C5$, package delivery): AeroEVA reaches the residence with van access in all three roll-outs, GOMAA-Geo once, GeoExplorer never. *Row 2* ($C6$, helicopter medical evacuation): the open landing ground is found by AeroEVA in every roll-out and by GOMAA-Geo in two; GeoExplorer reaches it once. *Row 3* ($C5$, fire-fighting water drafting point): the required shoreline has to be inferred from the description alone; AeroEVA succeeds once and both reactive policies fail.

C Implementation and Training Details

C.1 Visual Predictor and Feature Alignment

We use Qwen3-VL-4B-Instruct as the policy backbone. At the 224×224 resolution used for prediction, its post-merger visual grid has 8×8 tokens of dimension 2560. The predictor reads a band of width three from the side selected by the action: $n = 8$, $d_v = 2560$, and $w = 3$ in Eq. 3. We orient this band as an $8 \times 3 \times 2560$ tensor with the position nearest the next cell at band index zero. Tokens are ordered left-to-right for horizontal bands and top-to-bottom for vertical bands. The predictor outputs a single row of neighboring patch tokens (or a column, depending on the action). Tool scoring compares these tokens with goal patch tokens at the same positions.

For each of the eight positions along the input band, the three tokens across the band are concatenated into a 7680-dimensional vector. A linear projection maps it to 512 dimensions, and a learned action embedding is added. Two one-dimensional convolutional layers, each with 512 channels, kernel size three, padding one, and GELU activation, mix information along the selected row or column. Linear interpolation expands its length from eight to fourteen; a final linear layer maps each position to 1024 dimensions, giving $K = 14$ and $d_z = 1024$ in Eq. 3. The head has 6.0M parameters.

The frozen DINOv3 encoder produces fourteen-by-fourteen patch tokens from the next 224×224 image. We use the final-layer patch features, excluding class and register tokens. We select fourteen neighboring patch tokens as the target Z_{t+1} defined in Section 4.1. For unit-normalized tokens, the squared error summed over feature channels equals twice one minus their cosine similarity. Eq. 4 averages this error over aligned token positions and transitions. Target features are detached. The visual loss reaches the Qwen vision tower, its native merger, and the prediction head, but bypasses the language decoder. The separate language-hidden-state forward and inverse prediction heads and DINO-feature fusion are disabled in this recipe.

C.2 Trajectory Construction and Textual Targets

Recorded actions and supervision. Training trajectories sample legal cardinal moves uniformly until reaching the goal or exhausting the move budget. For each visited state, the optimal action label is sampled uniformly among legal moves that reduce Manhattan distance to the goal. These trajectories are generated offline: the action labels do not change the recorded behavior trajectory. The serialized assistant strings and subsequent observations retain that behavior. Only the action-word cross-entropy labels are replaced by the optimal action tokens; the input token sequence is unchanged. The implementation checks that the substituted spans have equal token length. Consequently, textual next-view targets and visual prediction targets both remain conditioned on the recorded action a_t.

Each conversation starts with the task instructions, goal image, and current observation, followed by alternating assistant responses and environment observations. Position, elapsed steps, legal moves, and tool evidence accompany each view. The last successor image is used as a visual target even when it does not appear as a further conversation turn.

Text schema. The complete response has the ordered fields

```
<belief>JUDGMENT</belief><answer>ACTION</answer>
<transitions>DESCRIPTION</transitions>
```

The first field encodes whether the goal is adjacent to the current position: `nearby` if the pre-action Manhattan distance is exactly one, and `faraway` otherwise. It is part of the structured textual modeling objective, not a separate model or training stage. The description predicts the cell reached by the recorded action in SFT and by the generated action at inference. Because it follows the action, it cannot condition that same action token. Earlier teacher-forced descriptions are available in the SFT history; inference uses the model's own generated responses.

We use SegEarth-OV3 to generate offline semantic maps and convert each recorded transition into a structured next-view description using the deterministic rules below. The maps use twelve classes: building, road, parking, vehicle, tree, grass, cropland, bare soil, water, pool, sports field, and sand.

