TOWARDS ACTIVE CROSS-VIEW OBJECT GEO-LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-view object geo-localization (CVOGL) typically assumes a fixed query image, overlooking the ability of mobile agents to actively acquire more informative observations. To address this limitation, we introduce **Active Cross-View Object Geo-Localization (ActiveGeo)**, where an agent sequentially selects new viewpoints and determines when to stop, aiming to improve localization with minimal observations. We further propose **ActiveMoPT**, an ActiveGeo framework with three-stage training. First, *Multi-View Prompt-Preserving Adaptation* enables the model to aggregate multiple query views while reusing the initial prompt. Second, *Trajectory-Guided Policy Initialization* uses supervised agent trajectories to learn viewpoint selection and initial stopping behavior. Third, *Cost-Aware Policy Refinement* employs GRPO with a gain-cost reward to jointly optimize localization accuracy and observation efficiency. We also construct **ActiveGeo-858**, a zero-shot test set containing 858 scenes and 1,716 target annotations. Experiments show that ActiveMoPT achieves state-of-the-art performance on MoP-UAV using only 1.45 query views on average, and substantially outperforms previous CVOGL approaches under zero-shot evaluation on ActiveGeo-858.

1 INTRODUCTION

Cross-view object geo-localization (CVOGL) aims to determine the geographic location of a specific object indicated by point prompts in a query image on the reference image (Sun et al., 2023). The query images can be captured from devices like phones, autonomous vehicles, and drones, while the reference images are typically satellite images. CVOGL is widely used in various applications, such as smart city management (Yao et al., 2022), disaster monitoring (Chini et al., 2009; Kumar et al., 2013), and robot navigation (Singamaneni et al., 2024; Zhai et al., 2024).

Despite the cross-view discrepancies in appearance, scale, and viewpoint (Sun et al., 2023), recent CVOGL approaches (Zhang et al., 2025a; Zhang & Zhu, 2025; Zhang et al., 2025b; 2026a) have gradually matured and achieved promising localization performance. Meanwhile, the recent study (Zhang et al., 2026b) has further extended CVOGL prompts from a single modality to flexible multimodal cues, substantially broadening the applicability of CVOGL in diverse interaction scenarios.

However, existing CVOGL approaches (Huang et al., 2025; Zhang et al., 2025a;b; 2026a) still follow a passive localization paradigm (Sun et al., 2023), where the query image is fixed and localization is performed from the given observation, as shown in Fig. 1(a). This overlooks the mobility of embodied agents, which can actively acquire new views when the current observation is occluded or otherwise unfavorable. Nevertheless, existing CVOGL works lack the ability to support such active view acquisition and decision-making.

Addressing this limitation is important for making CVOGL more consistent with real-world mobile-agent applications (Singamaneni et al., 2024; Chini et al., 2009). It also introduces agentic decision-making (Xi et al., 2025) into CVOGL, extending the task from passive prediction to active interaction and broadening its research scope. Therefore, we introduce **Active Cross-View Object Geo-Localization (ActiveGeo)**, where an agent sequentially selects new informative viewpoints and determines when to stop, with the goal of improving localization using minimal observations. UAVs are particularly suitable for ActiveGeo due to their high mobility and flexibility (Zheng et al., 2020). We therefore focus on the UAV-based setting, as shown in Fig. 1(b).

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.

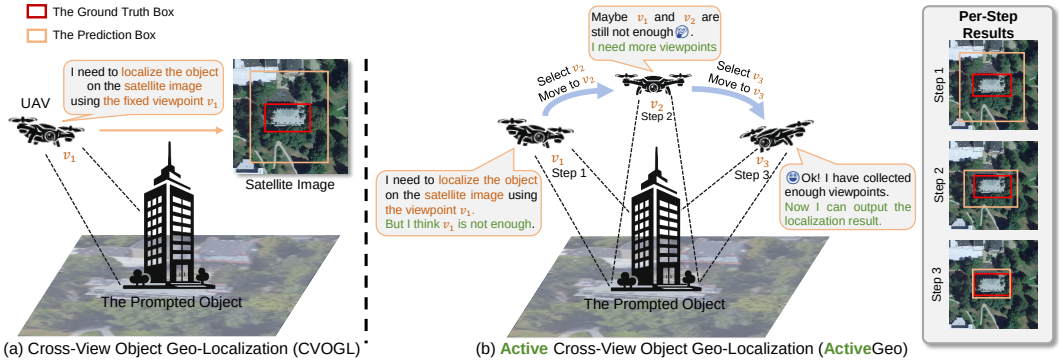

Figure 1: Comparison between previous passive cross-view object geo-localization (CVOGL) and our active cross-view object geo-localization (ActiveGeo). (a) Conventional CVOGL localizes the target from a fixed query view. (b) ActiveGeo enables a UAV agent to sequentially acquire informative viewpoints and adaptively stop when sufficient observations have been collected. Please refer to zoomed-in view for better visualization.

Under this new formulation, a practical ActiveGeo should satisfy three essential requirements. (1) *The model should support sequentially acquired query images while reusing the initial prompt*, since the target identity remains unchanged as the agent moves and repeated prompting is unrealistic. (2) *The agent should select informative viewpoints based on its current localization state*, since unobserved views are unavailable before movement and additional observations may be redundant or noisy. (3) *The agent should decide when to stop acquiring new views*, balancing potential localization gains against sensing, movement, and computation costs.

To meet these requirements, we build the **ActiveMoPT** upon MoPT (Zhang et al., 2026b) through a three-stage training framework. (1) In *Multi-View Prompt-Preserving Adaptation*, we introduce a multi-view adapter and gain-aware loss that enable MoPT to progressively aggregate multiple query observations while reusing the initial prompt. (2) In *Trajectory-Guided Policy Initialization*, we construct agent trajectory data and perform supervised fine-tuning (SFT), allowing ActiveMoPT to learn viewpoint selection and acquire an initial stopping capability. (3) In *Cost-Aware Policy Refinement*, we optimize the policy through group relative policy optimization (GRPO) with a gain-cost reward that jointly considers localization gain and observation cost, leading to a better accuracy-efficiency trade-off. Besides, we further construct a dedicated zero-shot test set **ActiveGeo-858** beyond the existing MoP-UAV benchmark (Zhang et al., 2026b) for ActiveGeo. It is built upon GeoText-1652 (Chu et al., 2024) and contains 858 scenes and 1,716 target annotations, with strict separation from MoP-UAV benchmark. Extensive experimental results show that ActiveMoPT achieves state-of-the-art localization performance on the MoP-UAV Benchmark (Zhang et al., 2026b) while using only 1.45 query views on average. Moreover, ActiveMoPT also substantially outperforms previous methods in zero-shot evaluation on ActiveGeo-858, demonstrating strong generalization to unseen scenes. Our main contributions are summarized as follows:

- We introduce **Active Cross-View Object Geo-Localization (ActiveGeo)**, extending conventional CVOGL from passive prediction to active interaction. The new task requires an agent to actively acquire and select informative viewpoints, and determine when to stop under a limited observation budget.
- We construct **ActiveMoPT** through a three-stage training framework, consisting of *Multi-View Prompt-Preserving Adaptation*, *Trajectory-Guided Policy Initialization*, and *Cost-Aware Reinforcement Refinement*.
- We construct **ActiveGeo-858**, a dedicated zero-shot test set containing 858 scenes and 1,716 target annotations for ActiveGeo.
- Extensive experiments show that ActiveMoPT achieves state-of-the-art performance on MoP-UAV benchmark (Zhang et al., 2026b) using only 1.45 query views on average, while also substantially outperforming previous approaches under zero-shot evaluation on ActiveGeo-858.

2 Related Work

Cross-view object geo-localization (CVOGL). CVOGL aims to localize a prompted object in a reference image from a cross-view query image. DetGeo (Sun et al., 2023) first formalizes this task, while subsequent approaches (Li et al., 2025; Zhang & Zhu, 2025; Huang et al., 2025; Zhang et al., 2026a) further improve prompt representation, cross-view interaction, and localization decoding.

