

Memory-Augmented Multimodal Large Language Models for Small Object Understanding in Streaming Aerial Videos

Penglei Sun¹, Yehua Huang¹, Zhuoli Tao², Xiang Li¹, Runwei Guan¹, Yaoxian Song³, Kaiyong Zhao⁴, Henghui Ding⁵, Bo Han⁶ *Senior Member, IEEE*, Yang Yang¹ *Fellow, IEEE*, Xiaowen Chu¹ *Fellow, IEEE*

Abstract—Language-guided aerial perception aims to understand user-specified tiny targets in complex unmanned aerial vehicle (UAV) scenes. In real UAV deployment, the UAV must respond while it flies, so such perception runs in an online streaming manner, where frames arrive sequentially and the model responds to each one without access to future frames. However, applying current Multimodal Large Language Models (MLLMs) to this setting raises two challenges. First, targets viewed from the air are often tiny, yet the visual compression in existing MLLMs treats all regions equally and discards their fine-grained details. Second, understanding a continuous stream requires past-frame context, yet retaining the entire history is infeasible on resource-constrained onboard hardware, whereas discarding it causes the target to drift or disappear. We address the tiny object and streaming challenges from both data and method perspectives. From the data perspective, we present DroneEyes, the first pixel-level and open-vocabulary referring-segmentation dataset for tiny aerial targets, comprising 2,140 high-definition videos and 176,623 pairs across Object Description and Referring Expression tasks, with dense per-frame masks. From the method perspective, we propose SkyAnchor, an MLLM with two designs to the above challenges: a Semantics-Aware Token Router that preserves small-target under a reduced visual-token budget, and a Hierarchical Memory Bank that keeps the target consistently understood on streams. Experiments on DroneEyes show that SkyAnchor outperforms existing MLLMs on both object description (by 2× in GPT score) and referring expression (by 15.3% in region-and-contour accuracy). On the unseen SkyFind benchmark, SkyAnchor improves over the previous domain-specific SOTA by 25.9% in IoU@0.5 and 95.5% in IoU@mean on the unseen maritime test split, demonstrating strong cross-domain generalization. Finally, we deploy SkyAnchor on a real UAV platform, where our optimization pipeline achieves a 3.05× speedup over vanilla PyTorch, demonstrating its practical value for UAV surveillance and operator assistance. Our project can be seen in <https://sites.google.com/view/skyanchor/main-page>.

✉ Corresponding author.

¹The authors are with Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, Emails: {psun012, yhuang704, xli906}@connect.hkust-gz.edu.cn, {runwayrwguan, yyiot, xwchu}@hkust-gz.edu.cn.

²The author is with University of Freiburg, Freiburg, German, Emails: zhuoli.tao@students.uni-freiburg.de.

³The author is with Hangzhou City University, Hangzhou, China, Emails: songyaoxian@westlake.edu.cn.

⁴The author is with XGRIDS, Shenzhen, China, Emails: kyzhao@xgrids.com.

⁵The author is with Fudan University, Shanghai, China, Emails: hhding@fudan.edu.cn.

⁶The author is with Department of Computer Science, Hong Kong Baptist University, Hong Kong, China, Emails: bhanml@comp.hkbu.edu.hk.



Fig. 1. Comparison of DroneEyes with existing daily-life and aerial offline tasks. (a) Daily-life datasets (e.g., MeViS) feature large objects irrelevant to aerial perception. (b) Existing aerial datasets (e.g., SkyFind) target small objects using bound box assume offline access to complete clips or static images. (c) DroneEyes try to understand small objects using segmentation in streaming aerial video. The model must respond at each step without access to future frames, while accurately understanding small objects.

Index Terms—Aerial Vision, Multimodal Large Language Models, Streaming Video

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are increasingly deployed in surveillance, search-and-rescue, and infrastructure inspection [1]–[4]. To assist operators in these applications, UAVs are expected to follow natural-language instructions and ground the referred targets in streaming aerial video, e.g., “Find the white truck.” [5]. Since the UAV must respond while it flies, this perception cannot be performed offline over a complete clip, but must run in an online streaming manner. This setting poses two intrinsic challenges. First, targets viewed from the air are often tiny, occupying only a small portion of high-resolution frames [6], which demands pixel-accurate perception to separate them from cluttered background. Second, frames are captured and consumed sequentially, so the model must respond to each frame the moment it arrives without accessing future frames [7], while the resource-constrained onboard device cannot cache the entire visual history of an ever-growing stream.

Existing datasets do not provide the annotations required for this task, as illustrated in Fig. 1. First, daily-life datasets such as MeViS and LaMOT [8], [9] feature large objects that are far from the tiny targets seen from aerial viewpoints, and they draw their referring expressions from a closed set of common categories. Second, existing aerial datasets such as SkyFind and AerialMind [10], [11] capture the bird’s-eye view but offer coarse supervision, typically bounding boxes or frame-level captions, without pixel-accurate masks that delineate small targets from cluttered background.

To fill this gap, we construct DroneEyes, the **first** pixel-level and open-vocabulary referring-segmentation dataset for tiny aerial targets. It comprises 2,140 high-definition drone videos and 176,623 pairs across Object Description and Referring Expression tasks. Unlike existing aerial datasets, DroneEyes preserves all videos at their native high resolution and annotates dense per-frame segmentation masks over an open vocabulary along continuous flight videos, so that the fine-grained shape of each tiny target is delineated from the background throughout the stream. The dataset is organized into two task levels from perception to end-to-end grounding. Level 1 (L1) provides a target region and requires the model to describe what is in that region, fostering fine-grained perception of small objects. Level 2 (L2) provides a natural-language referring expression and requires the model to segment the specified target, probing end-to-end referring segmentation.

Beyond data, although recent multimodal large language models (MLLMs) have shown strong video understanding capabilities, the two challenges above also carry over to the model side when they are applied to streaming aerial video [12]. First, some targets occupy only small portions of high-resolution frames [6], leaving most visual tokens on background regions [13]. Full-token processing is accurate but costly on edge devices, while existing compression in general MLLMs is efficient but degrades tiny-object information. For instance, Qwen series [14] applies a fixed-ratio token merging that reduces all spatial regions equally. It will treat a small vehicle the same as a large building, so that small targets are merged into surrounding background tokens. Second, understanding a continuous video stream requires past-frame context, yet retaining the entire history is infeasible for streams, whereas discarding it causes the target to drift or disappear. For instance, SAM 3 [15] accumulates per-frame memory features whose storage scales linearly with sequence length. This strategy **cannot** sustain long-term identity.

To address the above challenges of MLLMs, we propose SkyAnchor, an MLLM designed for small-object understanding in streaming aerial video. For the first challenge, we introduce a Semantics-Aware Token Router that preserves target-relevant information through semantics-aware weighted aggregation under a fixed token budget. For the second challenge, we design a Hierarchical Memory Bank that decouples temporal information into two complementary layers. The Semantic Memory layer stores object-level appearance features to preserve target identity over the streaming term, while the Tracking Memory layer caches recent spatial states to support short-term mask propagation. The two layers work jointly to keep a stable understanding of the target on streams.

Experiments on DroneEyes validate the effectiveness of SkyAnchor against both general-purpose MLLMs and specialist segmentation models. On the L1 object description task, SkyAnchor achieves $2\times$ performance over the strongest general-purpose baselines across metrics and demonstrating clear advantages in small-object understanding. On the L2 referring segmentation task, SkyAnchor-3B reaches **best region-and-contour accuracy** ($\mathcal{J}\&\mathcal{F}$), surpassing the existing models by 15.3% despite using far fewer parameters. To assess cross-domain generalization, we further evaluate SkyAnchor on the SkyFind benchmark without any fine-tuning. On the unseen maritime test split, SkyAnchor surpasses the previous domain-specific SOTA by 25.9% in IoU@0.5 and 95.5% in IoU@mean, demonstrating strong robustness under aerial domain shifts. Finally, we deploy SkyAnchor on a real UAV platform with NVIDIA Jetson AGX Orin 64GB. With our optimization pipeline, the system achieves a $3.05\times$ speedup over vanilla PyTorch. It demonstrates cloud-free online monitoring on the target platform [16], [17].

In conclusion, this paper makes the following contributions:

- From the data perspective, we construct **DroneEyes**, a large-scale aerial streaming video dataset comprising 2,140 high-definition drone videos and 176,623 pairs with dense per-frame segmentation masks. To our knowledge, DroneEyes is the **first** aerial streaming understanding dataset with the pixel-level annotation.
- From the method perspective, we propose **SkyAnchor**, an MLLM that introduces a Semantics-Aware Token Router for preserving fine-grained details of tiny targets under the same budget and a Hierarchical Memory Bank for consistent understanding on streams, enabling efficient small-object understanding in streaming aerial video.
- On DroneEyes, SkyAnchor with only 3B parameters achieves state-of-the-art results on both L1 object description and L2 referring segmentation, outperforming all MLLMs. On the SkyFind benchmark without any fine-tuning, SkyAnchor achieves improvements over the previous domain-specific SOTA on the unseen maritime test split, with relative gains of 25.9% in IoU@0.5 and 95.5% in IoU@mean, demonstrating strong cross-domain generalization.
- From the deployment perspective, we develop a optimization pipeline and validate SkyAnchor on real-world surveillance videos captured by the UAV. Our optimization pipeline achieves $3.05\times$ speedup over vanilla PyTorch, confirming practical applicability for onboard inference without cloud offloading.

The rest of this paper is structured as follows. We begin with a review of related work in Section II. Section III introduces the DroneEyes dataset with the construction engine. Section IV formulates the problem and introduces the proposed framework SkyAnchor. Section V presents the experiments and analyses. Finally, Section VI concludes the paper and outlines future directions.

TABLE I

COMPARATIVE DATASET FOR VIDEO SEGMENTATION. WE COMPARE THE EXISTING VIDEO SEGMENTATION DATASET FROM VENUES. SEG. MEANS SEGMENTATION TASK. GRANULARITY INDICATES THE SPATIAL PRECISION OF THE ANNOTATIONS. PIXEL-LEVEL DENOTES DENSE PER-FRAME SEGMENTATION MASKS AND BBOX-LEVEL DENOTES BOUNDING-BOX ANNOTATIONS. EXPR. DENOTES INSTANCE LEVEL REFERRING EXPRESSIONS FOR VISUAL GROUNDING. BBOXES ARE DETECTION ANNOTATIONS WITHOUT LANGUAGE. QA PAIRS SERVE VIDEO QUESTION ANSWERING. AVG O/I RATIO IS THE ARITHMETIC MEAN OF $(area_{object}/area_{image}) \times 100\%$ ACROSS ALL ANNOTATIONS IN THE DATASET.

