MF-RSVLM: MULTI-FEATURE FUSION VISION-LANGUAGE MODEL FOR REMOTE SENSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Vision-Language Models (VLMs) have demonstrated remarkable capabilities in the natural image domain, yet their adaptation to remote sensing (RS) imagery remains challenging. However, high-resolution and multi-scale inputs alone do not ensure effective integration of complementary visual details or sustained visual guidance during language-model processing. To address these challenges, we introduce MF-RSVLM, a multi-feature fusion vision–language model for RS understanding. Specifically, we extract complementary local features across spatial scales and encoder depths, organizing them into aligned detail representations alongside global context. To selectively integrate these details and sustain visual guidance, we use current visual states to guide adaptive fusion and regulate repeated injection at selected LLM layers, helping mitigate visual forgetting. To support unified multi-task learning, we curate a 293K-sample RS instruction corpus spanning six tasks. We evaluate MF-RSVLM across three benchmark families—image captioning, scene classification, and visual question answering. MF-RSVLM achieves state-of-the-art performance in the primary scene-classification and VQA comparisons, along with competitive image-captioning results. Complementary ablation and efficiency analyses examine the roles of detail sources, adaptive fusion, and repeated gated updates while quantifying their computational overhead. These analyses provide empirical guidance for multimodal model design through selective fusion and reuse of fine-grained visual information.

1 INTRODUCTION

Recent advancements in Large Vision-Language Models (VLMs) (Wu et al., 2023; Yin et al., 2024) have achieved remarkable success in the natural image domain. These models unify perception and reasoning, and can effectively perform various downstream tasks such as visual question answering (Lu et al., 2022), text recognition (Mathew et al., 2021; Yang et al., 2025), document understanding (Liu et al., 2024b), and grounding (Ahmadyan et al., 2021). However, remote sensing (RS) imagery poses unique challenges characterized by distinct acquisition geometries, vast spatial scale variations, and complex spectral properties (Xiao et al., 2025; Lu et al., 2025). Due to this significant domain discrepancy, general-purpose VLMs often fail to generalize effectively to the RS domain (Kuckreja et al., 2024; Soni et al., 2025; Pang et al., 2024).

To bridge this gap, research efforts have been dedicated to developing specialized Remote Sensing VLMs (RSVLMs) tailored for the domain (Pang et al., 2024; Zhan et al., 2025; Muhtar et al., 2024; Tseng et al., 2025). Specifically, several approaches (Wang et al., 2025a; Shabbir et al., 2025; Noman et al., 2025) adapt open-source frameworks by fine-tuning them on remote sensing instruction datasets. For instance, GeoChat (Kuckreja et al., 2024) builds upon LLaVA-v1.5. EarthGPT (Zhang et al., 2024) combines ViT and CNN visual backbones for multi-sensor perception, while SkyEyeGPT (Zhan et al., 2025) uses an EVA-CLIP visual encoder and a LLaMA2-Chat decoder. These domain-specific adaptations equip the models for complex scene understanding and fine-grained object localization. In addition, other studies (Chen et al., 2021; 2024b;a; 2023b; Zhang et al., 2023b; Li et al., 2023b; Hu et al., 2025) have proposed innovative modules to enhance the applicability of models in the remote sensing domain. As a unified end-to-end VLM, RSUniVLM (Liu & Lian, 2024) introduces an effective G-MOE architecture to decouple the model's perception capabilities at different visual levels. EarthDial (Soni et al., 2025) enables the processing of multi-spectral and multi-temporal remote sensing imagery through its adaptive high-resolution and data-fusion modules. Despite these

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.

advances, two complementary challenges remain: selectively integrating visual details according to the current visual states and mitigating visual forgetting during language-model processing.

Regarding the first challenge, high-resolution and multi-scale approaches such as EarthDial (Soni et al., 2025) and GeoLLaVA-8K (Wang et al., 2025a) provide richer visual inputs, but making these details available does not determine their relative contributions during fusion. Features from different spatial scales and encoder depths offer distinct sources of information, whose utility can vary with the evolving visual representation. A fixed combination cannot respond to these changes. The challenge is to adaptively weight aligned detail sources while retaining global context and exploiting the information available at higher input resolutions. The second challenge is sustaining visual guidance during language-model processing. Input-only visual conditioning relies on successive layers to retain image evidence, which may become less influential as textual reasoning accumulates (Sun et al., 2025). Deep visual injection, as explored in DeepStack (Meng et al., 2024), provides renewed access to image features, but repeated access alone does not guarantee their usefulness to the evolving representation. Effective visual grounding therefore calls for regulating the contribution of injected features according to the current visual states. *These limitations motivate repeated gated injection of visual features at selected LLM layers to mitigate visual forgetting.*