More recently, MoPT (Zhang et al., 2026b) broadens the prompting interface of CVOGL by allowing the target object to be specified using point, box, language, or their combinations. It encodes different prompt modalities and concatenates them into prompt tokens, which sequentially interact with UAV and satellite features to establish cross-view target correspondence. The resulting tokens are finally used to regress the target location on the satellite image.

Despite these advances, existing CVOGL approaches remain passive. The query image is fixed before localization and the model cannot decide whether or where to acquire additional observations. ActiveGeo extends this formulation to active perception, requiring a mobile agent to sequentially select new viewpoints and determine when to stop.

3 Problem Formulation

From passive CVOGL to ActiveGeo. Cross-view object geo-localization (CVOGL) assumes a fixed query observation. In the UAV-based setting considered in this work, given a satellite reference image I^s, a query UAV image I_{v_1}, and a prompt p specifying the target object, the model directly predicts its location $\hat{b}$ as

$$\hat{b} = f(I^s, I_{v_1}, p), \tag{1}$$

where $\hat{b}$ denotes the predicted target box. Since existing CVOGL relies entirely on the fixed query view I_{v_1}, it cannot acquire additional observations to resolve ambiguities and improve localization when the initial observation is unfavorable.

In contrast, ActiveGeo allows a UAV agent to sequentially acquire observations from new viewpoints. Let $\mathcal{V} = \{v_1, \ldots, v_M\}$ denote the candidate viewpoints. Each episode starts from an initial view I_{v_1} and a target prompt p. The prompt is provided only with the initial observation, while subsequent views must be associated with the same target without repeated prompting.

At step t, let $\tilde{v}_t$ denote the viewpoint acquired, with $\tilde{v}_1 = v_1$. The observation history $\mathcal{H}_t$ is

$$\mathcal{H}_t = \{(I_{\tilde{v}_1}, p), I_{\tilde{v}_2}, \ldots, I_{\tilde{v}_t}\}, \tag{2}$$

from which the model predicts the current localization $\hat{b}_t$ at step t

$$\hat{b}_t = f(I^s, \mathcal{H}_t). \tag{3}$$

The agent then decides its action d_t about whether to stop or continue, where $d_t \in \{\textsc{Stop}, \textsc{Continue}\}$. If it continues, it selects an unseen viewpoint $\tilde{v}_{t+1} \in \mathcal{V} \setminus \mathcal{V}_t^{\text{obs}}$, where $\mathcal{V}_t^{\text{obs}}$ denotes the viewpoints already acquired. The UAV then moves to $\tilde{v}_{t+1}$ and obtains the corresponding observation, which is appended to $\mathcal{H}_t$. If it stops, the current prediction $\hat{b}_t$ is returned as the final localization result.

Notably, the visual content of an unseen viewpoint is unavailable before it is selected. The agent must therefore infer where to observe next from its current localization state and observation history. ActiveGeo further seeks accurate localization with as few additional observations as necessary, requiring the agent to jointly determine *where to observe* and *when to stop*.

4 Methodology

We build ActiveMoPT upon MoPT (Zhang et al., 2026b), whose prompt tokens mediate UAV-satellite interaction and provide a natural foundation for preserving the initially specified target across subsequent unprompted UAV views.

As shown in Fig. 2, ActiveMoPT consists of a multi-view localizer and an action policy. The localizer progressively integrates newly acquired UAV views and updates the localization of the target

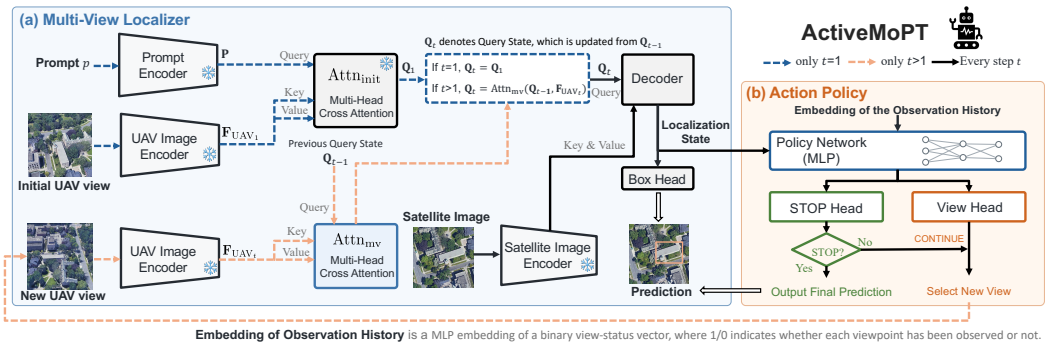

Figure 2: Overall architecture of ActiveMoPT, an agent system for ActiveGeo that performs localization, viewpoint selection, and stopping. ActiveMoPT is initialized from MoPT (Zhang et al., 2026b). At $t = 1$, before acquiring additional UAV views, its localizer is identical to MoPT. (a) The multi-view localizer updates the query state with newly acquired UAV views and predicts the target location on the satellite image. During Multi-View Prompt-Preserving Adaptation (Sec. 4.1), the encoders and $\text{Attn}_{\text{init}}$ are frozen. (b) The action policy decides whether to stop or acquire a new UAV view. Candidate UAV images are unavailable before selection, and the View Head selects among unobserved viewpoint IDs. Please refer to zoomed-in view for better visualization.

specified in the initial prompt, while the policy decides whether to stop and, if not, which unseen viewpoint to acquire next. ActiveMoPT is trained in three stages. (1) *Multi-View Prompt-Preserving Adaptation* trains the localizer to integrate additional views without repeated prompting. (2) With the localizer frozen, *Trajectory-Guided Policy Initialization* trains the policy from oracle trajectories to learn initial stopping and viewpoint selection behavior. (3) *Cost-Aware Policy Refinement* further optimizes the policy through closed-loop interaction, balancing performance improvement against view acquisition cost. Together, these stages turn ActiveMoPT into an ActiveGeo agent that can continuously update its localization, actively acquire informative viewpoints, and stop once sufficient cross-view information has been collected.

4.1 Multi-View Prompt-Preserving Adaptation

To build the foundation of ActiveGeo, ActiveMoPT must first be able to accumulate evidence from newly acquired views while preserving the target specified by the initial prompt. Since MoPT (Zhang et al., 2026b) only localizes the target from a single prompted UAV view, we adapt it to handle a sequence of UAV observations where only the first view is prompted. The adaptation needs to address two challenges, *i.e.*, (1) maintaining the target specified by the initial prompt when incorporating new UAV views, and (2) effectively improving localization with newly acquired views while avoiding performance drops caused by uninformative observations.

Prompt-preserving query update. In MoPT (Zhang et al., 2026b), the prompt tokens attend to the initial UAV view and become query tokens carrying information about the prompted target. We therefore reuse these tokens for each newly acquired view, allowing new observations to update the same tokens without requiring another prompt.

Specifically, as shown in Fig. 2, let $\mathbf{P}$ denote the tokens encoded from the initial prompt and $\mathbf{F}_{\text{UAV}_t}$ the UAV features at step t. Using the original Prompt-UAV cross attention $\text{Attn}_{\text{init}}$ in MoPT, we obtain the initial query tokens $\mathbf{Q}_1$

$$\mathbf{Q}_1 = \text{Attn}_{\text{init}}\left(\mathbf{P}, \mathbf{F}_{\text{UAV}_1}\right) \tag{4}$$

where $\mathbf{P}$ serves as queries and $\mathbf{F}_{\text{UAV}_1}$ as keys and values. For each subsequent UAV view, we use a shared multi-view cross attention module Attn_{mv} to update the query tokens $\mathbf{Q}_t$

$$\mathbf{Q}_t = \text{Attn}_{\text{mv}}\left(\mathbf{Q}_{t-1}, \mathbf{F}_{\text{UAV}_t}\right), \qquad t = 2, \ldots, T, \tag{5}$$

where $\mathbf{Q}_{t-1}$ serves as queries and $\mathbf{F}_{\text{UAV}_t}$ as keys and values.