Dataset	Venue	Aerial Perspective	Granularity	Modality	Open Vocab.	Tiny Target	Avg O/I Ratio	Tasks		Videos	Expression	Resolution
								Seg.	QA			
Ref-Youtube-VOS [18]	ECCV 2020		Pixel-Level	Video	✓		18.31 %	✓		3,978	15K Expr.	≤ 2.7K
ShareGPT4Video [19]	NIPS 2024		Frame-Level	Video	✓		-		✓	40K	40K Captions	≤ 720p
MeViS [20]	TPAMI 2025		Pixel-Level	Video	✓		1.4%	✓		2,006	33K Expr.	≤ 720p
JTD-UAV [21]	CVPR 2025	✓	BBox-Level	Video		✓	0.21%		✓	1,328	3K QA pairs	1080p
LaMOT [9]	ICRA 2025		BBox-Level	Video	✓	✓	0.63%		✓	62	145 Expr.	≤ 720p
AerialMind [11]	AAAI 2026	✓	BBox-Level	Video	✓	✓	0.25%			93	24.6K Expr.	≤ 720p
SkyFind [10]	TPAMI 2026	✓	BBox-Level	Image	✓	✓	0.76%			35K Images	1M Expr.	≤ 4K
DroneEyes (ours)	-	✓	Pixel-Level	Video	✓	✓	0.24%	✓	✓	2,140	176K pairs	≤ 4K

II. RELATED WORK

A. Referring Video Object Segmentation

Referring video object segmentation (RVOS) segments target objects in videos given natural language expressions. Existing methods follow two main lines. The first builds upon foundation segmentation models. VISA [22], Sa2VA [23] and UniPixel [24] uses a language model to reason on a single frame and hands the mask to SAM 2 [25] for propagation. The second pursues end-to-end architectures that jointly handle language grounding and mask prediction. VideoGLaMM [26] produces spatially grounded masks alongside language outputs. GLUS [27] and VRS-HQ [28] integrate reasoning within a single large language model. InstructSeg [29] and further unify diverse referring and segmentation tasks at the pixel level. However, these methods are designed for daily life level videos with relatively large objects and do not address the challenges specific to aerial scenarios, including small targets, dense distributions, large viewpoint changes, and the requirement for continuous streaming processing.

B. MLLMs for Aerial Understanding

MLLMs have recently been adapted to remote sensing and aerial domains. GeoChat [30] SkySenseGPT [31] and EarthGPT [32] improve perception by enriching training data or broadening input modalities to SAR and infrared. SkyFind [32] introduces a million-scale UAV referring expression benchmark and reveals that current MLLMs struggle with small-target localization under heavy background clutter, while JTD-UAV [21] combines MLLMs with tracking for joint localization and intent description in thermal sequences. Both remain limited to box-level output or static grounding without pixel-level video segmentation. Our work addresses this gap by performing referring segmentation on aerial video streams with hierarchical memory and token compression tailored for small objects at high resolution.

III. DRONEEYES DATASET

A. Data Collection and Diversity

We curate a dataset of 2,140 high-definition drone videos sourced from the open-access platform Pexels¹. Unlike

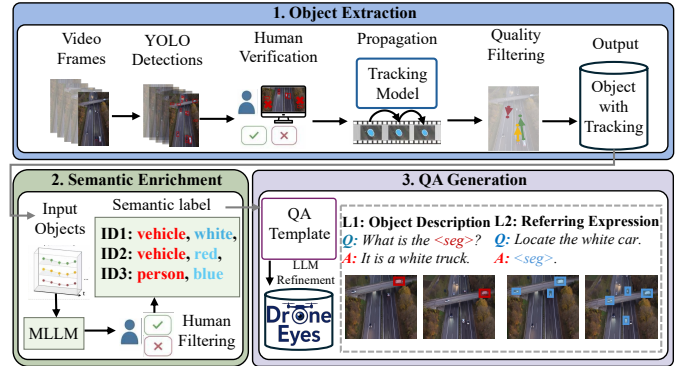


Fig. 2. The data label pipeline for DroneEyes. (1) **Object Extraction**. The candidate objects are detected with human verification, followed by mask propagation. (2) **Semantic Enrichment**. We use VLM to enrich attribute descriptions via context-aware visual prompting. (3) **QA Generation**. Two-level QA pairs are generated with LLM-based linguistic refinement.

ground-level datasets, aerial visual understanding depends on fine-grained spatial details. Therefore, we preserve all videos at their native resolutions, typically 1080p or 4K, to retain the visual cues of small objects.

To ensure data diversity, the collected videos cover various real-world topologies including urban cityscapes, highway traffic, agricultural fields, and natural terrain under different illumination conditions, altitudes, and camera trajectories such as top-down, oblique, and orbital. This collection provides a diverse distribution of common aerial targets, including vehicles, pedestrians, and infrastructure, supporting the training of grounding-capable vision-language models.

B. Data Engine

Annotating dense spatial grounding and complex understanding pairs for drone videos is labor-intensive due to the massive number of small targets. To scale up dataset construction, we develop an automated data engine that couples foundation segmentation models with large language models. The pipeline consists of three main stages: extracting consistent spatial-temporal object trajectories, and enriching these bounding boxes with semantic descriptions to construct difficulty-progressive tasks, as shown in Fig 2.

¹<https://www.pexels.com/>

1) *Object Extraction*: To establish consistent object trajectories, we implement a hybrid pipeline combining automated detection and human verification. We first utilize a YOLO-based detector to generate initial bounding box proposals on video keyframes. Human annotators then verify and filter these proposals to ensure high-precision initialization. Using these verified boxes as spatial prompts, we employ Segment Anything Model 3 as our foundation tracker. The tracker propagates segmentation masks bidirectionally through the video to maintain consistent object identities. To construct the final ground truth, we enforce a strict quality control mechanism by retaining only the tracking outputs with high confidence scores. A final manual sampling inspection guarantees the overall accuracy and reliability of the generated annotations.

2) *Semantic Enrichment*: Raw spatial trajectories lack semantic information. We leverage a Vision-Language Model to generate descriptive attributes for each tracked object. To overcome the recognition challenge of small aerial targets, we apply context-aware visual prompting. Specifically, we extract object regions with expanded contextual boundaries and apply explicit visual markers. This spatial guidance allows the MLLM to accurately produce category labels and fine-grained visual details. The resulting data bridges the gap between geometric tracking and semantic understanding.

3) *Data Generation*: Aerial visual reasoning requires understanding objects at multiple scales, from identifying the attributes of a single tiny vehicle to inferring the functional layout of an entire intersection. To systematically evaluate and train these capabilities, we build upon the semantically enriched annotations to design a two level generation strategy. This strategy requires complex visual understanding:

- **Level 1 (L1): Object Description.** The model identifies an object specified by a bounding box region and describes its visual attributes.
- **Level 2 (L2): Referring Expression.** The model locates a target object based on a natural language description and provides its exact spatial coordinates.

To further improve linguistic diversity, we apply an LLM refinement stage. A primary challenge during this text polishing process is preventing the LLM from hallucinating or modifying the original spatial coordinates. We address this issue by decoupling the coordinates from the text using an abstract placeholder mechanism during the LLM prompt stage. The exact bounding box values are restored only in the final post-processing step.

C. Data Statistics

As summarized in Table I, existing video benchmarks each cover only part of what real aerial perception requires, whether limited to a single task or relying on heavily compressed footage. Prior aerial video datasets in particular target geometric perception such as box-level detection, where annotations are closed-set category labels with bounding boxes. DroneEyes instead provides open-vocabulary, language-driven supervision. Rather than coarse bounding boxes, it offers pixel-level segmentation masks, and it is the first to pair referring segmentation with question answering on the same continuous

aerial stream, with masks and answers aligned to individual frames. In Table I, Our DroneEyes dataset has an average object-to-image ratio of only 0.24%, lower than LaMOT (0.63%) and SkyFind (0.76%), making it one of the more demanding benchmarks for small object drone visual understanding. Such pixel-level annotation is far more informative than bounding boxes for tiny targets, and together with native uncompressed streams ranging from 1080p to 4K, it is what preserves their structural details and enables accurate pixel-level mask generation in complex aerial sensing environments. DroneEyes dataset comprises **176,623** pairs generated from **2,140** valid drone videos (videos containing no detectable objects were excluded). Fig. 3 summarizes the more data statistics for DroneEyes.

IV. SKYANCHOR METHOD

SkyAnchor answers the language query and segments the target across a drone video stream with a vision-language backbone and a mask decoder (Sec. IV-A). Building on this, we design two components: a hierarchical memory bank (Sec. IV-B) and a semantics-aware token router (Sec. IV-C). We further deploy it with an edge inference schedule (Sec. IV-D). We summarize the overall procedure in Alg. 1.

A. Model Overview

1) *Task Statement*: We address the task of **online referring video object segmentation** in drone-captured aerial footage. Given a continuous video stream $\{I_t\}_{t=1}^{\infty}$ and a natural language query Q , the system must produce a segmentation mask M_t and the answer A for the query for each frame upon arrival:

$$M_t, A = \mathcal{F}(I_t, Q, S_{<t}), \quad (1)$$

where $S_{<t}$ denotes the accumulated memory state. This online, causal formulation is essential for real-world drone applications requiring low-latency streaming results.

Our model realizes $\mathcal{F}$ through a unified architecture comprising two principal components: (i) a vision-language model (VLM) backbone $\mathcal{F}_{\text{vlm}}$ that reasons over visual and textual inputs, and (ii) a mask prediction decoder $\mathcal{F}_{\text{dec}}$ that produces per-frame segmentation masks. To handle arbitrarily videos, the memory is explicitly decoupled into permanent semantic anchors S^{sem} and sliding tracking states S^{trk} (detailed in Sec. IV-B).