To address these challenges, we introduce a Multi-Feature Fusion Remote Sensing Vision–Language Model (MF-RSVLM) that incorporates visual details into intermediate language-model representations. Specifically, we extract local features from overlapping windows at multiple spatial scales and encoder depths, alongside a global scene representation. We organize these local features into aligned detail representations to preserve local structures and support their integration with global context. We then use the current visual states at selected LLM layers to fuse these detail representations and adjust their injection strength to help sustain visual guidance. Applying these updates across layers allows the model to reuse image details beyond the initial visual input. To support unified remote-sensing multi-task learning, we further curate a comprehensive 293K-sample instruction-following dataset spanning six core tasks. We evaluate MF-RSVLM across image captioning, scene classification, and VQA benchmarks and achieve strong performance across these tasks. Additionally, component comparisons, behavioral diagnostics, and efficiency analyses examine the roles of visual detail features, adaptive fusion, and repeated gated updates. These analyses provide empirical guidance for designing multimodal large language models that selectively integrate and reuse fine-grained visual information. Overall, the main contributions can be outlined as follows:

- We propose MF-RSVLM, a unified remote-sensing vision–language framework that integrates visual detail features into intermediate LLM processing. We curate a 293K-sample instruction-tuning corpus spanning six core tasks to support learning within this unified framework.
- We develop adaptive detail fusion guided by current visual states to integrate complementary visual cues. Repeated gated injection regulates how much detail information enters selected LLM layers, helping maintain visual guidance and mitigate visual forgetting.
- We evaluate MF-RSVLM on image captioning, scene classification, and VQA benchmarks. Component ablations assess task-dependent performance. Routing and gating diagnostics examine feature utilization, while efficiency analyses quantify accuracy–cost trade-offs.

2 Related Work

Multimodal Large Language Models. Multimodal large language models (MLLMs) integrate visual representations with language generation and reasoning (Wu et al., 2023, Yin et al., 2024, Li et al., 2023a). Their foundations include contrastive image–text alignment, as in CLIP and ALIGN (Radford et al., 2021, Jia et al., 2021), and visually conditioned generation, as in SimVLM (Wang et al., 2021). Building on cross-modal alignment, visual instruction tuning enables interactive multimodal responses (Liu et al., 2023a). Supporting richer inputs also requires balancing visual detail with computational cost. Accordingly, dynamic-resolution processing, tiling, and token compression offer complementary strategies for high-resolution perception (Bai et al., 2025a, Wang et al., 2025b, Wu et al., 2024), while unified resampling extends efficient encoding to both images and video (Yu et al., 2026). Beyond input encoding, multilevel visual injection enables additional interaction within LLM layers (Bai et al., 2025a). In parallel, reinforcement learning and routing between fast responses and deeper reasoning target multimodal reasoning capabilities (Hong et al., 2025, Singh et al., 2025).

These developments highlight the complementary roles of cross-modal alignment, efficient perception, and reasoning in multimodal model design.

Remote Sensing Vision Language Models. Remote sensing VLMs adapt visual and language representations to diverse tasks and sensor modalities (Weng et al., 2025, Xiao et al., 2025). GeoChat (Kuckreja et al., 2024) and SkyEyeGPT (Zhan et al., 2025) use instruction tuning for grounded interaction, while VHM (Pang et al., 2024) emphasizes versatile tasks and honest responses. EarthGPT (Zhang et al., 2024) and EarthDial (Soni et al., 2025) extend visual language learning to heterogeneous remote sensing inputs, and GeoLLaVA-8K (Wang et al., 2025a) uses dynamic tiling for large images. RSThinker (Liu et al., 2025) introduces perceptually grounded reasoning traces, and GeoSolver (Sun et al., 2026) adds detailed process supervision and verification during inference. RSCoVLM (Li et al., 2026) jointly trains scene understanding, grounding, and detection within a unified framework. Earth-OneVision (Cai et al., 2026) covers six sensor modalities and nine task categories through hierarchical alignment and progressive adaptation across modalities. We complement these efforts with spatially aligned detail extraction and gated reuse conditioned on current LLM visual states.