Preserve-Improve loss. Sequentially adding UAV views does not guarantee a better localization, since a newly acquired view may provide either useful or misleading information. We therefore

expect each view update to improve the current prediction when useful, while avoiding causing performance degradation.

Specifically, for sample i, let u_1^i and u_T^i denote the IoUs between predicted boxes and ground-truth boxes obtained from the initial UAV view and after multi-view updating, respectively. Given a localization success threshold τ, we define

$$\mathcal{L}_{\mathrm{PI}}^i = \begin{cases} \max\left(0, \tau - u_T^i\right), & u_1^i \geq \tau, \\ \max\left(0, u_1^i - u_T^i\right) + \max\left(0, \tau - u_T^i\right), & u_1^i < \tau. \end{cases} \tag{6}$$

For an initially successful prediction ($u_1^i \geq \tau$), the first term penalizes the updated result only if it falls below the success threshold. For an initially failed prediction ($u_1^i < \tau$), the latter two terms encourage the updated IoU to exceed the initial IoU and further reach the success threshold.

Finally, the multi-view localizer is trained with

$$\mathcal{L}_{\mathrm{MV}} = \mathcal{L}_{\mathrm{Det}}^{(1)} + \mathcal{L}_{\mathrm{Det}}^{(T)} + \mathcal{L}_{\mathrm{PI}}, \tag{7}$$

where $\mathcal{L}_{\mathrm{Det}}^{(1)}$ and $\mathcal{L}_{\mathrm{Det}}^{(T)}$ are DETR-style detection losses (Carion et al., 2020) and supervise the initial and final localization results, respectively. The $\mathcal{L}_{\mathrm{Det}}$ optimizes localization accuracy, while $\mathcal{L}_{\mathrm{PI}}$ constrains how the prediction changes after additional views are included.

4.2 Trajectory-Guided Policy Initialization

With the multi-view localizer in place, ActiveMoPT still needs to learn *when to stop* and *which view to acquire next*. Therefore, we introduce a supervised fine-tuning (SFT) stage (Fig. 3(a)) to initialize the action behavior, which provides a cold start and equips the model with a basic awareness of the action policy, ensuring the subsequent reinforcement learning (RL) explores within a meaningful policy space (Luo et al., 2025; Zheng et al., 2026).

Construction of the oracle trajectory dataset. During training, all candidate UAV views and the ground-truth box are available for each scene. We use this information offline to construct supervised trajectories by evaluating candidate viewpoints and assigning an action at each step.

Given the current observation history $\mathcal{H}_t$, the frozen localizer produces a prediction $\hat{b}_t$. Its IoU u_t with the ground-truth box b^* is

$$u_t = \mathrm{IoU}(\hat{b}_t, b^*). \tag{8}$$

For each unseen viewpoint v, we evaluate the frozen localizer using $\mathcal{H}_t$ together with I_v, and denote the resulting IoU by $u_{t+1}^{(v)}$. The localization gain $\Delta u_t^{(v)}$ of this candidate is

$$\Delta u_t^{(v)} = u_{t+1}^{(v)} - u_t. \tag{9}$$

If the largest gain exceeds a predefined threshold, we assign CONTINUE and the corresponding viewpoint as the action labels. The selected view is then added to the observation history, and the process repeats. Otherwise, we assign STOP. The resulting sequence of observation histories and action labels forms the oracle trajectory dataset for SFT.

Policy learning in SFT. During SFT, we freeze the localizer of ActiveMoPT, while the generated trajectories provide supervision for the two policy heads. The stop head learns whether to terminate the episode, while the view head learns which unseen viewpoint to select when continuing. The training objective $\mathcal{L}_{\mathrm{SFT}}$ is

$$\mathcal{L}_{\mathrm{SFT}} = \mathcal{L}_{\mathrm{stop}} + \mathcal{L}_{\mathrm{view}}, \tag{10}$$

where $\mathcal{L}_{\mathrm{stop}}$ is binary cross-entropy and $\mathcal{L}_{\mathrm{view}}$ is cross-entropy. At step t, acquired and unavailable viewpoints are masked during view selection.

4.3 Cost-Aware Policy Refinement

The SFT policy learns initial viewpoint selection and stopping behavior by imitating greedy oracle trajectories. However, the oracle is constructed from localization gains and does not explicitly account for the cost of acquiring additional views. We therefore further refine the policy through

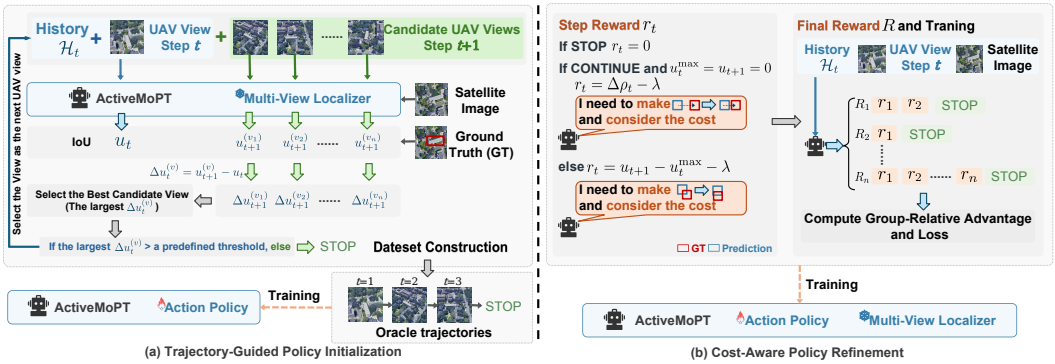

Figure 3: Training stages of the ActiveMoPT action policy. (a) Trajectory-Guided Policy Initialization. We construct greedy oracle trajectories to supervise the action policy during SFT. (b) Cost-Aware Policy Refinement. Based on GRPO (Shao et al., 2024), the action policy is further optimized in closed-loop interaction. A gain-cost reward evaluates localization improvement while eliminating unnecessary observations. Please refer to zoomed-in view for better visualization.

closed-loop RL to learn the trade-off between localization improvement and observation cost. During this stage, the localizer remains frozen, and only the action policy is optimized using group relative policy optimization (GRPO) (Shao et al., 2024).

Gain-cost reward. As shown in Fig. 3(b), we reward performance improvement while charging for each additional UAV view. For each trajectory, let u_t denote the localization IoU after step t, and let

$$u_t^{\max} = \max_{k \le t} u_k \tag{11}$$

denote the best IoU obtained so far at step t. The step reward r_t is defined as

$$r_t = \begin{cases} 0, & d_t = \textsc{Stop}, \\ \Delta\rho_t - \lambda, & d_t = \textsc{Continue}, \quad u_t^{\max} = 0,\ u_{t+1} = 0, \\ (u_{t+1} - u_t^{\max}) - \lambda, & \text{otherwise}, \end{cases} \tag{12}$$

where λ is the cost of acquiring one additional view. The IoU term rewards improvement over $u_t^{\max}$ and penalizes any drops below that level. When both $u_t^{\max}$ and u_{t+1} are zero, IoU cannot indicate whether the new view improves the prediction. We therefore use the change in center distance as the reward signal in this case. Let ρ_t denote the normalized center distance between the predicted and ground-truth boxes, and let $\rho_t^{\min} = \min_{k \le t} \rho_k$ be the smallest distance obtained so far. We define $\Delta\rho_t = \rho_t^{\min} - \rho_{t+1}$, which is positive when the new prediction moves closer to the target.

To account for the rewards from subsequent decisions, the final reward R is computed from each step to the end of the trajectory,

$$R = \sum_{k=t}^{T-1} r_k, \tag{13}$$

where T is the number of observations acquired before termination.

Policy optimization in RL. For each initial episode, we compare R of sampled trajectories and standardize them to obtain the group-relative advantage A. The action policy is then optimized using the clipped GRPO objective $\mathcal{L}_{\mathrm{RL}}$ (Shao et al., 2024)

$$\mathcal{L}_{\mathrm{RL}} = -\mathbb{E}_t \left[\min \left(\omega_t A,\ \mathrm{clip}(\omega_t, 1-\epsilon_c, 1+\epsilon_c) A \right) \right] + \beta\, \widehat{\mathrm{KL}} \left(\pi_\phi \,\|\, \pi_{\mathrm{SFT}} \right), \tag{14}$$

where $\omega_t = \pi_\phi / \pi_{\mathrm{old}}$ is the policy probability ratio, ϵ_c is the clipping radius, and β controls the sampled KL penalty to the frozen SFT policy π_{SFT}.