2) *Vision-Language Backbone*: We build upon Qwen2.5-VL. At each anchor frame t_a (triggered every N frames), the visual encoder extracts dense spatial features from the image. To prevent these from overwhelming the LLM’s attention computation, our Token Router (Sec. IV-C) dynamically compresses them into a compact visual token sequence Z_v . The query Q is tokenized into text embeddings Z_q . Additionally, a reference encoder provides spatial prompt embeddings Z_r , and the semantic memory provides appearance tokens Z_m from past anchors. The backbone jointly processes these inputs:

$$H = \mathcal{F}_{\text{vlm}}([Z_v; Z_q; Z_r; Z_m]). \quad (2)$$

Note that $\mathcal{F}_{\text{vlm}}$ runs only at anchor frames to update the semantic understanding; between anchors, mask prediction relies efficiently on the decoder’s tracking memory.

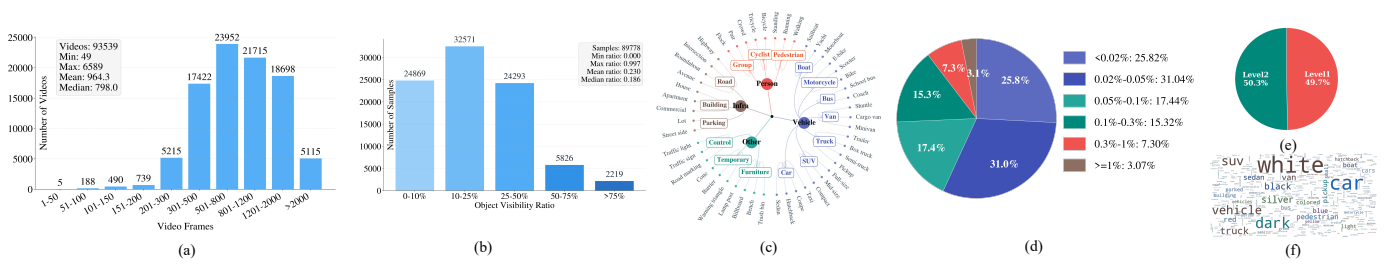


Fig. 3. The statistic of DroneEyes. (a) Distribution of video frame counts across the dataset. (b) Distribution of object visibility ratios across annotation samples. (c) Hierarchical taxonomy of object categories in the dataset. (d) Distribution of object area ratios relative to the image area. (e) Split between Level 1 and Level 2 annotations in the dataset. (f) Word cloud of query text, with word size reflecting term frequency.

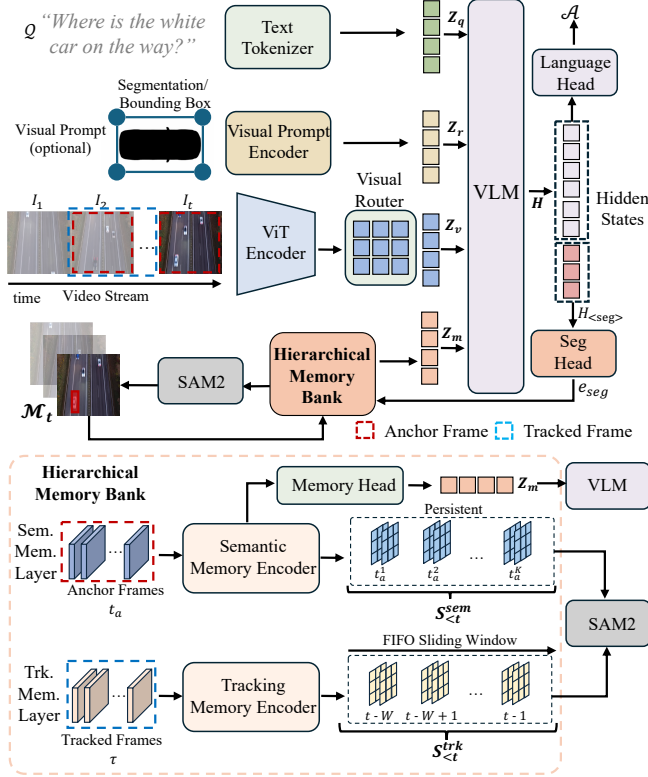


Fig. 4. Overview of SkyAnchor. (A) **Model Overview**: Q , optional visual prompt, and $\{I_1, \dots, I_t\}$ are encoded into Z_q, Z_r, Z_v ; together with Z_m , the VLM produces $\mathcal{H}$, from which the Language Head outputs $\mathcal{A}$ and the Seg Head outputs e_{seg} to drive SAM2 for M_t . (B) **Hierarchical Memory Bank**: the Semantic Memory Encoder registers persistent anchors $S_{<t}^{sem}$ from anchor frames (red dashed) and exports Z_m to the VLM via the Memory Head. The Tracking Memory Encoder caches recent frames (blue dashed) in a FIFO window $S_{<t}^{trk}$. (C) **Semantics-Aware Token Router**: the Visual Router compresses dense ViT tokens into compact Z_v by merging background tokens while retaining small-target details.

3) **Mask Prediction Decoder**: We adopt SAM2 as the decoder $\mathcal{F}_{dec}$. Internally, a segmentation head transforms the backbone’s hidden states $\mathcal{H}$ at $\langle seg \rangle$ positions into dense prompt embeddings e_{seg} . The decoder then consumes the memory from Hierarchical Memory Bank based on the e_{seg} . The Hierarchical Memory Bank produces combined spatial memory and the SAM2 decoder predict the binary mask:

$$M_t = \mathcal{F}_{dec}(S_{<t}^{sem} \cup S_{<t}^{trk}). \quad (3)$$

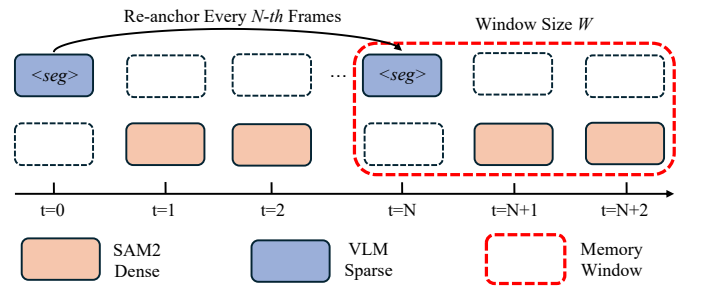


Fig. 5. Edge inference schedule of SkyAnchor. The top row is the backbone, which runs only on anchor frames, re-anchoring every N -th frame. Dashed boxes mark frames where it is skipped and the cached token is reused. The bottom row is the SAM 2 decoder (orange), which runs on every frame and propagates from the memory bank between anchors. The red dashed box is the tracking memory, a fixed-size FIFO window of size W that slides forward as each new frame enters and the oldest leaves.

By querying both the permanent semantic anchors (S^{sem}) and the recent tracking states (S^{trk}), this formulation seamlessly unifies long-term identity preservation and short-term multi-frame propagation.

B. Hierarchical Memory Bank

A fundamental challenge in online video object segmentation is maintaining object identity across temporal spans. We observe that “*what an object is*” changes slowly, while “*where it is*” changes rapidly. This motivates a two-level hierarchical memory bank equipped with a Semantic Memory Encoder and a Tracking Memory Encoder.

1) **Semantic Memory Layer**: This layer provides persistent identity information to both the VLM branch and the decoder, preventing long-term drift. Let $\{t_a^{(i)}\}_{i=1}^{K_t}$ denote all previous anchor frames before the current frame t , where $t_a^{(i)} < t$. For each anchor frame $t_a^{(i)}$, the Semantic Memory Encoder registers a semantic anchor and materialized differently in the two branches:

- **VLM-side**: When reference masks are available, object-specific appearance features are pooled and projected into a token $z_{t_a}^{sem}$. These tokens form the memory embeddings $Z_m = \{z_{t_a}^{sem} \mid t_a < t\}$ injected into Eq. (2).
- **Decoder-side**: The hidden state at t_a acts as a sparse conditioning token. The resulting anchor prediction is encoded into SAM2’s native memory format as a 2D

spatial feature $\mathbf{m}_{t_a}^{\text{sem}}$. The spatial semantic memory is defined as:

$$\mathcal{S}_{<t}^{\text{sem}} = \left\{ \mathbf{m}_{t_a^{(i)}}^{\text{sem}} \mid t_a^{(i)} < t \right\}_{i=1}^K. \quad (4)$$

These continuous representations are stored persistently as conditioning memories. Unlike discretized binary masks, they preserve rich categorical details (e.g., “silver sedan”), which is critical for resolving multi-object ambiguities during decoder-side memory attention.

2) *Tracking Memory Layer*: This layer captures fine-grained spatial continuity. Between consecutive anchor frames, SAM2 propagates masks, and the Tracking Memory Encoder writes each tracked frame τ into a 2D tracking state $\mathbf{m}_{\tau}^{\text{trk}}$. The tracking memory is defined as:

$$\mathcal{S}_{<t}^{\text{trk}} = \left\{ \mathbf{m}_{\tau}^{\text{trk}} \mid \tau \in [t - W, t) \right\}. \quad (5)$$

Unlike persistent semantic anchors, these are transient, non-conditioning memories consumed by SAM2’s memory attention before mask decoding. To bound memory consumption, they are maintained in a FIFO sliding window of size W .

C. Semantics-Aware Token Router

Processing high-resolution drone imagery requires a balance between visual detail and computational cost. We resolve this with a semantics-aware Token Router that dynamically compresses the dense visual tokens into the compact $\mathbf{Z}_v$ consumed by $\mathcal{F}_{\text{vlm}}$.

1) *Token Merging*: The Token Router partitions the dense spatial token grid into non-overlapping $k \times k$ blocks. Within each block, a lightweight scoring network evaluates the semantic salience of each raw token $\mathbf{v}_i$ to produce an importance score s_i . The block is then merged into a single output token via softmax-weighted aggregation:

$$\hat{\mathbf{z}} = \text{Proj} \left(\sum_{i \in \mathcal{B}} \text{softmax}(s_i) \cdot \mathbf{v}_i \right), \quad (6)$$

where $\text{Proj}(\cdot)$ is a learnable projection. The collection of all $\hat{\mathbf{z}}$ across blocks constitutes $\mathbf{Z}_v$. The Router uses the same compression ratio for all spatial blocks, but adjusts the aggregation weights according to their visual content.