High-Resolution Visual Conditioning. Preserving visual detail requires spatial coverage and sustained visual conditioning (Ye et al., 2023, Awadalla et al., 2023, Meng et al., 2024). UReader (Ye et al., 2023) adapts cropping to image shape to preserve fine details. Standard InternVL3.5 (Wang et al., 2025b) applies dynamic tiling, compresses each tile from 1024 to 256 tokens through pixel shuffle, and projects features into the LLM with an MLP. Its Flash variant selects tile compression rates adaptively before LLM processing. Take Along Visual Conditioning (TVC) (Sun et al., 2025) reintroduces image conditions during long reasoning chains to counter visual forgetting. DeepStack (Meng et al., 2024) injects visual tokens from high-resolution views into multiple LLM layers. Qwen3-VL (Bai et al., 2025a) uses dedicated mergers to add features from three encoder depths to the first three LLM layers. We instead construct an aligned detail bank from overlapping windows across scales and encoder depths, using current visual states to guide stack fusion and gated updates at each position and channel within intermediate language-model layers.

3 METHOD

This section presents MF-RSVLM in three stages. We first introduce the architecture and preliminaries (Sections 3.1 and 3.2), then construct aligned visual detail representations (Section 3.3), and finally describe their adaptive fusion and repeated gated injection at selected LLM layers (Section 3.4).

3.1 MODEL ARCHITECTURE

Our model adopts a vision-encoder $\rightarrow$ MLP projector $\rightarrow$ LLM pipeline for remote-sensing imagery. We use pre-trained CLIP ViT-L/14@336 (Radford et al., 2021) as the vision backbone and Vicuna-v1.5 (7B) (Chiang et al., 2023) as the instruction-following LLM, with a lightweight MLP mapping CLIP features into the Vicuna embedding space. As shown in Figure 1, the shared encoder extracts a global representation and local features at complementary scales. The detail pathway organizes local features across scales and encoder layers for adaptive fusion and gated updates. Pretraining adapts the vision encoder, projector, and LLM to remote-sensing imagery. Supervised fine-tuning starts from this checkpoint and freezes the vision encoder while optimizing the projector, fusion, routing and gating modules, and LLM on the instruction corpus.

3.2 PRELIMINARIES

An input RGB image is denoted by $X \in \mathbb{R}^{C\times H\times W}$, where C=3 is the number of channels, and H, W are the original height and width. We construct a square high-resolution canvas $\mathcal{C} \in \mathbb{R}^{C\times S\times S}$ by resizing X to a fixed resolution $S \times S$ (we use S=672 in our experiments). A vision backbone F_v encodes image patches into visual tokens of dimension d_{vit}. Features extracted at encoder block ℓ are denoted using $F_v^{(\ell)}$. Unless stated otherwise, all subsequent equations use a single-image convention and omit the batch dimension, batched implementations apply the same operations independently to each image. Indices s, ℓ, and l identify spatial scales, encoder blocks, and LLM layers, respectively. We denote the aligned visual sequence length by T_v and the LLM hidden dimension by d_{llm}.

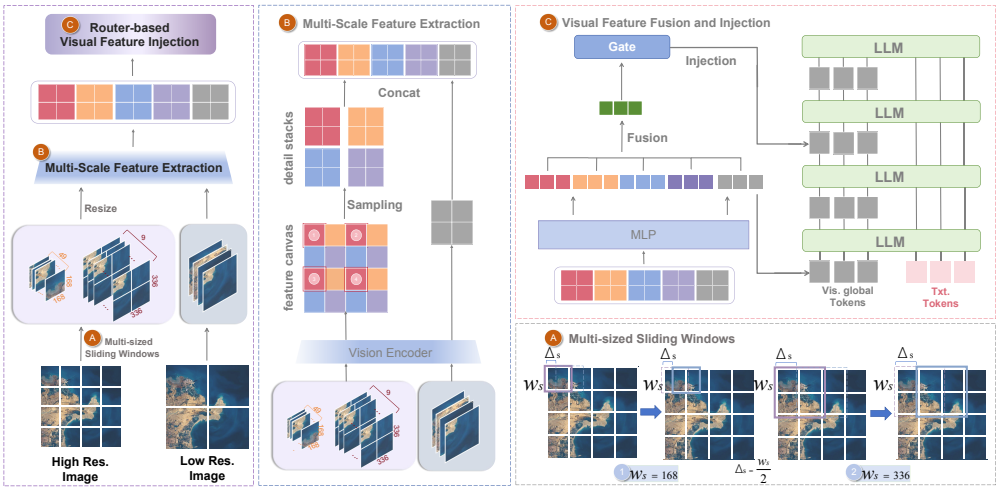

Figure 1: **Overview of our method.** The input image provides a low-resolution global view and a high-resolution canvas. Token-aligned windows extract overlapping crops from the canvas (A). A shared vision encoder processes these crops and the global view, retaining features from selected encoder layers. Local features are stitched onto feature canvases and phase-subsampled to form detail stacks (B). The global and detail sequences are concatenated and projected into the LLM embedding space through an MLP. At selected LLM layers, adaptive fusion weights the detail stacks using the current visual states, and a gate controls the residual update to visual tokens (C).