5 EXPERIMENTS

5.1 DATASETS AND EVALUATION PROTOCOL

MoP-UAV. We follow the MoP-UAV split (Zhang et al., 2026b), which contains 598 training scenes and 100 test scenes, and evaluate all 17,447 target episodes in the test split. Each episode starts from one prompted UAV image. ActiveMoPT may then select an unseen UAV view or output STOP, while reusing the point (**P**), box (**B**), and/or language (**L**) prompt provided with the initial observation.

ActiveGeo-858. To evaluate zero-shot generalization beyond MoP-UAV (Zhang et al., 2026b), we construct ActiveGeo-858 from the GeoText-1652 test split (Chu et al., 2024). It contains 858 scenes that do not overlap with the MoP-UAV (Zhang et al., 2026b). Experts annotate two visually identifiable targets in each scene, yielding 1,716 evaluation episodes. Each target is annotated with a point (**P**) and a box (**B**) prompt. No ActiveGeo-858 image is used for training, checkpoint selection, or hyperparameter tuning.

Evaluation protocol. Every episode starts from one initial UAV observation. ActiveMoPT then selects additional views and returns the prediction after selecting STOP or reaching a 7-view limit, whereas passive approaches (Sun et al., 2023; Zhang et al., 2026a; Zhang & Zhu, 2025; Zhang et al., 2026b; Huang et al., 2025) use only the initial UAV view. Ground-truth IoU is never available to the policy during evaluation.

5.2 EVALUATION METRICS

Localization accuracy. We report mean intersection over union (mIoU) and Acc@τ (Sun et al., 2023), the fraction of final predictions whose IoU is at least τ, for $\tau \in \{0.25, 0.50\}$.

Observation efficiency. We report Avg. Views, the average number of UAV observations used per episode, including the mandatory initial image. A lower Avg. Views indicates that the agent reaches its final prediction with fewer additional observations.

5.3 IMPLEMENTATION DETAILS

Please refer to the appendix.

5.4 COMPARISON RESULTS

Quantitative results. Table 1 compares ActiveMoPT with previous CVOGL approaches on both MoP-UAV (Zhang et al., 2026b) and ActiveGeo-858 dataset.

(1) *Active viewpoint selection efficiently enhances localization performance.* Under the point-prompt (**P**) setting, ActiveMoPT achieves the best localization results on both benchmarks. On MoP-UAV (Zhang et al., 2026b) dataset, it improves the MoPT from 0.368 to 0.384 mIoU while using only 1.291 views on average. The improvement can also be observed in Table 1(b) and Table 1(c). These results show the efficiency of ActiveMoPT.

(2) *ActiveMoPT generalizes well to unseen scenes.* On ActiveGeo-858, ActiveMoPT consistently improves MoPT (Table 1(b)). Since no image from ActiveGeo-858 is used during checkpoint selection in training, these results show that the learned multi-view localization and action policy transfer well beyond MoP-UAV (Zhang et al., 2026b) dataset.

Qualitative results. Please refer to the appendix.

5.5 ABLATION STUDY

Component and training stage ablation. Table 2 evaluates the components of Multi-View Prompt-Preserving Adaptation and the learned action policy from SFT and RL stage.

(1) *Multi-view adaptation enables consistent gains from additional observations* (Table 2(a)). With $\mathrm{Attn}_{\mathrm{init}}$, $\mathrm{Attn}_{\mathrm{mv}}$, and $\mathcal{L}_{\mathrm{PI}}$, mIoU improves from 0.389 with one view to 0.398, 0.399, and 0.400 with two, three, and four views, respectively. These results show that the adapted localizer consistently benefits from additional observations, supporting our central hypothesis that acquiring new

Table 1: Comparison on MoP-UAV (Zhang et al., 2026b) and ActiveGeo-858. (a) Comparisons of all methods under the point-prompt protocol. (b) Comparisons under **B** and **P+B** prompts on both benchmarks. (c) Comparisons of different prompt combinations on MoP-UAV (Zhang et al., 2026b) dataset since ActiveGeo-858 contains no language annotations.

	MoP-UAV (Zhang et al., 2026b)				ActiveGeo-858			
Method	mIoU	Acc@.25	Acc@.50	Avg. Views	mIoU	Acc@.25	Acc@.50	Avg. Views
(a) Point prompt (P)								
DetGeo (Sun et al., 2023)	0.327	0.483	0.321	1.000	0.287	0.487	0.270	1.000
OCGNet (Huang et al., 2025)	0.337	0.497	0.335	1.000	0.255	0.432	0.210	1.000
ReCOT (Zhang et al., 2026a)	0.318	0.583	0.280	1.000	0.367	0.631	0.346	1.000
TROGeo (Zhang & Zhu, 2025)	0.314	0.586	0.265	1.000	0.293	0.536	0.213	1.000
MoPT (Zhang et al., 2026b)	0.368	0.598	0.405	1.000	0.381	0.640	0.379	1.000
ActiveMoPT (**Ours**)	**0.384**	**0.611**	**0.420**	1.291	**0.391**	**0.643**	**0.409**	1.195
(b) Comparison with spatial prompts (B and P+B)								
B								
MoPT (Zhang et al., 2026b)	0.373	0.617	0.407	1.000	0.386	0.657	0.405	1.000
ActiveMoPT (**Ours**)	**0.389**	**0.630**	**0.432**	1.384	**0.414**	**0.689**	**0.451**	1.429
P+B								
MoPT (Zhang et al., 2026b)	0.375	0.619	0.405	1.000	0.394	0.672	0.421	1.000
ActiveMoPT (**Ours**)	**0.395**	**0.639**	**0.434**	1.403	**0.422**	**0.696**	**0.476**	1.371

(c) Comparisons of different prompt combinations on MoP-UAV (Zhang et al., 2026b)

	MoPT (Zhang et al., 2026b)				**ActiveMoPT (Ours)**			
Prompt modality	mIoU	Acc@.25	Acc@.50	Avg. Views	mIoU	Acc@.25	Acc@.50	Avg. Views
L	**0.365**	**0.611**	0.405	1.000	**0.365**	0.600	**0.407**	1.478
P+L	0.378	0.621	0.430	1.000	**0.390**	**0.626**	**0.448**	1.417
B+L	0.385	0.628	0.439	1.000	**0.407**	**0.649**	**0.472**	1.452
P+B+L	0.392	0.637	0.444	1.000	**0.414**	**0.659**	**0.478**	1.450

ReCOT (Zhang et al., 2026a) uses its selected checkpoint at 6 recurrent steps. Best and second-best localization results of part (a), (b), and (c) are shown in **bold red** and blue, respectively.

Table 2: Ablations on MoP-UAV (Zhang et al., 2026b) dataset using prompt **P+B+L**. (a) Ablation of the components in Multi-View Prompt-Preserving Adaptation stage (b) Ablation of the SFT and RL stage. "Random" uses a sampled number of 1-4 UAV views, while "Adaptive" lets the learned active action policy determine when to stop, up to the maximum number of UAV views.