2) *Routing Loss*: To guide the importance scores toward target-related regions, we guide it with an auxiliary loss supervised by the ground-truth segmentation mask:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{vlm}} + \lambda_1 \cdot \mathcal{L}_{\text{seg}} + \lambda_2 \cdot \mathcal{L}_{\text{route}}, \quad (7)$$

where $\mathcal{L}_{\text{route}} = \text{BCE}(\mathbf{s}, \bar{\mathbf{M}}_{\text{gt}})$, with $\mathbf{s}$ being the predicted importance scores and $\bar{\mathbf{M}}_{\text{gt}}$ the ground-truth mask downsampled to the token grid resolution.

D. Edge Deployment

We run SkyAnchor on an edge platform as shown in Fig 5. The multimodal inference runs sparsely in time, while the tracking that emits a mask on every frame stays dense and cheap, so most frames skip the heavy backbone entirely. The backbone $\mathcal{F}_{\text{vlm}}$ is invoked only on anchor frames, spaced N frames apart, where it produces the conditioning token $\mathbf{e}_{\text{seg}}$ that drives the decoder. This call is asynchronous, so the

Algorithm 1 SkyAnchor Online Inference

Require: Video stream $\{I_t\}$, query $\mathcal{Q}$, anchor interval N , FIFO size W , block size k
Ensure: Mask stream $\{\mathbf{M}_t\}$ and answer $\mathcal{A}$

```

1: Init:  $\mathcal{S}^{\text{sem}} \leftarrow \emptyset$ ,  $\mathcal{S}^{\text{trk}} \leftarrow \emptyset$  (capacity  $W$ ),  $\mathbf{e}_{\text{seg}} \leftarrow \emptyset$ 
2: for each frame  $I_t$  do
3:   if  $t \bmod N == 1$  then
4:      $\hat{\mathbf{z}} \leftarrow \text{Proj}(\sum_{i \in \mathcal{B}} \text{softmax}(s_i) \mathbf{v}_i)$ 
5:      $\mathbf{Z}_v \leftarrow \{\hat{\mathbf{z}}\}$ 
6:      $\mathbf{H} \leftarrow \mathcal{F}_{\text{vlm}}([\mathbf{Z}_v; \mathbf{Z}_q; \mathbf{Z}_r; \mathbf{Z}_m])$ 
7:      $\mathbf{e}_{\text{seg}} \leftarrow \text{SegHead}(\mathbf{H})$ 
8:      $\mathcal{S}^{\text{sem}}.append(\mathbf{m}_t^{\text{sem}})$ 
9:   end if
10:   $\mathbf{M}_t \leftarrow \mathcal{F}_{\text{dec}}(\mathcal{S}_{<t}^{\text{sem}} \cup \mathcal{S}_{<t}^{\text{trk}}, \mathbf{e}_{\text{seg}})$ 
11:   $\mathcal{S}^{\text{trk}}.append(\mathbf{m}_t^{\text{trk}})$ 
12:  if  $|\mathcal{S}^{\text{trk}}| > W$  then
13:    pop oldest from  $\mathcal{S}^{\text{trk}}$ 
14:  end if
15:  yield  $\mathbf{M}_t, \mathcal{A}$ 
16: end for
```

per-frame path keeps running on the cached token until the next one is ready and the backbone never stalls the stream. Every frame then passes through the SAM 2 path and memory attention propagates the target over the tracking memory $\mathcal{S}_{<t}^{\text{trk}}$, and the decoder $\mathcal{F}_{\text{dec}}$ combines the hierarchical memory with the cached token to emit the mask $\mathbf{M}_t$, so a coherent mask is available even while the backbone is silent. The tracking memory is a fixed-size FIFO window of capacity W , where each frame appends a new state and evicts the oldest once full. On top of this, the backbone, the SAM 2 image encoder and the mask decoder are compiled into static-shape BF16 TensorRT engines, while the per-frame memory attention is captured into a CUDA Graph that replays its many small kernels as one operation.

V. EXPERIMENT

A. Experimental Setup

1) *Datasets*: We evaluate our proposed SkyAnchor comprehensively across two distinct domains.

- For the primary dynamic video scenarios, we utilize our proposed **DroneEyes** dataset, which provides unconstrained drone trajectories and pixel-level mask annotations. The dataset is split into training, validation, and testing sets with a ratio of 8:1:1.
- To assess cross-domain generalization on static aerial imagery, we further benchmark our model on the recently released **SkyFind** [10] dataset, following its official validation and test splits. Its validation set (5,000 samples) shares the same distribution as the training set (331,364 samples), whereas its test set (16,546 samples) consists of unseen maritime scenarios and thus introduces a clear domain shift. We perform no training on SkyFind and evaluate our model directly on both the validation and test split.

TABLE II

QUANTITATIVE COMPARISON OF OBJECT DESCRIPTION AND REFERRING SMALL OBJECT SEGMENTATION PERFORMANCE ON THE DRONEEYES. EXISTING MLLM BASELINES ARE EVALUATED USING THEIR NATIVE OFFLINE INFERENCE MODE, WHEREAS SKYANCHOR PERFORMS ONLINE STREAMING INFERENCE. SEG. LLM DENOTES THE MLLM WITH SEGMENTATION CAPABILITIES. FT DENOTES THE MODEL FINETUNED ON THE DRONEEYES TRAIN SET.

Model	Type	Source	Size	L1 - Object Description			L2 - Referring Expression		
				ROUGE-L	METEOR	GPT Scores	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
Gemini-3.0	General MLLM	Closed	-	18.40	19.10	1.21	1.51	4.01	2.76
GPT-5	General MLLM	Closed	-	17.44	16.10	1.61	0.87	2.62	1.75
GPT-4o	General MLLM	Closed	-	18.91	21.11	1.42	0.75	2.04	1.39
Claude-Sonnet-4.5	General MLLM	Closed	-	16.36	20.50	1.47	0.54	1.37	0.95
Kimi-K2.5 [33]	General MLLM	Closed	-	18.13	22.40	1.43	4.14	7.85	6.00
Qwen-3.6-A3B [34]	General MLLM	Open	35B	17.61	21.80	1.46	13.12	24.63	18.88
Gemma-4	General MLLM	Open	31B	19.06	18.85	1.30	16.21	30.10	23.16
SAM3 [15]	Seg. MLLM	Open	1B	-	-	-	12.67	14.04	13.35
VISA [22]	Seg. MLLM	Open	7B	-	-	-	10.05	10.43	10.24
VideoGlaMM [26]	Seg. MLLM	Open	7B	-	-	-	11.34	15.54	13.44
InstructSeg [29]	Seg. MLLM	Open	3B	-	-	-	27.16	36.59	31.87
GLUS [27]	Seg. MLLM	Open	7B	-	-	-	20.50	25.55	23.03
VRS-HQ [28]	Seg. MLLM	Open	7B	-	-	-	27.19	36.26	31.73
Sa2VA [23]	Seg. MLLM	Open	4B	-	-	-	21.81	29.10	25.45
Sa2VA [23]	Seg. MLLM	Open	8B	-	-	-	21.36	30.20	25.78
Sa2VA [23]	Seg. MLLM	Open	26B	-	-	-	27.61	38.49	33.05
UniPixel [24]	Seg. MLLM	Open	3B	-	-	-	23.69	31.83	27.76
UniPixel [24]	Seg. MLLM	Open	7B	-	-	-	27.96	36.57	32.26
Sa2VA-FT [23]	Seg. MLLM	Open	4B	-	-	-	24.21	31.41	27.81
UniPixel-FT [24]	Seg. MLLM	Open	3B	-	-	-	25.16	33.51	29.34
InstructSeg-FT [29]	Seg. MLLM	Open	3B	-	-	-	28.61	37.76	33.19
SkyAnchor (ours)	Unified	Open	3B	45.40	46.31	3.22	30.96	45.23	38.10

2) *Evaluation Metrics*: For the DroneEyes-L1 subset, we evaluate the generated descriptions using standard natural language generation metrics, including METEOR, ROUGE-L and GPT scores. For GPT scores, we employ a GPT-based score (0–5 scale) using GPT-5.2 as the judge, which evaluates category accuracy, attribute correctness, spatial reasoning, and descriptive fluency given the original video frames as visual context as the existing setting [35]. For DroneEyes-L2 subset, we adopt the widely acknowledged standard metrics: the region similarity ($\mathcal{J}$), the contour accuracy ($\mathcal{F}$), and their average ($\mathcal{J}\&\mathcal{F}$). For referring image segmentation on SkyFind, we report the standard Intersection over Union at a threshold of 0.5 (IoU@0.5) and the mean IoU (IoU@mean), following the SkyFind [10] setting. To better characterize performance under different aggregation protocols, we report both macro-average and micro-average results. The macro-average treats the validation and test splits equally, while the micro-average weights each sample according to the split size.

3) *Implementation Details*: Our architecture is built upon the pre-trained Qwen2.5-VL (3B parameters) as the vision-language backbone and SAM 2 as the mask prediction decoder. The model is optimized using AdamW with an initial learning rate of 2×10^{-5} and a cosine annealing scheduler. The LLM backbone is fine-tuned via LoRA ($r=128$, $\alpha=256$). The training is conducted on 8 H800 GPUs with per-device batch size 4. For the Token Router, the merge size is set to 2, the maximum patch size to 960, and the auxiliary routing loss weight $\lambda_1 = \lambda_2 = 1$. The router is trained with a separate learning rate of 10^{-4} , while the SAM 2 decoder uses 5×10^{-6} . In the benchmark experiment, the models are inferenced on the H800 GPU.

4) *Comparative Methods*: To evaluate our models, we benchmark it against diverse state-of-the-art methods aligned with different tasks.

- For the Object Description task (DroneEyes Level 1 Task), we compare SkyAnchor against general MLLMs. For all baseline models, we sample frames from the input video and provide the target bounding box as normalized text coordinates $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ in the prompt, asking the model to describe the specified object.
- For the Referring Expression task (DroneEyes Level 2 Task), we evaluate spatial localization acuity against existing general and segmentation MLLMs. For general MLLMs, we provide the query and frames and ask them to output the target bounding box as normalized text coordinates $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$. Segmentation MLLMs are designed for offline video understanding and do not provide a continuous streaming inference interface. We therefore follow their offline evaluation setting. Besides, we finetune SA2VA-4B, UniPixel-3B and InstructSeg-3B on the train set of DroneEyes to compare with SkyAnchor-3B (denoted as the FT series model).
- Finally, to demonstrate our model’s generality, we benchmark on the external SkyFind dataset against visual grounding models reported following the original benchmark paper [10]. These visual grounding model are training on the training set of SkyFind.