3.3 Visual Detail Representation

Low-Resolution Global Feature Extraction. Given an input image $X \in \mathbb{R}^{C\times H\times W}$, we first resize X to a low resolution (H', W') and feed it into the visual encoder F_v:

$$G = F_v\big(\mathrm{Resize}(X, H', W')\big) \in \mathbb{R}^{T_v \times d_{\mathrm{vit}}}. \tag{1}$$

Here, G denotes the global visual tokens, and T_v is the corresponding number of visual tokens.

High-Resolution Local Feature Extraction. To retain fine-grained information during local feature extraction, we use the high-resolution canvas $\mathcal{C}$ defined above, with S=672 pixels per side. Cropping at complementary scales (Figure 1-A) exposes local structures within differently sized spatial neighborhoods, while the global view retains scene-wide context. We discretize $\mathcal{C}$ into a $(S/p) \times (S/p)$ token grid using patch size p=14, providing shared coordinates for consistent window placement and feature stitching across complementary spatial scales and selected encoder layers.

For each spatial scale s, we choose an even number of tokens per side t_s and define a square window of size w_s=$t_s p$ with stride $\Delta_s = \frac{t_s}{2}\, p$. This yields overlapping windows that are aligned on the shared token grid, requiring t_s to divide S/p to ensure that windows at different scales are mutually aligned. Let (i, j) index a window at scale s by its top-left pixel coordinates $(i\,\Delta_s,\ j\,\Delta_s)$ on the canvas. The corresponding image patch is

$$P_{i,j}^{(s)} = \mathcal{C}\big[\,:,\ i\Delta_s : i\Delta_s + w_s,\ j\Delta_s : j\Delta_s + w_s\big]. \tag{2}$$

Regardless of its source-window size, each patch is resized to $w \times w$, where w=336 is the common encoder input side length, then encoded by F_v and spatially aligned as

$$H_{i,j}^{(s,\ell)} = F_v^{(\ell)}\big(\mathrm{Resize}(P_{i,j}^{(s)},\ w \times w)\big), \qquad \widetilde{H}_{i,j}^{(s,\ell)} = \mathcal{A}_{q\to t_s}\Big(H_{i,j}^{(s,\ell)}\Big). \tag{3}$$

Here, q=w/p=24; H and $\widetilde{H}$ use $q \times q$ and $t_s \times t_s$ grids with feature dimension d_{vit}. The operator $\mathcal{A}_{q\to t_s}$ is identity for 336×336 source windows and 2×2 average pooling for 168×168 source windows. Thus, the two scales differ in source field of view and magnification while sharing the same encoder resolution. Two scales and layers $\{8, 16, 24\}$ yield six feature sources.

Feature Stitching and Stack Construction. Aligned local coordinates are mapped by

$$(u, v) = (u' + i \cdot t_s/2, v' + j \cdot t_s/2). \tag{4}$$

Let $\mathcal{N}(u, v)$ denote the windows covering canvas position (u, v). We combine their features using 2D Hann weights $W_{i,j}[u, v]$ to compute $I^{(s,\ell)}[u, v]$:

$$I^{(s,\ell)}[u, v] \;=\; \frac{\sum_{(i,j)\in\mathcal{N}(u,v)} W_{i,j}[u, v]\, \widetilde{H}^{(s,\ell)}_{i,j}[u - it_s/2,\, v - jt_s/2]}{\sum_{(i,j)\in\mathcal{N}(u,v)} W_{i,j}[u, v]}\,. \tag{5}$$

For each feature map $I^{(s,\ell)}$ from scale s at layer ℓ, we convert it into a bank of ordered sequences by sub-sampling the token canvas with a fixed downsampling factor f, obtaining $\mathcal{B} = \{D_k\}_{k=1}^{N_{\text{stack}}}$. With two source-window sizes $w_s \in \{336, 168\}$ and three layers $\ell \in \{8, 16, 24\}$, the total number of sequences is $N_{\text{stack}} = 6f^2$ (e.g., N_{stack}=24 when f=2).