Configuration	Input UAV Views	mIoU	Acc@.25	Acc@.50	Avg. Views Used
(a) Multi-view prompt-preserving adaptation					
with $\mathrm{Attn}_{\mathrm{init}}$ and $\mathcal{L}_{\mathrm{PI}}$	1	0.392	0.637	0.444	1.000
	2	0.155	0.255	0.051	2.000
with $\mathrm{Attn}_{\mathrm{init}}$ and $\mathrm{Attn}_{\mathrm{mv}}$	1	0.388	0.637	0.437	1.000
	2	0.395	**0.640**	0.448	2.000
	3	0.395	0.633	0.449	3.000
	4	0.396	0.633	0.448	4.000
	Random	0.393	0.637	0.446	2.500
with $\mathrm{Attn}_{\mathrm{init}}$, $\mathrm{Attn}_{\mathrm{mv}}$, and $\mathcal{L}_{\mathrm{PI}}$	1	0.389	0.632	0.440	1.000
	2	0.398	**0.640**	0.455	2.000
	3	0.399	0.639	**0.461**	3.000
	4	**0.400**	0.639	0.459	4.000
	Random	0.396	0.636	0.454	2.500
(b) Acquisition policy					
Passive Policy (MoPT (Zhang et al., 2026b) Baseline)	1	0.392	0.637	0.444	1.000
Active Policy (ActiveMoPT, SFT)	Adaptive	0.411	0.657	0.474	1.886
Active Policy (ActiveMoPT, SFT+GRPO)	Adaptive	**0.414**	**0.659**	**0.478**	1.450

For the 1-4 and Random settings, the initial view is always included, and the remaining views are randomly sampled. Best and second-best localization results of parts (a) and (b) are shown in **bold red** and blue, respectively.

viewpoints can improve localization. The performance reaches a plateau after 2-3 views on MoP-UAV (Zhang et al., 2026b), further supporting our motivation to control acquisition cost.

(2) *Both* $\mathrm{Attn}_{\mathrm{mv}}$ *and* $\mathcal{L}_{\mathrm{PI}}$ *contribute to performance improvement* (Table 2(a)). Introducing $\mathrm{Attn}_{\mathrm{mv}}$ avoids the performance drop caused by reusing $\mathrm{Attn}_{\mathrm{init}}$. Adding $\mathcal{L}_{\mathrm{PI}}$ further improves multi-view performance. These results show that $\mathrm{Attn}_{\mathrm{mv}}$ enables the model to incorporate new observations, while $\mathcal{L}_{\mathrm{PI}}$ helps maintain and enhance localization performance across views.

Table 3: Ablations of key hyperparameters on MoP-UAV (Zhang et al., 2026b) dataset using prompt **P+B+L**. (a) Ablations of the success thresholds τ in $\mathcal{L}_{\text{PI}}$. (b) Ablations of the acquisition cost λ in r_t. "Random" uses a sampled number of 1-4 UAV views, while "Adaptive" lets the learned active action policy determine when to stop, up to the maximum number of UAV views.

Configuration	Input UAV Views	mIoU	Acc@.50	Avg. Views Used
(a) Threshold τ in $\mathcal{L}_{\text{PI}}$				
$\tau = 0.3$	1	0.389	0.439	1.000
	2	0.393	0.444	2.000
	3	0.388	0.438	3.000
	4	0.386	0.434	4.000
	Random	0.389	0.438	2.500
$\tau = 0.5$	1	0.389	0.440	1.000
	2	0.398	0.455	2.000
	3	0.399	**0.461**	3.000
	4	**0.400**	0.459	4.000
	Random	0.396	0.454	2.500
$\tau = 0.7$	1	0.387	0.436	1.000
	2	0.392	0.446	2.000
	3	0.394	0.450	3.000
	4	0.396	0.452	4.000
	Random	0.391	0.444	2.500
(b) Acquisition cost λ				
GRPO, $\lambda = 0$	Adaptive	**0.417**	**0.479**	1.933
GRPO, $\lambda = 0.01$	Adaptive	0.414	0.478	1.450
GRPO, $\lambda = 0.1$	Adaptive	0.389	0.440	1.000

For the 1-4 and Random settings, the initial view is always included, and the remaining views are randomly sampled. Shaded rows indicate the selected hyperparameters. Best and second-best localization results of part (a) and (b) are shown in **bold red** and blue, respectively.

(3) *SFT enables active acquisition, while GRPO further improves efficiency* (Table 2(b)). After SFT, ActiveMoPT gains the ability to actively acquire viewpoints and outperforms both the passive MoPT (Zhang et al., 2026b) baseline and the fixed/random acquisition results in Table 2(a), demonstrating the effectiveness of the action policy. GRPO further improves mIoU to 0.414 and Acc@.50 to 0.478, while reducing the average number of views from 1.886 to 1.450. These results show that cost-aware refinement makes ActiveMoPT more efficient by reducing unnecessary observations without sacrificing localization performance.

Hyperparameter ablation. (1) *A moderate success threshold τ provides effective supervision for multi-view adaptation* (Table 3(a)). When τ is too small, low-quality predictions can already be regarded as successful, weakening the constraint of $\mathcal{L}_{\text{PI}}$ on subsequent multi-view updates. A larger τ imposes a stricter success criterion and places stronger constraints on difficult samples, resulting in smaller multi-view gains at $\tau = 0.7$. In contrast, $\tau = 0.5$ consistently improves localization as additional views are introduced and achieves the best overall multi-view performance. We therefore use $\tau = 0.5$ to balance insufficient and overly strict supervision.

(2) *Acquisition cost λ affects the efficiency of the learned policy* (Table 3(b)). Setting $\lambda = 0$ achieves the highest mIoU of 0.417 but uses 1.933 views on average, indicating a more aggressive action policy. Increasing λ to 0.1 reduces the average number of views to 1.000, indicating that the policy becomes overly conservative and falls back to passive method. We therefore select $\lambda = 0.01$, which achieves 0.414 mIoU and 0.478 Acc@.50 with only 1.450 views on average. This provides a favorable trade-off between localization performance and cost, making ActiveMoPT more efficient without sacrificing performance.

6 Conclusion

We introduced ActiveGeo, extending CVOGL from a fixed-query task to an active setting where an agent can acquire additional viewpoints to enhance performance and decide when to stop. To address this task, we developed ActiveMoPT, which integrates new observations while preserving the initial prompt and employs a cost-aware action policy. Experiments demonstrate the effectiveness and efficiency of ActiveMoPT. ActiveGeo provides a promising direction for boosting CVOGL research.

AI USAGE DISCLOSURE

Generative AI tools were used only for code checking, manuscript language refinement, and organizational suggestions. They were not involved in the development of the methodology, experimental design, hypothesis formulation, implementation of the proposed method, or interpretation of experimental results. All AI-assisted outputs were manually reviewed and verified by the authors. The authors take full responsibility for the final manuscript, including all text, claims, code, and other AI-assisted content.

REFERENCES

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229, 2020.

Marco Chini, Nazzareno Pierdicca, and William J. Emery. Exploiting SAR and VHR optical images to quantify damage caused by the 2003 Bam earthquake. *IEEE Transactions on Geoscience and Remote Sensing*, 47(1):145–152, 2009.

Meng Chu, Zhedong Zheng, Wei Ji, Tingyu Wang, and Tat-Seng Chua. Towards natural language-guided drones: GeoText-1652 benchmark with spatial relation matching. In *Computer Vision – ECCV 2024*, volume 15069 of *Lecture Notes in Computer Science*, pp. 213–231. Springer, 2024. doi: 10.1007/978-3-031-73247-8_13.

Zheyang Huang, Jagannath Aryal, Saeid Nahavandi, Xuequan Lu, Chee Peng Lim, Lei Wei, and Hailing Zhou. Object-level cross-view geolocalization with location enhancement and multihead cross attention. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18:22880–22890, 2025. doi: 10.1109/JSTARS.2025.3603506.

Anuj Kumar, Hiesik Kim, and Gerhard P. Hancke. Environmental monitoring systems: A review. *IEEE Sensors Journal*, 13(4):1329–1339, 2013.

Zhongyang Li, Xin Yuan, Wei Liu, and Xin Xu. VAGeo: View-specific attention for cross-view object geo-localization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi-Xin Yang. AdaThinkDrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 18–24 Jul 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Phani Teja Singamaneni, Pilar Bachiller-Burgos, Luis J. Manso, Anaís Garrell, Alberto Sanfeliu, Anne Spalanzani, and Rachid Alami. A survey on socially aware robot navigation: Taxonomy and future challenges. *The International Journal of Robotics Research*, 43(10):1533–1572, 2024.