The general MLLMs in our study include closed-source (Gemini-3.0, GPT-5, GPT-4o, Claude-Sonnet-4.5, Kimi-K2.5) and open-source (Qwen-3.6, Gemma-4) models. For segmentation MLLMs, we evaluate the multimodal segmentation

TABLE III
ABLATION STUDY ON COMPONENTS ON SKYANCHOR.

Components		L1 - Object Description			L2 - Referring Expression		
Token Router	Memory Bank	ROUGE-L	METEOR	GPT	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
✓		34.26	35.18	2.31	22.34	32.18	27.26
		37.52	38.64	2.56	24.17	35.60	29.89
	✓	40.13	41.27	2.82	28.43	41.56	35.00
✓	✓	45.40	46.31	3.22	30.96	45.23	38.10

TABLE IV
ABLATION STUDY ON MULTI-LAYER MEMORY ON SKYANCHOR.

Memory Bank		L1 - Object Description			L2 - Referring Expression		
Semantic Memory	Tracking Memory	ROUGE-L	METEOR	GPT	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
✓		37.52	38.64	2.56	24.17	35.60	29.89
		43.18	44.52	3.05	26.13	38.85	32.49
	✓	40.26	41.73	2.81	28.84	42.67	35.76
✓	✓	45.40	46.31	3.22	30.96	45.23	38.10

model SAM3, as well as MLLMs with inherent segmentation abilities (VISA, VideoGlaMM, InstructSeg, GLUS, VRS-HQ, SA2VA, and Unipixel).

5) *Training Strategy*: We adopt a progressive multi-stage training strategy. In Stages 1, we align the base vision language model backbone (Qwen2.5-VL-3B) and the segmentation projection head, and the SAM 2 decoder on a mixture of general-purpose referring segmentation datasets, establishing the model’s core ability to perform language-guided segmentation on individual frames. In Stage 2, we introduce the Token Router and train it jointly on our DroneEyes dataset. All modules are jointly optimized with learning rates.

6) *Real-world Deployment*: We deploy SkyAnchor on an NVIDIA Jetson AGX Orin 64GB for real-world edge inference. Aerial videos are collected on the HKUST(GZ) campus in Guangzhou, China, using a DJI Matrice 4E UAV equipped with a 20-MP 4/3 CMOS wide-angle camera. The drone is flying at altitudes of 45 m and 77 m.

B. Results on DroneEyes Video Dataset

Table II jointly reports the Level-1 object description and Level-2 referring segmentation results on DroneEyes.

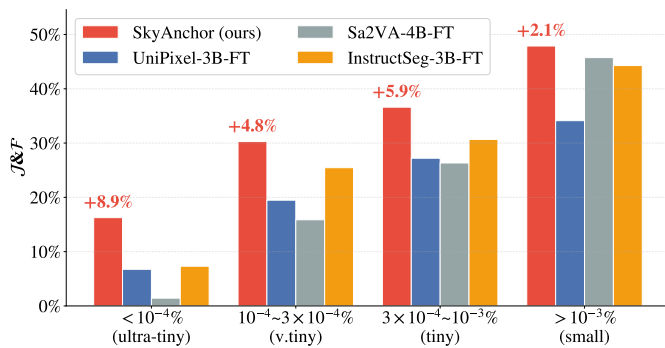


Fig. 6. Per-size $\mathcal{J}\&\mathcal{F}$ comparison. All three models are of comparable size (3B–4B) and fine-tuned on the same DroneEyes dataset. SkyAnchor outperforms three finetuned models across all size buckets, with a more margin on tiny objects. Numbers above bars denote the improvement of SkyAnchor over the best baseline.

TABLE V
COMPARISON WITH OTHER COMPONENTS USING IN EXISTING MLLMS.

Method	L1 - Object Description			L2 - Referring Expression		
	ROUGE-L	METEOR	GPT	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
<i>Token Compression</i>						
Average Pooling	37.92	38.84	2.58	25.43	37.28	31.36
Pixel-Unshuffle [36]	40.16	41.05	2.79	26.88	39.42	33.15
<i>Memory Strategy</i>						
SAM2 Native Memory [25]	42.74	43.68	3.02	27.06	39.74	33.40
StreamingVLM Memory [37]	43.55	44.47	3.10	28.12	41.18	34.65
SkyAnchor (Ours)	45.40	46.31	3.22	30.96	45.23	38.10

1) *L1 Task*: Table II reports the Level-1 object description results. SkyAnchor achieves 45.40 ROUGE-L, 46.31 METEOR, and 3.22 GPT Score with only 3B parameters, consistently outperforming all compared models. General MLLMs struggle on this task. This shows that textual coordinate prompting is insufficient for precise spatial grounding in aerial imagery, where target objects are typically small and visually similar to surrounding context. We attribute our advantage to the semantics-aware Token Router, which preserves fine-grained visual details of small objects during token compression. By dynamically assigning higher importance scores to target-relevant tokens, the Token Router retains discriminative features of small objects that would otherwise be smoothed out by uniform downsampling. This enables SkyAnchor to generate accurate attribute descriptions even for objects occupying a small portion of the aerial frame.

2) *L2 Task*: As shown in Table II, SkyAnchor (3B) achieves the best performance across all metrics, obtaining $\mathcal{J}$, $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$ scores of 30.96%, 45.23%, and 38.10%, respectively, outperforming the second-best method SA2VA (26B, $\mathcal{J}\&\mathcal{F}$ = 33.05%). General-purpose MLLMs including Gemini-3.0, GPT-5 and Claude-4.5 struggle to produce valid segmentation outputs on this task, indicating limitations of generic multi-modal models in fine-grained segmentation of small objects in aerial scenes. Among open-source segmentation-specific models, the Sa2VA series shows gradual improvement as model size increases (4B→8B→26B, $\mathcal{J}\&\mathcal{F}$ from 25.45% to 33.05%), yet even the 26B variant falls far short of the 3B SkyAnchor, demonstrating that simply scaling up parameters cannot effectively address domain adaptation challenges in aerial imagery. Overall, SkyAnchor achieves state-of-the-art performance with the smallest parameter count, fully validating the effectiveness of its targeted design for small object referring segmentation in low-altitude aerial scenarios.

C. Results on the Tiny Objects

Figure 6 compares SkyAnchor with UniPixel-3B-FT and Sa2VA-4B-FT on the DroneEyes dataset, reporting $\mathcal{J}\&\mathcal{F}$ across buckets defined by the target’s relative area. The three models have comparable parameter counts (3B–4B) and were all fine-tuned (FT) on the same dataset. SkyAnchor consistently outperforms both baselines across all size buckets, and its advantage grows as the target size decreases: in the two smallest-target intervals, ultra-tiny (relative area $< 10^{-4}\%$) and very-tiny ($10^{-4}\% \sim 3 \times 10^{-4}\%$), SkyAnchor leads the strongest baseline by 9.5 and 10.8 percentage points,

TABLE VI

QUANTITATIVE COMPARISON WITH EXISTING SOTA MODELS IN THE SKYFIND DATASET TO EVALUATE SKYANCHOR GENERATION. WE REPORT BOTH MACRO-AVERAGE AND MICRO-AVERAGE RESULTS. MACRO-AVERAGE TREATS VALIDATION AND TEST SPLITS EQUALLY, WHILE MICRO-AVERAGE WEIGHTS EACH SAMPLE ACCORDING TO THE SPLIT SIZE.

Method	IoU@0.5				IoU@mean			
	Val	Test	Macro Avg.	Micro Avg.	Val	Test	Macro Avg.	Micro Avg.
FAOA [38]	23.10	13.08	18.09	15.41	10.21	6.91	8.56	7.68
RSC [39]	29.61	17.59	23.60	20.38	16.38	9.01	12.70	10.72
RefTR [40]	31.22	22.68	26.95	24.66	15.80	13.46	14.63	14.00
TransVG [41]	35.49	22.00	28.75	25.13	20.56	11.90	16.23	13.91
VLVTG [42]	30.29	23.52	26.91	25.09	14.40	13.32	13.86	13.57
SeqTR [43]	37.49	25.74	31.62	28.47	24.45	12.57	18.51	15.33
QRNet [44]	33.90	26.21	30.05	27.99	22.10	11.22	16.66	13.74
SimREC [45]	31.17	21.50	26.34	23.74	22.09	12.15	17.12	14.46
PolyFormer [46]	42.84	31.01	36.93	33.76	25.50	16.44	20.97	18.54
AerialREC [10]	45.21	38.13	41.67	39.77	29.78	20.38	25.08	22.56
SkyAnchor(ours)	32.66	48.02	40.34	44.46	30.50	39.85	35.18	37.68

respectively. This suggests that SkyAnchor’s design offers an advantage in small-target segmentation from a drone’s perspective, rather than the improvement stemming merely from model scale or differences in fine-tuning data.

D. Ablation Studies

1) *Effectiveness of Components*: We examine the contribution of each proposed component by adding the Token Router and Memory Bank. As reported in Table III, the base model without either component yields a $\mathcal{J}\&\mathcal{F}$ of 27.26%, indicating that the plain VLM with SAM2 struggles with small object segmentation in aerial videos. Adding the Token Router alone improves $\mathcal{J}\&\mathcal{F}$ to 29.89%, as it reduces background token interference and allows the model to attend more to foreground regions. Introducing the Memory Bank alone brings a gain (35.00%), since temporal context across frames provides useful cues for locating and tracking small objects. When both components are combined, the model reaches 38.10% in L2 $\mathcal{J}\&\mathcal{F}$ metrics, with consistent improvements on both L1 description metrics. This indicates that the two modules address complementary aspects of the problem. The Token Router improves spatial focus, while the Memory Bank supplies temporal information, and their combination yields the best overall performance.