Class proportions exclude unlabeled pixels. The four fields always occur in the order `cover`, `entry`, `change`, and `salient`:

- `cover`: up to two largest next-cell class proportions of at least 2%. Their bins are `low` below 25%, `medium` from 25% to below 50%, `high` from 50% to below 75%, and `dominant` at least 75%.
- `entry`: up to two most prevalent classes with at least 15% coverage in the next cell's 10%-width strip nearest the current cell.
- `change`: up to two classes with the largest absolute coverage changes of at least ten percentage points. A class appears or disappears when its coverage crosses the 2% presence threshold; otherwise the label is `increase` or `decrease`.
- `salient`: vehicle, pool, and sports field, in that fixed order, when next-cell coverage is at least 0.3%.

Class names and attributes use underscores, fields are separated by semicolons, and items by commas. Empty entry, change, or salient lists are written as `none`. If either transition endpoint has less than 70% labeled pixels, we omit its transition-description tag while retaining the action and goal-relative text.

An observed training label for a rightward move is

```
cover=building_medium,grass_medium;
entry=building,grass;
change=bare_soil_decrease;salient=vehicle
```

This label describes the move from cell $(0, 0)$ to $(0, 1)$ in training scene `img_394`. Semantic maps and hidden goal coordinates supply training labels; neither is inserted into an observation prompt.

SFT corpus. The SFT corpus contains 24,000 trajectories from 800 MASA training scenes, with transition descriptions retained for 96.2% of its 230,369 steps. The remaining 31 training scenes supply 300 internal validation trajectories (2,850 steps), separate from the 178-scene benchmark validation split. These are automatic semantic pseudo-labels; coverage does not establish annotation accuracy.

Tool evidence and availability. The observation contains a `<pred>` field mapping `up`, `down`, `left`, and `right` to scores or `null`. The system prompt explains the one-step prediction interface, asks the agent to assess goal adjacency, and cautions that a favored direction is unreliable for a distant goal. In SFT, the availability draw is deterministic for each trajectory record, so all its turns retain the same mask across epochs and workers. In RL, the draw is deterministic for the episode seed; the eight rollouts of a group share it. Each procedure drops evidence with probability 0.5 over its records or seeds. Dropping evidence preserves the field and sets all four values to `null`.

C.3 Optimization and Parameter Updates

Table 13 separates initialization from the full joint SFT and subsequent RL. Both SFT phases update the full VLM and the visual prediction head. The tool uses an independent frozen copy from initialization, including while the trainable head continues to change. We export the joint-SFT VLM for RL; the training-only visual head is not an auxiliary RL actor branch.

Table 13: **Joint SFT trains foresight and actions; RL refines the VLM policy.**

Phase	Objective	Updated modules	Duration
Predictor initialization	Action-token loss $+ 0.5\mathcal{L}_{\text{vis}}$	Full VLM and predictor	2 epochs
Joint multimodal SFT	Action/text-token loss $+ 0.5\mathcal{L}_{\text{vis}}$	Full VLM and predictor	1 epoch
Multi-turn RL	Navigation GRPO and entropy	Full VLM	300 updates

Both SFT phases use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, cosine learning-rate decay, and 3% warmup. The VLM, including the vision tower and native visual merger, uses learning rate 10^{-5}; the prediction head uses 10^{-4}. Four GPUs, two trajectories per device, and eight gradient-accumulation steps give a global batch of 64 trajectories. Token losses supervise the response format and targets active in each phase: initialization uses action-format supervision, while joint SFT adds the textual targets and mixed tool evidence.