Yuxi Sun, Yunming Ye, Jian Kang, Rubén Fernández-Beltran, Shanshan Feng, Xutao Li, Chuyao Luo, Puzhao Zhang, and Antonio Plaza. Cross-view object geo-localization in a local region with satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–16, 2023.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and

Yu-Gang Jiang. AgentGym-RL: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*, 2025.

Jing Yao, Danfeng Hong, Lianru Gao, and Jocelyn Chanussot. Multimodal remote sensing benchmark datasets for land cover classification. In *2022 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 4807–4810, 2022.

Ruoming Zhai, Jingui Zou, Vincent J.L. Gan, Xianquan Han, Yushuo Wang, and Yinzhi Zhao. Semantic enrichment of bim with indoorgml for quadruped robot navigation and automated 3D scanning. *Automation in Construction*, 166:105605, 2024.

Qingwang Zhang and Yingying Zhu. Breaking rectangular shackles: Cross-view object segmentation for fine-grained object geo-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8197–8206, October 2025.

Xiaohan Zhang, Si-Yuan Cao, Zhu Yu, Zhe Wu, Xue Zhang, Xiaokai Bai, Beinan Yu, and Hui-Liang Shen. GeoFormer: Boosting object distinguishing and prompt understanding for cross-view object geo-localization. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–16, 2025a.

Xiaohan Zhang, Si-Yuan Cao, Xiaokai Bai, Yiming Li, Zhangkai Shen, Zhe Wu, Lun Luo, Qi Ming, Xiaoxi Hu, and Hui-Liang Shen. Recurrent cross-view object geo-localization. In *European Conference on Computer Vision*, 2026a. To appear.

Xiaohan Zhang, Zhangkai Shen, Si-Yuan Cao, Xiaokai Bai, Yiming Li, Zheheng Han, Zhe Wu, Qi Ming, and Hui-Liang Shen. Learning better UAV-based cross-view object geo-localization from multi-modal prompts: MoP-UAV benchmark and MoPT framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(15):12843–12851, Mar. 2026b.

Zehao Zhang, Yuqi Sun, Lei Ding, Huayu Zhang, Yufeng Wang, and Wenrui Ding. HALO: Hierarchical adaptive feature learning and cross-view interaction for object-level geo-localization. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–15, 2025b.

Weicheng Zheng, Xiaofei Mao, Nanfei Ye, Pengxiang Li, Kun Zhan, Xianpeng Lang, and Hang Zhao. DriveAgent-R1: Advancing VLM-based autonomous driving with active perception and hybrid thinking. In *International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=r2g8TV4nJy`.

Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1395–1403, 2020.

APPENDIX

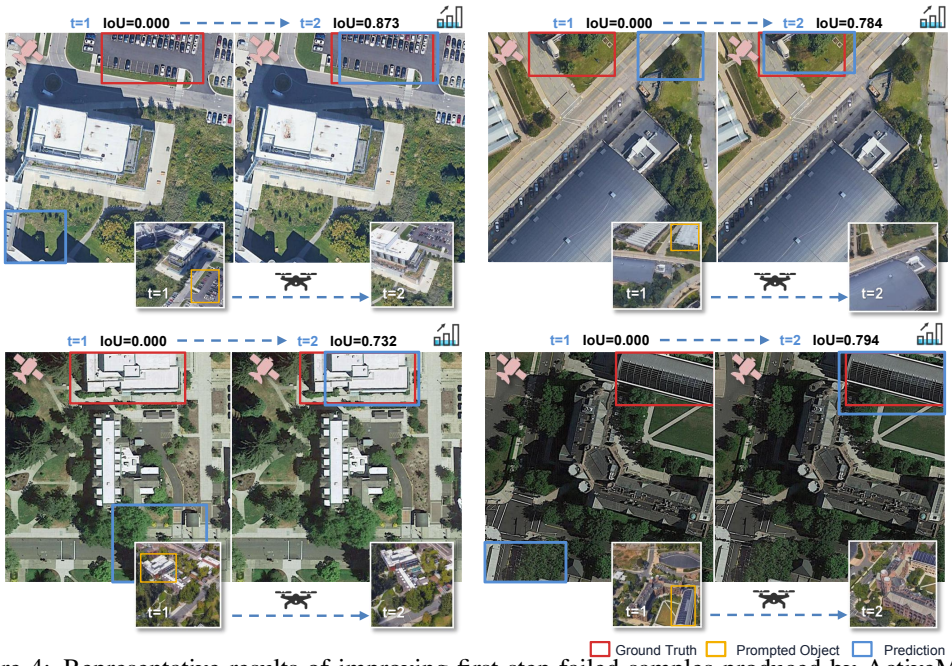

Figure 4: Representative results of improving first-step failed samples produced by ActiveMoPT. Notably, at $t = 1$, before acquiring additional UAV views, the localizer of ActiveMoPT is identical to MoPT (Zhang et al., 2026b). Please refer to the zoomed-in view for better visualization.



Figure 5: Representative results produced by ActiveMoPT. Notably, at $t = 1$, before acquiring additional UAV views, the localizer of ActiveMoPT is identical to MoPT (Zhang et al., 2026b). Please refer to zoomed-in view for better visualization.

A Implementation Details

Multi-view prompt-preserving adaptation. ActiveMoPT initializes its localizer from MoPT (Zhang et al., 2026b), with DINOv2-Base and CLIP ViT-L/14 (Radford et al., 2021) as the visual and language encoders. Images are resized to 512×512, and the localizer uses 256-dimensional states and 100 object queries. We train this stage for 24 epochs with 2-4 UAV observations per trajectory and a learning rate of 5×10^{-5}. A randomly sampled non-empty subset of **P**, **B**, and **L** is fixed throughout each trajectory. For $\mathcal{L}_{\mathrm{PI}}$, we set the localization success threshold to $\tau = 0.5$.

Trajectory-guided policy initialization. We freeze the adapted localizer and train the 770K-parameter policy for 24 SFT epochs with a learning rate of 10^{-4}. The view head scores 54 candidate viewpoints, while the stop head predicts whether another observation is needed. During oracle trajectory generation, the oracle outputs STOP when the localization IoU exceeds 0.70. Otherwise, it selects the unseen viewpoint with the largest localization gain while this gain exceeds 0.015, and stops when no remaining viewpoint satisfies this condition. Each oracle trajectory contains at most 7 observations. The SFT checkpoint with the best closed-loop performance on MoP-UAV (Zhang et al., 2026b) is used to initialize the following GRPO stage.

Cost-aware policy refinement. Starting from the selected SFT policy, we keep the localizer frozen and optimize only the policy for 2 GRPO epochs. Each episode uses groups of 32 sampled trajectories. The policy learning rate is 10^{-5}, the clipping radius is $\epsilon_{\mathrm{c}} = 0.2$, and the KL coefficient is $\beta = 0.02$. We set $\lambda = 0.01$.

Viewpoint representation. Following the viewpoint setting of University-1652 (Zheng et al., 2020), each satellite image is associated with 54 predefined UAV viewpoints. These viewpoints are sampled along three circular trajectories around the target from higher to lower elevations, resulting in 54 discrete viewpoint IDs. We therefore formulate viewpoint selection as a 54-way classification problem. Each ID corresponds to one fixed UAV viewpoint, and the view head predicts a score over the 54 IDs. At each step, viewpoints that have already been observed are masked out, and the highest-scoring remaining ID is selected when the policy decides to continue.

B More Experimental Results

B.1 Qualitative Results

(1) *ActiveMoPT learns to select UAV views that are easier to match with the satellite image* (Fig. 4). In these representative cases, the second UAV view selected by ActiveMoPT is often more consistent with the satellite image in object orientation and surrounding spatial layout than the initial view. This reduces the cross-view discrepancy and makes the target easier to match, allowing several first-step failures with zero IoU to be corrected after acquiring the new view. Although no explicit alignment objective is introduced, the learned policy shows an implicit tendency to select viewpoints that are easier to register with the satellite image. This indicates that ActiveMoPT does not simply acquire more observations, but actively searches for viewpoints that are more favorable for cross-view localization.