2) *Comparison with Other Components*: Table V presents a replacement-based comparison of the main components in SkyAnchor. In this table, we replace our Token Router with Average Pooling, a simple generic compression baseline, and Pixel-Unshuffle, the fixed-ratio token compression strategy used in Qwen2-VL and Qwen2.5-VL [36]. We also replace our Hierarchical Memory Bank with the native memory used in SAM2 and the streaming memory mechanism used in StreamingVLM [37]. The results show that these alternative designs all lead to lower performance than SkyAnchor. This suggests that our Token Router is more suitable for preserving small-object information during token compression, while our Hierarchical Memory Bank is more effective at maintaining temporal context across frames.

3) *Ablation on Hierarchical Memory*: We further investigate the design of the Memory Bank by ablating its two

layers: Semantic Memory and Tracking Memory. In this experiment, the Token Router is kept active in all configurations. As shown in Table IV, without any memory layer the model achieves 29.89% $\mathcal{J}\&\mathcal{F}$, relying solely on per-frame processing. Adding Semantic Memory improves the L1 description quality noticeably (GPT score from 2.56 to 3.05), as it preserves object-level semantic features that help generate more accurate descriptions. Adding Tracking Memory instead leads to a gain on L2 segmentation ($\mathcal{J}\&\mathcal{F}$ from 29.89% to 35.76%), since it maintains spatial location history that directly benefits mask prediction. The full two-layer memory achieves the best results across all metrics, confirming that Semantic Memory and Tracking Memory serve different purposes and are both needed for the complete pipeline.

4) *Memory Capacity*: As shown in Fig.7(a), $\mathcal{J}\&\mathcal{F}$ improves sharply from 20.72% (2 frames) and peaks around 16 frames, after which adding more frames brings no further gain. Fig.7(b) shows the same peak holds regardless of video length, but long videos (> 500 frames) degrade slightly beyond 16 frames due to attention dilution from stale context. This indicates that more memory is not always better: 16 frames already capture the effective temporal context.

5) *Token Routing*: We conduct an ablation study on the token routing strategy to investigate the trade-off between model effectiveness and inference efficiency. As shown in Fig. 8, we compare four configurations: no routing (retaining 100% of visual tokens), and routing with 2×2 , 3×3 , and 4×4 spatial merging grids that reduce the token count to 25%, 11.1%, and 6.25%, respectively. Without token routing, the model achieves a $\mathcal{J}\&\mathcal{F}$ of 38.12% but requires 13.38 seconds per query averagely. Introducing the 2×2 token router reduces inference time by 24.6% (from 13.38s to 10.09s) with a performance drop of 0.02 points in $\mathcal{J}\&\mathcal{F}$. The additional reduction in inference time is relatively small compared with the decrease in performance. We therefore use 2×2 merging as the default setting.

6) *Ablation on Anchor Interval*: Figure 9 evaluates the effect of the anchor interval N , which determines how often the VLM branch updates the semantic anchor. The segmentation results vary within a limited range for $N \in \{4, 8, 16, 32\}$,

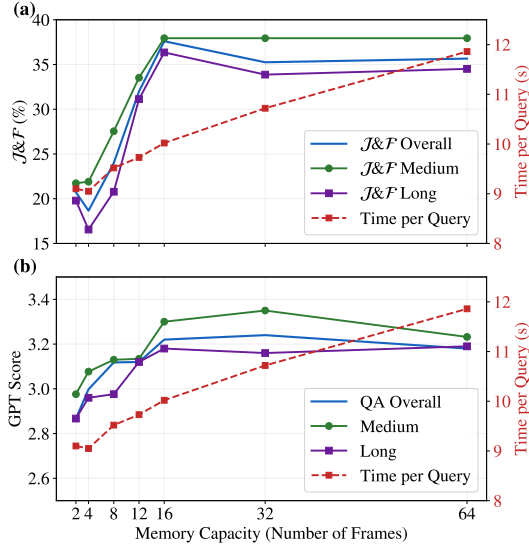


Fig. 7. Ablation study on memory capacity. (a) Segmentation quality ($J\&F$) and inference time as a function of the number of memory frames. (b) QA quality (GPT Score) and inference time as a function of the number of memory frames.

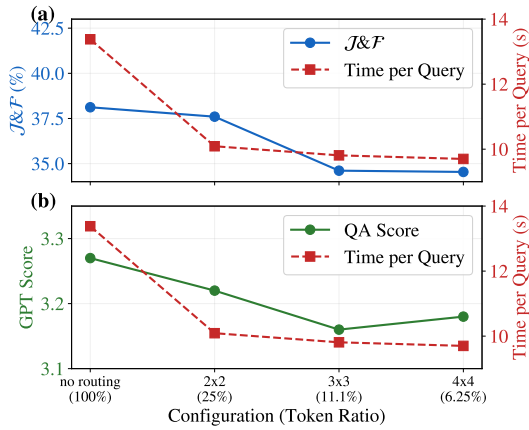


Fig. 8. Ablation study on Token Router. (a) Segmentation quality ($J\&F$) and inference time under different setting. (b) QA quality (GPT Score) and inference time under different setting.

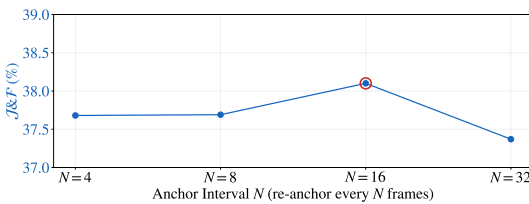


Fig. 9. Ablation study of the anchor interval N on SkyAnchor.

with the highest score obtained at $N = 16$. We use $N = 16$ as the default setting because it gives the highest measured score while avoiding frequent VLM inference.

E. Cross-Domain Generalization to Unseen Scene

We benchmark SkyAnchor-3B against SOTA methods on SkyFind to evaluate cross-domain generalization, as shown in Table VI. Unlike existing methods optimized on the SkyFind training split, SkyAnchor is evaluated without any additional training on this dataset. On the unseen maritime scenes in the test set, SkyAnchor outperforms the SOTA AerialREC on both metrics, improving IoU@0.5 from 38.13 to 48.02 and IoU@mean from 20.38 to 39.85. Under the micro-average evaluation, SkyAnchor achieves the best overall performance, obtaining 44.46 IoU@0.5 and 37.68 IoU@mean, compared with 39.77 and 22.56 for AerialREC, respectively. Under the macro-average evaluation, which assigns equal importance to the validation and test splits, SkyAnchor achieves the best IoU@mean performance (35.18), while AerialREC obtains a slightly higher IoU@0.5 score (41.67 vs. 40.34). This difference reflects the complementary strengths of the two models: AerialREC performs strongly on the in-domain validation split, whereas SkyAnchor demonstrates superior robustness under the unseen domain shift. These results indicate that models specifically optimized for the source domain may achieve stronger in-domain accuracy, while SkyAnchor provides more stable generalization when the distribution shifts.

F. Device Deployment Analysis

Table VII reports the deployment efficiency of SkyAnchor on an NVIDIA Jetson AGX Orin 64GB device paired with a DJI Matrice 4E UAV. The vanilla PyTorch implementation achieves only 1.9 FPS while consuming approximately 12.8 GB of GPU memory. TensorRT compilation increases the throughput to 3.2 FPS ($1.68\times$) and reduces memory usage to approximately 9.0 GB through FP16 kernel fusion and reduced Python-side overhead. CUDA Graph capture of the SAM2 memory-attention path further improves the throughput to 3.7 FPS ($1.95\times$), while asynchronous scheduling of the VLM and tracking branches delivers the largest gain, reaching 5.8 FPS ($3.05\times$).

The optimized system produces a complete output every approximately 172 ms, supporting cloud-free online monitoring for latency-tolerant applications such as aerial surveillance, infrastructure inspection, and operator assistance. Although it is not intended for high-speed closed-loop control or native-camera-rate processing, this throughput is consistent with aerial tracking and edge video analytics systems operating at low frame rates or invoking expensive perception modules only every few frames [16], [17]. An SAM2-only configuration reaches 6.9 FPS, or approximately 145 ms per output, indicating that the remaining VLM-related overhead is only about 27 ms, corresponding to 16% of the complete pipeline latency.