C.4 Multi-Turn RL Recipe

Return and reward gates. Let d_t be the pre-action Manhattan distance to the goal, $b_t = \mathbf{1}\{d_t = 1\}$ its adjacency target, and $\widehat{b}_t$ the parsed textual judgment. Let the binary indicators m_t^{parse}, m_t^{fmt}, and m_t^{goal} denote, respectively, a parsable judgment, a valid complete response format with an accepted cardinal action, and an executed move that reaches the goal. The active per-turn reward is

$$r_t = m_t^{\text{goal}} + 0.05 m_t^{\text{fmt}} + m_t^{\text{parse}}\, w_{b_t}\big(2\mathbf{1}\{\widehat{b}_t = b_t\} - 1\big). \tag{7}$$

Here w_{b_t} is the proximity-reward weight for adjacency target b_t, with $w_1 = 0.10$ and $w_0 = 0.02$. The proximity term is zero for a missing or invalid judgment and is otherwise independent of the full-format gate. The parser can extract a cardinal action even when the complete response is malformed, so success does not require $m_t^{\text{fmt}} = 1$. Full-format validity includes the grammar of the transition-description field, but no reward scores that description's semantic content. An off-grid cardinal action can be syntactically valid; it leaves the position unchanged and consumes a move. There is no distance-progress shaping, step penalty, revisit penalty, or separate invalid-action penalty. Entering the target on the final allowed move still counts as success. Returns are sums over at most ten interaction turns.

Trajectory advantages and policy surrogate. For each initial episode, we sample $G = 8$ trajectories with the same goal, start state, and tool-availability mask. Let $R_i = \sum_t r_{i,t}$ be trajectory i's undiscounted return. Its group-relative advantage is

$$A_i = \frac{R_i - \bar{R}}{\sigma_R + 10^{-6}}, \qquad \bar{R} = \frac{1}{G}\sum_{i=1}^{G} R_i, \tag{8}$$

where σ_R is the sample standard deviation within the group. The same episode advantage supervises all generated assistant tokens, with environment and user tokens masked out. Writing $\rho_{i,k}$ for the policy likelihood ratio used by the actor update at generated token k, the clipped surrogate is

$$\ell_{i,k} = \min(\rho_{i,k} A_i, \text{clip}(\rho_{i,k}, 1-\epsilon, 1+\epsilon) A_i)\,, \qquad \epsilon = 0.2. \tag{9}$$

The update averages generated-token objectives within each trajectory and includes entropy regularization, as specified below.

Group filtering and objective. Each update samples 32 initial episodes and eight trajectories per episode, for 256 trajectories before filtering. We compute Eq. 8, then sort episode groups by their return variance and retain groups accounting for at least 90% of the total variance. If every group has zero variance, all are retained. With N original groups and N_{keep} retained groups, the implementation rescales their advantages as $A'_i = \sqrt{N_{\text{keep}}/N}\, A_i$. Let $\ell'_{i,k}$ be the clipped surrogate in Eq. 9 with A'_i substituted. The minimized policy loss is

$$\begin{aligned} \mathcal{L}_{\text{RL}} &= -\frac{1}{256}\sum_{i\in\text{kept}} \frac{1}{|\mathcal{M}_i|}\sum_{k\in\mathcal{M}_i}\big[u_{i,k} + 0.005\mathcal{H}_{i,k}\big], \\ u_{i,k} &= \begin{cases} \max(\ell'_{i,k}, 3A'_i), & A'_i < 0, \\ \ell'_{i,k}, & A'_i \geq 0. \end{cases} \end{aligned} \tag{10}$$

Here $\mathcal{M}_i$ contains trajectory i's generated assistant tokens and $\mathcal{H}_{i,k}$ is the policy entropy at that token. With one trajectory per microbatch, token averaging occurs within each trajectory; accumulation retains the nominal batch denominator of 256 after filtering. Neither a KL reward penalty nor an actor KL loss is enabled. The configured clipping interval is $[0.8, 1.2]$ with dual-clip coefficient three. The recipe uses one PPO epoch and a nominal single global minibatch of 256 trajectories before filtering. In this single-minibatch path, the implementation uses detached current log probabilities as the denominator, so the likelihood ratio is one at the update and clipping is inactive.

The RL learning rate is 10^{-6}, gradient clipping is one, and the per-device microbatch is one trajectory on four GPUs. Training uses temperature one and top-p one; validation uses greedy decoding. The rollout allows at most 96 generated tokens per assistant turn. The main training environment uses a 5×5 grid, 224×224 observations, initial Manhattan distances from three to six, and a ten-move budget.