(2) *ActiveMoPT progressively improves localization* (Fig. 5). The examples show that newly selected views can continuously refine the localization. In the first case, the prediction remains inaccurate after the second view but is corrected after the third view, with IoU increasing from 0.000 to 0.033 and then to 0.608. In the second case, the first-step prediction is already successful, yet additional views further improve the IoU from 0.535 to 0.699 and 0.708. These examples show that different samples require different amounts of visual evidence, and that ActiveMoPT can progressively accumulate information from multiple observations to refine the target localization.

B.2 More Ablation Results

Effect of the observation-history embedding. Table 4 evaluates whether the action policy needs explicit information about previously observed viewpoints. The observation-history embedding (Fig. 2(b)) provides a binary view-status representation, indicating which candidate viewpoints have already been observed and which remain unseen. Removing this input reduces localization performance while also decreasing the average number of acquired views. This suggests that, without

Table 4: Ablation of the input of action policy on MoP-UAV (Zhang et al., 2026b) dataset using prompt **P+B+L**. Best results are shown in **bold red**

Input of the Action Policy	mIoU ↑	Acc@0.25 ↑	Acc@0.50 ↑	Avg. views ↓
w/ Embedding of the Observation History	**0.414**	**0.659**	**0.478**	1.450
w/o Embedding of the Observation History	0.411	0.657	0.470	**1.304**

explicit observation history, the policy tends to stop earlier because it cannot fully distinguish how much viewpoint coverage has already been obtained from the current state alone. In contrast, providing the history embedding allows the policy to condition its stopping and viewpoint-selection decisions on the accumulated observation process, leading to more informative additional acquisitions and better localization. These results show that the observation history is not merely auxiliary state information, but an important cue for active decision making.

Table 5: Ablation of $\Delta\rho_t$ in the step reward r_t on MoP-UAV (Zhang et al., 2026b) dataset using **P+B+L**. Best results are shown in **bold red**

Reward Setting	mIoU ↑	Acc@0.25 ↑	Acc@0.50 ↑	Avg. views ↓
w/ $\Delta\rho_t$	**0.414**	**0.659**	**0.478**	1.450
w/o $\Delta\rho_t$	0.411	0.654	0.473	**1.235**

Effect of the $\Delta\rho_t$ in r_t. Table 5 evaluates the effect of $\Delta\rho_t$ when both the current best IoU and the new IoU are zero. In this case, IoU provides no information about whether the new UAV view improves the prediction. Without the $\Delta\rho_t$ term, acquiring an additional view receives only the acquisition cost, which encourages the policy to stop earlier even when the prediction is moving toward the correct direction. This is reflected by the lower average number of views but also worse localization performance. By introducing $\Delta\rho_t$, the policy can distinguish useful progress before box overlap is achieved and continue acquiring views when they move the prediction closer to the target. The result shows that $\Delta\rho_t$ provides an important learning signal for difficult cases where IoU alone is too sparse to guide the policy.

C Discussion

C.1 Discussion of $\mathcal{L}_{\text{PI}}$

We further discuss the design of the Preserve-Improve (PI) loss $\mathcal{L}_{\text{PI}}$ and the role of each term in the $\mathcal{L}_{\text{PI}}$. In Sec. 4.1, the $\mathcal{L}_{\text{PI}}$ of sample i is defined as

$$\mathcal{L}_{\text{PI}}^{i} = \begin{cases} \max\left(0, \tau - u_T^i\right), & u_1^i \geq \tau, \\ \max\left(0, u_1^i - u_T^i\right) + \max\left(0, \tau - u_T^i\right), & u_1^i < \tau, \end{cases} \tag{15}$$

where u_1^i denotes the localization IoU between predicted boxes and ground-truth boxes at the first step using only the initial UAV view, and u_T^i denotes the final IoU after incorporating additional views. We regard the first-step localization as successful if $u_1^i \geq \tau$, and as failed otherwise.

The $\mathcal{L}_{\text{PI}}$ treats these two cases differently. For an initially successful sample, we require the final prediction to remain above the success threshold:

$$\max\left(0, \tau - u_T^i\right). \tag{16}$$

This prevents additional UAV views from turning an already successful localization into a failed one. If the first-step prediction fails ($u_1^i < \tau$), two terms are used:

$$\max\left(0, u_1^i - u_T^i\right) + \max\left(0, \tau - u_T^i\right). \tag{17}$$

The first term $\max\left(0, u_1^i - u_T^i\right)$ becomes non-zero only when the final IoU is lower than the first-step IoU, and therefore discourages new views from making the localization worse. The second term $\max\left(0, \tau - u_T^i\right)$ remains non-zero as long as the final IoU is below τ, encouraging the model to continue improving the prediction until it reaches the success threshold. For example, if a sample

improves from $u_1^i = 0.3$ to $u_T^i = 0.4$ with $\tau = 0.5$, the first term becomes zero because the localization has improved, while the second term remains active because the prediction is still unsuccessful. If the final IoU reaches 0.5 or higher, both terms become zero.

Weighted $\mathcal{L}_{\mathrm{PI}}$. To validate the contribution of these three terms, we assign a separate weight to each of them and define a weighted $\mathcal{L}_{\mathrm{PI}}$ as

$$\mathcal{L}_{\mathrm{PI}}^i = \begin{cases} w_{\mathrm{s}} \max\left(0, \tau - u_T^i\right), & u_1^i \geq \tau, \\ w_{\mathrm{m}} \max\left(0, u_1^i - u_T^i\right) + w_{\mathrm{r}} \max\left(0, \tau - u_T^i\right), & u_1^i < \tau. \end{cases} \tag{18}$$

Here, w_{s} controls how strongly we keep first-step successful samples above the success threshold, w_{m} controls how strongly we avoid making first-step failed samples worse, and w_{r} controls how strongly we encourage failed samples to reach the success threshold.

Table 6 verifies these roles by changing one weight at a time from the default setting $(w_{\mathrm{s}}, w_{\mathrm{m}}, w_{\mathrm{r}}) = (3, 1, 0.5)$. Removing w_{m} causes the largest performance drop, from 0.398 to 0.382 mIoU and from 0.457 to 0.435 Acc@.50, showing that preventing harmful multi-view updates is particularly important. Setting $w_{\mathrm{r}} = 0$ also decreases performance to 0.389 mIoU and 0.445 Acc@.50, showing that merely avoiding worse predictions is insufficient. Failed samples should also be encouraged to reach the success threshold. The w_{s} ablation shows that protecting first-step successful predictions is also useful, while an excessively large weight can restrict the benefit of later views. Overall, the default weights provide the best balance among these three items.

Table 6: Ablation of $(w_{\mathrm{s}}, w_{\mathrm{m}}, w_{\mathrm{r}})$ at $\tau = 0.5$ in $\mathcal{L}_{\mathrm{PI}}$ on MoP-UAV (Zhang et al., 2026b) dataset using prompt **P+B+L**. "Mean mIoU" and "Mean Acc@.50" represent the averages of mIoU and Acc@.50 among fixed 2/3/4/Random number of UAV views. "Random" uses a sampled number of 1-4 UAV views.

Setting	w_{s}	w_{m}	w_{r}	Mean mIoU	Mean Acc@.50
Default	3	1	0.5	**0.398**	**0.457**
$w_{\mathrm{s}} = 0$	0	1	0.5	0.396	0.454
$w_{\mathrm{s}} = 1$	1	1	0.5	0.390	0.444
$w_{\mathrm{s}} = 5$	5	1	0.5	0.387	0.439
$w_{\mathrm{m}} = 0$	3	0	0.5	0.382	0.435
$w_{\mathrm{m}} = 0.5$	3	0.5	0.5	0.387	0.442
$w_{\mathrm{m}} = 2$	3	2	0.5	0.396	0.453
$w_{\mathrm{r}} = 0$	3	1	0	0.389	0.445
$w_{\mathrm{r}} = 0.25$	3	1	0.25	0.389	0.441
$w_{\mathrm{r}} = 1$	3	1	1	0.394	0.443

Shaded rows indicate the selected hyperparameters. Best results are shown in **bold red**.