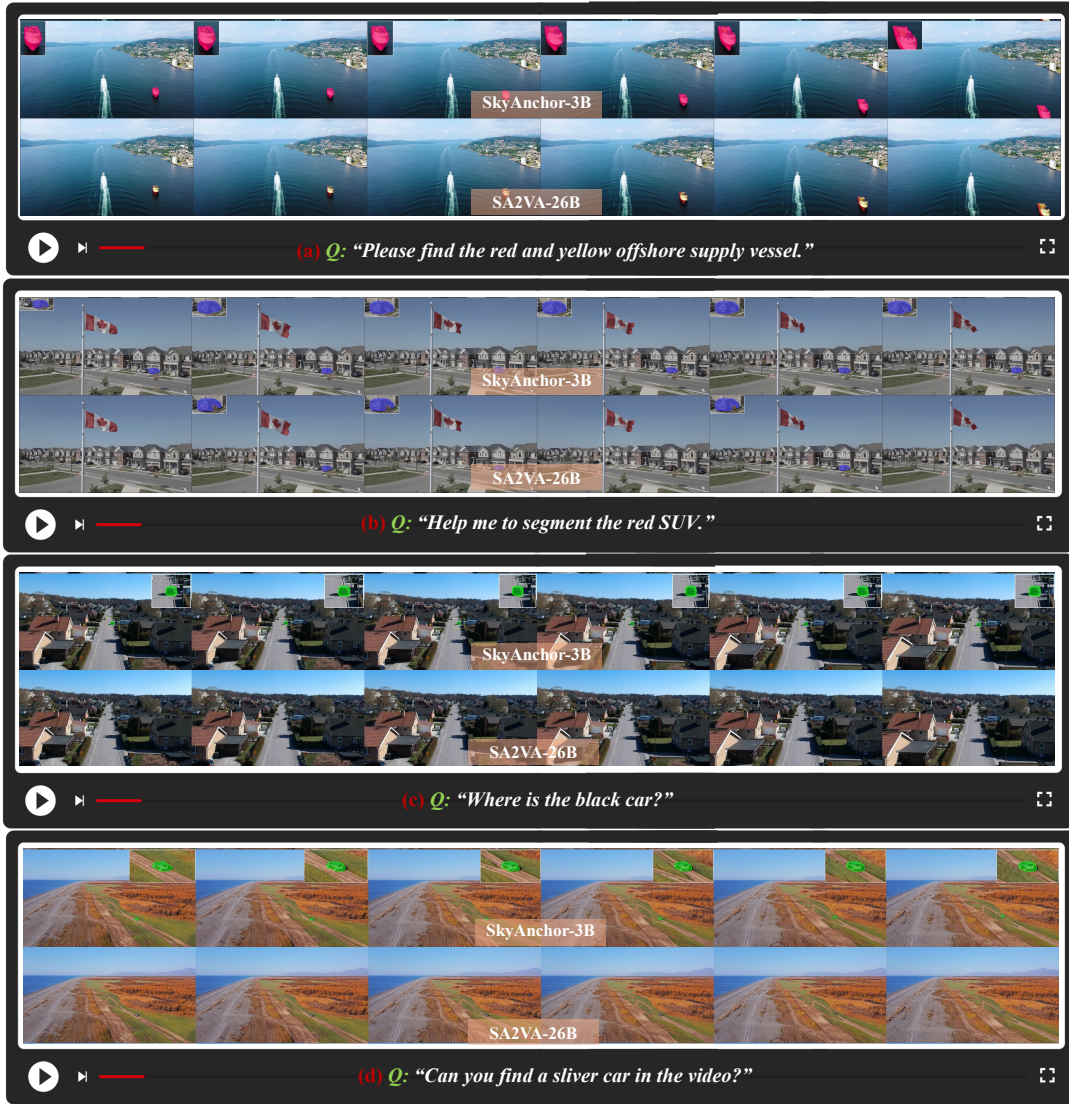


Fig. 10. Qualitative results of SkyAnchor on four challenging video sequences. The input queries are shown in the green boxes. For each case, we display the result from SA2VA-26B (the best baseline model in Table II) and ours with zoomed-in insets for better visualization of details. In case (a), (c) and (d), SA2VA-26B cannot detect the target in the video. In case (b), SA2VA-26B loses the target in some frames.

TABLE VII
ABLATION STUDY ON PROGRESSIVE OPTIMIZATION FOR SKYANCHOR DEPLOYMENT.

Configuration	FPS $\uparrow$	GPU Mem $\downarrow$	Speedup
Vanilla PyTorch (Base)	1.9	$\sim 12.8\text{G}$	1.0 $\times$
<i>TensorRT Backbone Optimization</i>			
TensorRT	3.2		1.68 $\times$
+ CUDA Graph Cache	3.7	$\sim 9.0\text{G}$	1.95 $\times$
+ Async VLM	5.8		3.05 $\times$
SAM2-Only (Upper Bound)	6.9		3.63 $\times$

G. Qualitative Results on DroneEyes

Figure 10 presents qualitative comparisons between SkyAnchor-3B and SA2VA-26B across four scenarios. SA2VA achieves the second best results in the Table II. In scenario (a), following the query “Please find the red and yellow offshore supply vessel,” SkyAnchor-3B maintains tracking

and segmentation as the vessel approaches from a distance, with relatively accurate boundaries under scale changes and textureless water surface. SA2VA-26B loses the target. For scenario (b), with the query “Help me to segment the red SUV,” SkyAnchor-3B segments the target vehicle in a residential street scene, distinguishing it from other parked cars without being affected by roadside trees and flags. SA2VA-26B detects the car in some frames but loses it in others. We attribute it to the memory usage in our SkyAnchor-3B. In scenario (c), under the aerial-view query “Where is the black car?”, SkyAnchor-3B localizes the small target spanning only a few pixels, without drifting to dark distractors such as shadows or roof textures. SA2VA-26B misses the vehicle in this bird’s-eye view. In scenario (d), answering “Can you find a silver car in the video?”, SkyAnchor-3B separates the distant silver car from the visually similar gravel road in an open landscape, showing some ability to recognize color and texture differences. SA2VA-26B fails to identify the target in this

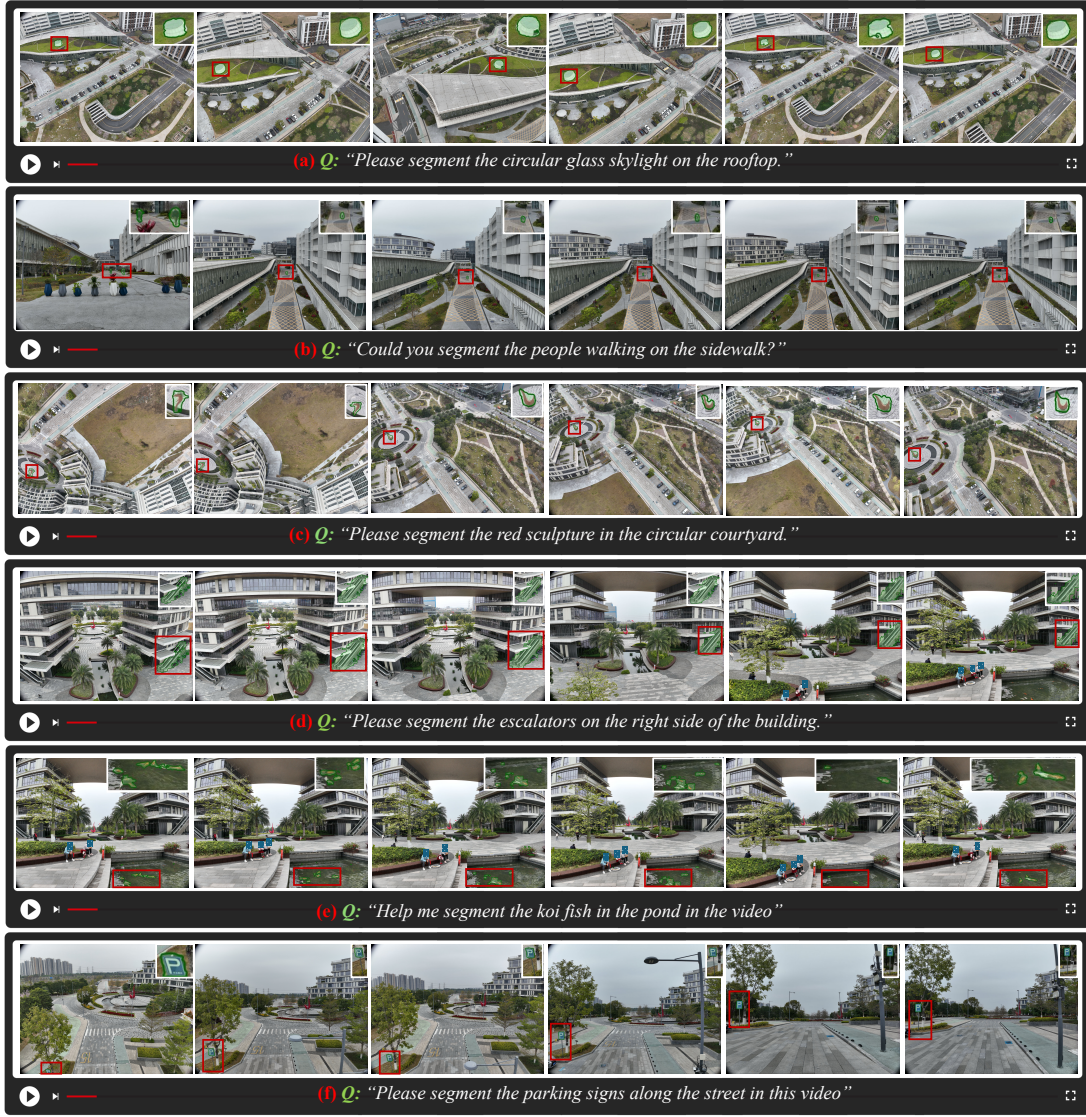


Fig. 11. Qualitative results of SkyAnchor for the real world deployment. The six queries are absent in the DroneEyes dataset. For privacy protection purposes, facial features in the video footage have been anonymized.

low-contrast environment. Across these examples, SkyAnchor shows three clear advantages. First, it maintains more stable tracking under large scale changes and long-range motion, as shown in the ship and red SUV cases. Second, it is more reliable for small targets, such as the black car that occupies only a few pixels in the aerial view. Third, it is less likely to drift to visually similar distractors, including shadows, roof textures, and low-contrast background regions.

H. Qualitative Results on Realworld Deployment

Figure 11 shows the online segmentation results across different campus environments, guided by natural language queries². The test cases cover six queries absent in the DroneEyes. First, UAV maneuvers introduce continuous view-point changes: in cases (a) and (c), the circular skylight and sculpture are observed from varying oblique angles throughout

the flight, yet the hierarchical memory bank preserves their structural semantics and keeps the masks stable without drifting to surrounding regions. Second, target scale varies dramatically across scenes. Case (d) involves large architectural structures such as building-side escalators, while cases (e) and (f) push to the opposite extreme where koi fish and street parking signs occupy only a handful of pixels. The semantics-aware Token Router selectively retains fine-grained cues for small targets while compressing uninformative background regions. Third, complex urban backgrounds introduce severe clutter and distractors. In case (b), pedestrians must be separated from standing plants using only the motion attribute in the query, and in case (c), the red sculpture must be grounded among other red-colored elements via a compositional spatial phrase. The MLLM leverages the full language query, including attribute, category, and spatial modifiers, to suppress distractors and maintain tracking continuity across frames. Across all six scenarios, SkyAnchor produces temporally coherent masks

²More video visualization demos can be seen in our website: <https://sites.google.com/view/skyanchor/main-page>

without any per-clip fine-tuning, demonstrating strong cross-domain robustness in real deployment. These deployment results indicate that the model generalizes to different object categories and flight conditions.

VI. CONCLUSION

This paper present DroneEyes, the [first](#) large-scale aerial video referring segmentation dataset, and SkyAnchor, a memory-augmented MLLM framework for online referring segmentation in UAV video streams. DroneEyes addresses the absence of high-resolution aerial video benchmarks with dense pixel-level annotations by providing 2,140 drone videos and 176,623 grounding QA pairs organized into perception-description and referring-segmentation task levels. SkyAnchor tackles two core challenges of this domain: the Semantics-aware Token Router mitigates the extreme foreground-background imbalance by compressing irrelevant background tokens while preserving fine-grained target representations, and the Hierarchical Memory Bank decouples long-term semantic identity from short-term spatial tracking states. Extensive experiments show that SkyAnchor achieves state-of-the-art performance on both DroneEyes and the cross-domain SkyFind benchmark without fine-tuning. Furthermore, we deploy SkyAnchor on the edge platform, achieving streaming inference without cloud offloading. Real-world experiments on campus surveillance videos validate the practical viability of the entire pipeline for online deployment.