C.2 Discussion of $\mathcal{L}_{\mathrm{MV}}$

In Multi-View Prompt-Preserving Adaptation (Sec. 4.1), we define $\mathcal{L}_{\mathrm{MV}}$ as

$$\mathcal{L}_{\mathrm{MV}} = \mathcal{L}_{\mathrm{Det}}^{(1)} + \mathcal{L}_{\mathrm{Det}}^{(T)} + \mathcal{L}_{\mathrm{PI}}, \tag{19}$$

where $\mathcal{L}_{\mathrm{Det}}^{(1)}$ and $\mathcal{L}_{\mathrm{Det}}^{(T)}$ are detection losses (Carion et al., 2020) and supervise the localization results at the first and final steps, respectively, and $\mathcal{L}_{\mathrm{PI}}$ constrains how localization changes after additional UAV views are incorporated. We further discuss two design choices in $\mathcal{L}_{\mathrm{MV}}$, *i.e.*, (1) why detection supervision ($\mathcal{L}_{\mathrm{Det}}^{(1)}$ and $\mathcal{L}_{\mathrm{Det}}^{(T)}$) and $\mathcal{L}_{\mathrm{MV}}$ are both needed, and (2) why detection supervision ($\mathcal{L}_{\mathrm{Det}}^{(1)}$ and $\mathcal{L}_{\mathrm{Det}}^{(T)}$) is applied only at the first ($\mathcal{L}_{\mathrm{Det}}^{(1)}$) and final steps ($\mathcal{L}_{\mathrm{Det}}^{(T)}$) rather than at every step.

C.2.1 Why Are Detection Supervision and PI Both Needed?

The two losses serve different purposes. Detection supervision ($\mathcal{L}_{\mathrm{Det}}^{(1)}$ and $\mathcal{L}_{\mathrm{Det}}^{(T)}$) directly optimizes the predicted target box, whereas $\mathcal{L}_{\mathrm{PI}}$ focuses on whether incorporating additional views leads to a desirable performance improvement in localization results. For example, once the final IoU reaches the success threshold τ, the $\mathcal{L}_{\mathrm{PI}}$ becomes zero, while the detection loss can still improve the localization performance beyond this threshold. Conversely, $\mathcal{L}_{\mathrm{PI}}$ explicitly discourages incorporating

additional UAV views from turning a good localization into a failed one or making an initially failed localization even worse.

The ablation in Table 2(a) supports this complementary design. With both $\mathrm{Attn}_{\mathrm{init}}$ and $\mathrm{Attn}_{\mathrm{mv}}$, adding $\mathcal{L}_{\mathrm{PI}}$ improves mIoU. These results show that detection supervision alone can train the localizer, while $\mathcal{L}_{\mathrm{PI}}$ further helps the model make better use of additional views.

C.2.2 Why Supervise Only the First and Final Steps?

A multi-view sequence contains several localization steps. One straightforward choice is to apply a detection loss after every new UAV view is incorporated. However, we supervise only two predictions, *i.e.*, the first-step prediction obtained from the initial UAV view, and the final prediction after all sampled UAV views have been incorporated. This design follows the role of the two endpoints in our setting. The first-step loss maintains the original single-view localization ability of MoPT (Zhang et al., 2026b), while the final-step loss directly optimizes the result after multi-view integration.

The intermediate states serve two roles, *i.e.*, producing the current localization and carrying information for subsequent view updates. Applying a detection loss at every step directly optimizes each intermediate prediction for its current localization accuracy. However, a state that is optimal for the current step is not necessarily the most suitable state for incorporating later observations. Since the recurrent updates share parameters across steps, repeatedly optimizing all intermediate predictions may therefore limit the model's ability to organize the query state for better final multi-view integration (Zhang et al., 2026a). Notably, the intermediate updates are still trained because the final loss is backpropagated through the entire recurrent update process. In addition, the number of UAV views varies across training samples, so predictions under different numbers of observations can also become final predictions and receive direct supervision.

To further validate our design, we also quantitatively compare it with supervision at every step. As shown in Table 7, supervising only the first and final steps outperforms. This shows that directly focusing supervision on the initial and final predictions better helps the model translate additional observations into localization gains.

Table 7: Ablation of supervision steps in $\mathcal{L}_{\mathrm{MV}}$ of Sec. 4.1 on MoP-UAV (Zhang et al., 2026b) dataset using prompt **P+B+L**. We report mIoU at different fixed number of input UAV views. "Random" uses a sampled number of 1-4 UAV views

Supervision steps of $\mathcal{L}_{\mathrm{MV}}$	1 view	2 views	3 views	4 views	Random
First and final steps	**0.389**	**0.398**	**0.399**	**0.400**	**0.396**
Every step	0.388	0.390	0.390	0.388	0.389

Shading indicates the selected supervision scheme of $\mathcal{L}_{\mathrm{MV}}$. The better result is shown in **bold red**.

C.3 Discussion of the Predefined Threshold in SFT Dataset Construction

During oracle trajectory construction in Sec. 4.2, the predefined threshold determines whether an additional UAV view provides sufficient localization improvement to justify continuing the trajectory. Specifically, if the largest IoU gain among the unseen viewpoints exceeds this threshold, the corresponding view is selected. Otherwise, the trajectory stops. The threshold therefore controls how aggressively the oracle acquires additional observations.

We set this threshold to 0.015 based on statistics of the training data. Setting it to 0 makes the oracle overly aggressive, since even a very small positive gain leads to another view acquisition. In contrast, a larger threshold can stop the trajectory too early and discard useful additional observations. With a threshold of 0.015, 49.32% of the constructed trajectories obtain localization improvement, whereas increasing the threshold to 0.03 reduces this proportion to 37.01%. We therefore choose 0.015 as a moderate value that retains useful multi-view trajectories without encouraging unnecessary acquisitions.

We note that a fixed threshold cannot perfectly determine the best action for every sample. This is also why SFT is used only to initialize the action policy. In the subsequent RL stage, the policy no longer follows this predefined threshold. Instead, it interacts with the frozen localizer and learns

from the gain-cost reward to determine whether the expected localization improvement is worth the cost of acquiring another view. Therefore, the final policy can further adjust the trajectories learned from SFT and reduce its dependence on the manually chosen threshold.

C.4 Discussion of the Distribution in UAV View Selection

Table 8: Distribution of the most frequently selected UAV viewpoints at the second and third steps. (a) Distribution of the UAV viewpoints selected as the second observation. (b) Distribution of the UAV viewpoints selected as the third observation.

(a) Distribution of the Second-View Selection

UAV Viewpoint ID	54	45	46	50	53	44	9	14	47
Ratio	30.44%	9.67%	7.35%	5.67%	4.95%	3.70%	3.55%	3.14%	2.48%

(b) Distribution of the Third-View Selection

UAV Viewpoint ID	45	37	53	54	44	50	41	5	38
Ratio	22.03%	15.25%	15.25%	6.78%	6.78%	5.08%	5.08%	5.08%	5.08%

Table 8 shows that ActiveMoPT exhibits clear but non-degenerate preferences when selecting additional UAV views. For the second observation, View 54 is selected most frequently. This is consistent with the qualitative results in Fig. 4, where the policy tends to favor UAV views whose orientation and spatial layout are more similar to the satellite image, making cross-view matching easier. In MoP-UAV (Zhang et al., 2026b) dataset, View 54 often provides such a favorable viewpoint, so its high selection frequency reflects a meaningful preference.

At the same time, the policy does not collapse to View 54. The remaining selections are distributed across multiple viewpoints, indicating that the preferred view depends on the current observation and localization state. A similar pattern appears at the third observation. The dominant choices shift to Views 45, 37, and 53, showing that the policy adapts its next-view selection after incorporating previous observations.

C.5 Limitations and Future Work

Our current formulation considers a discrete viewpoint space, where the view head selects among predefined viewpoint IDs through classification. This setting allows us to focus on the key challenge of introducing ActiveGeo, but it does not model continuous UAV motion between viewpoints. As an initial step toward agentic CVOGL, ActiveMoPT establishes the basic interaction loop of localization, viewpoint selection, and stopping. The next step is to extend this framework to continuous trajectory control, where the UAV jointly plans its motion and acquires informative observations during navigation.