Future work will pursue further model compression and hardware-aware optimization, enabling the pipeline to better match the native capture rate under fast target or platform motion. Beyond efficiency, we will explore integrating active camera control and autonomous navigation with the segmentation pipeline, where the referring-segmentation output directly drives UAV attitude adjustment and path planning to adaptively keep the target within the field of view under an optimal scale and viewpoint. We further plan to extend the framework toward multi-target referring as well as compositional queries involving relational and long-range temporal reasoning.

REFERENCES

- [1] X. Huang, “The small-drone revolution is coming—scientists need to ensure it will be safe,” *Nature*, vol. 637, no. 8044, pp. 29–30, 2025.
- [2] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, “Champion-level drone racing using deep reinforcement learning,” *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.
- [3] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, “Video object segmentation without temporal information,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1515–1530, 2018.
- [4] P. Sun, Y. Song, X. Liu, X. Yang, Q. Wang, T. Li, Y. Yang, and X. Chu, “3d question answering for city scene understanding,” in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 2156–2165.
- [5] J. Weyler, F. Magistri, E. Marks, Y. L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, and J. Behley, “Phenobench: A large dataset and benchmarks for semantic image interpretation in the agricultural domain,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 12, pp. 9583–9594, 2024.
- [6] A. Rozantsev, V. Lepetit, and P. Fua, “Detecting flying objects using a single moving camera,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 5, pp. 879–892, 2016.
- [7] B. A. Plummer, K. J. Shih, Y. Li, K. Xu, S. Lazebnik, S. Sclaroff, and K. Saenko, “Revisiting image-language networks for open-ended phrase detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 4, pp. 2155–2167, 2020.
- [8] H. Ding, C. Liu, S. He, X. Jiang, and C. C. Loy, “Mevis: A large-scale benchmark for video segmentation with motion expressions,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 2694–2703.
- [9] Y. Li, X. Liu, L. Liu, H. Fan, and L. Zhang, “Lamot: Language-guided multi-object tracking,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 6816–6822.
- [10] K. Wang, G. Wu, X. Fu, X. Wang, K. Liu, X. Lu, C. Ge, W. Zhai, and Z. Zha, “Skyfind: A large-scale benchmark unveiling referring expression comprehension for uav,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.
- [11] C. Chen, S. Liang, R. Guan, X. Sun, H. Zhao, H. Jiang, T. Huang, H. Ding, and Q.-L. Han, “Aerialmind: Towards referring multi-object tracking in uav scenarios,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 4, 2026, pp. 2805–2813.
- [12] J. Liu, H. Ding, S. He, and Y.-G. Jiang, “Event-aware instructed assistant for referring video segmentation,” *IEEE Transactions on Image Processing*, 2026.
- [13] P. Sun, Y. Song, X. Zhu, X. Liu, Q. Wang, Y. Liu, C. Xia, T. Li, Y. Yang, and X. Chu, “City-vlm: Towards multidomain perception scene understanding via multimodal incomplete learning,” in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 3448–3457.
- [14] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, “Qwen2.5 technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2412.15115>
- [15] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang *et al.*, “Sam 3: Segment anything with concepts,” *arXiv preprint arXiv:2511.16719*, 2025.
- [16] A. Ghosh, S. Iyengar, S. Lee, A. Rathore, and V. N. Padmanabhan, “React: Streaming video analytics on the edge with asynchronous cloud support,” in *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*, 2023, pp. 222–235.
- [17] T. Yang, D. Li, Y. Bai, F. Zhang, S. Li, M. Wang, Z. Zhang, and J. Li, “Multiple-object-tracking algorithm based on dense trajectory voting in aerial videos,” *Remote Sensing*, vol. 11, no. 19, p. 2278, 2019.
- [18] S. Seo, J.-Y. Lee, and B. Han, “Urvos: Unified referring video object segmentation network with a large-scale benchmark,” in *European conference on computer vision*. Springer, 2020, pp. 208–223.
- [19] L. Chen, X. Wei, J. Li, X. Dong, P. Zhang, Y. Zang, Z. Chen, H. Duan, B. Lin, Z. Tang *et al.*, “Sharegpt4video: Improving video understanding and generation with better captions,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 19472–19495, 2024.
- [20] H. Ding, C. Liu, S. He, K. Ying, X. Jiang, C. C. Loy, and Y.-G. Jiang, “Mevis: A multi-modal dataset for referring motion expression video segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [21] Y. Wang, J. Zhao, Z. Fan, X. Zhang, X. Wu, Y. Zhang, L. Jin, X. Li, G. Wang, M. Jia *et al.*, “Jtd-uav: Mllm-enhanced joint tracking and description framework for anti-uav systems,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1633–1644.
- [22] C. Yan, H. Wang, S. Yan, X. Jiang, Y. Hu, G. Kang, W. Xie, and E. Gavves, “Visa: Reasoning video object segmentation via large language models,” in *European Conference on Computer Vision*. Springer, 2024, pp. 98–115.
- [23] H. Yuan, X. Li, T. Zhang, Y. Sun, Z. Huang, S. Xu, S. Ji, Y. Tong, L. Qi, J. Feng *et al.*, “Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos,” *arXiv preprint arXiv:2501.04001*, 2025.
- [24] Y. Liu, Z. Ma, J. Pu, Z. Qi, Y. Wu, Y. Shan, and C. Chen, “Unipixel: Unified object referring and segmentation for pixel-level visual reasoning,” *Advances in Neural Information Processing Systems*, vol. 38, pp. 126 078–126 108, 2026.
- [25] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, “Sam 2: Segment anything in images and videos,” *arXiv preprint arXiv:2408.00714*, 2024.
- [26] S. Munasinghe, H. Gani, W. Zhu, J. Cao, E. Xing, F. S. Khan, and S. Khan, “Videoglamm: A large multimodal model for pixel-level visual

- grounding in videos,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 19 036–19 046.
- [27] L. Lin, X. Yu, Z. Pang, and Y.-X. Wang, “Glus: Global-local reasoning unified into a single large language model for video segmentation,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 8658–8667.
- [28] S. Gong, Y. Zhuge, L. Zhang, Z. Yang, P. Zhang, and H. Lu, “The devil is in temporal token: High quality video reasoning segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 29 183–29 192.
- [29] C. Wei, Y. Zhong, H. Tan, Y. Zeng, Y. Liu, H. Wang, and Y. Yang, “Instructseg: Unifying instructed visual segmentation with multi-modal large language models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 20 193–20 203.
- [30] K. Kuckreja, M. S. Danish, M. Naseer, A. Das, S. Khan, and F. S. Khan, “Geochat: Grounded large vision-language model for remote sensing,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 27 831–27 840.
- [31] J. Luo, Z. Pang, Y. Zhang, T. Wang, L. Wang, B. Dang, J. Lao, J. Wang, J. Chen, Y. Tan *et al.*, “Skysensegpt: A fine-grained instruction tuning dataset and model for remote sensing vision-language understanding,” *arXiv preprint arXiv:2406.10100*, 2024.
- [32] W. Zhang, M. Cai, T. Zhang, Y. Zhuang, and X. Mao, “Earthgpt: A universal multimodal large language model for multisensor image comprehension in remote sensing domain,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–20, 2024.
- [33] K. Team, T. Bai, Y. Bai, Y. Bao, S. Cai, Y. Cao, Y. Charles, H. Che, C. Chen, G. Chen *et al.*, “Kimi k2. 5: Visual agentic intelligence,” *arXiv preprint arXiv:2602.02276*, 2026.
- [34] Qwen Team, “Qwen3.6-Plus: Towards real world agents,” April 2026. [Online]. Available: <https://qwen.ai/blog?id=qwen3.6>
- [35] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.
- [36] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, “Qwen2.5-vl technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2502.13923>
- [37] R. Xu, G. Xiao, Y. Chen, L. He, K. Peng, Y. Lu, and S. Han, “Streamingvlm: Real-time understanding for infinite video streams,” *arXiv preprint arXiv:2510.09608*, 2025.
- [38] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo, “A fast and accurate one-stage approach to visual grounding,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4683–4693.
- [39] Z. Yang, T. Chen, L. Wang, and J. Luo, “Improving one-stage visual grounding by recursive sub-query construction,” in *European conference on computer vision*. Springer, 2020, pp. 387–404.
- [40] M. Li and L. Sigal, “Referring transformer: A one-step approach to multi-task visual grounding,” *Advances in neural information processing systems*, vol. 34, pp. 19 652–19 664, 2021.
- [41] J. Deng, Z. Yang, T. Chen, W. Zhou, and H. Li, “Transvg: End-to-end visual grounding with transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1769–1779.
- [42] L. Yang, Y. Xu, C. Yuan, W. Liu, B. Li, and W. Hu, “Improving visual grounding with visual-linguistic verification and iterative reasoning,” in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 9499–9508.
- [43] C. Zhu, Y. Zhou, Y. Shen, G. Luo, X. Pan, M. Lin, C. Chen, L. Cao, X. Sun, and R. Ji, “Seqtr: A simple yet universal network for visual grounding,” in *European Conference on Computer Vision*. Springer, 2022, pp. 598–615.
- [44] J. Ye, J. Tian, M. Yan, X. Yang, X. Wang, J. Zhang, L. He, and X. Lin, “Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding,” in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 15 502–15 512.
- [45] G. Luo, Y. Zhou, J. Sun, X. Sun, and R. Ji, “A survivor in the era of large-scale pretraining: An empirical study of one-stage referring expression comprehension,” *IEEE Transactions on Multimedia*, vol. 26, pp. 3689–3700, 2023.
- [46] J. Liu, H. Ding, Z. Cai, Y. Zhang, R. K. Satzoda, V. Mahadevan, and R. Manmatha, “Polyformer: Referring image segmentation as sequential polygon generation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 18 653–18 663.