3.4 Adaptive Fusion and Gated Injection

Feature Fusion. We introduce an adaptive fusion module (Figure 1-C) that weights detail stacks across spatial scales and encoder layers using the current visual stream. Given the global vision tokens G and the detail bank $\{D_i\}_{i=1}^{N_{\text{stack}}}$, we concatenate the sequences along the token axis and map their features into the LLM embedding space using a shared learned projector:

$$V = \text{concat}(G, D_1, \ldots, D_{N_{\text{stack}}}), \qquad U = \text{projector}(V). \tag{6}$$

After the shared projection, U is partitioned back into the projected global sequence $U[\text{vis}]$ and projected detail sequences $U_i \in \mathbb{R}^{T_v \times d_{\text{llm}}}$. Only $U[\text{vis}]$ initializes the LLM visual prefix, the U_i sequences remain in the detail bank for intermediate-layer fusion. Each detail stack has the same sequence length T_v as the global stream, and the initial visual prefix therefore remains T_v=576 tokens. At the selected layers $l \in \{2, 4, 6, 8\}$, let $H^{(l)} \in \mathbb{R}^{T \times d_{\text{llm}}}$ be the hidden states, where T is the total sequence length (text tokens and visual tokens) and d_{llm} is the hidden dimension of the LLM, and let $H^{(l)}[\text{vis}] \in \mathbb{R}^{T_v \times d_{\text{llm}}}$ denote their visual subsequence. These states provide the context for detail routing at each layer.

To obtain one routing score per stack, we summarize the current visual stream and each detail stack by mean pooling over their T_v positions, yielding compact summaries for comparing detail sources:

$$\bar{h}^{(l)} = \tfrac{1}{T_v} \sum_{t=1}^{T_v} H^{(l)}[\text{vis}]_t, \quad \bar{u}_i = \tfrac{1}{T_v} \sum_{t=1}^{T_v} U_i[t]. \tag{7}$$

Here, $\bar{h}^{(l)}$ summarizes the evolving visual states, while $\bar{u}_i$ summarizes detail source i. Pooling is used only to compute routing weights, leaving the full sequences U_i available for fusion. Learned mappings at layer l transform these summaries into a query and stack keys:

$$Q^{(l)} = \text{RouterQ}^{(l)}(\bar{h}^{(l)}), \quad K_i^{(l)} = \text{RouterK}^{(l)}(\bar{u}_i), \tag{8}$$

where $Q^{(l)}, K_i^{(l)} \in \mathbb{R}^{d_{\text{llm}}}$. Their scaled dot products measure the compatibility of each detail source with the current visual stream. Softmax normalization across stacks produces weights that sum to one and are shared across token positions. We apply these weights to the full detail sequences:

$$M_t^{(l)} \;=\; \sum_{i=1}^{N_{\text{stack}}} \frac{\exp\big((K_i^{(l)})^\top Q^{(l)}/\sqrt{d_{\text{llm}}}\big)}{\sum_{r=1}^{N_{\text{stack}}} \exp\big((K_r^{(l)})^\top Q^{(l)}/\sqrt{d_{\text{llm}}}\big)}\, U_i[t], \tag{9}$$

where $t = 1, \ldots, T_v$. Each output $M_t^{(l)}$ combines features at the corresponding sequence index t. Routing controls the contribution of each stack without attention across token positions. The following gate separately controls the update strength at each position and channel.

Gated injection. We inject $M^{(l)}$ into the visual stream through a lightweight gate that controls update strength at each position and channel. For each visual-token position t, layer normalization (LN) is first applied independently to the current visual state and the fused detail:

$$\tilde{H}^{(l)}[\text{vis}]_t = \text{LN}(H^{(l)}[\text{vis}]_t), \qquad \tilde{M}_t^{(l)} = \text{LN}(M_t^{(l)}). \tag{10}$$

The normalized features are concatenated, mapped by a position-wise linear layer, and passed through a sigmoid. The resulting per-channel gate then modulates the separately projected detail feature:

$$g_t^{(l)} = \sigma\Big(W_g^{(l)}[\tilde{H}^{(l)}[\text{vis}]_t; \tilde{M}_t^{(l)}] + b_g^{(l)}\Big), \delta_t^{(l)} \quad = g_t^{(l)} \odot \text{Proj}^{(l)}(\tilde{M}_t^{(l)}). \tag{11}$$

Here, $W_g^{(l)}$ and $b_g^{(l)}$ parameterize the gate's linear map, σ is the sigmoid function, $[\,;\,]$ denotes concatenation, $\mathrm{Proj}^{(l)} : \mathbb{R}^{d_{\mathrm{llm}}} \to \mathbb{R}^{d_{\mathrm{llm}}}$ is position-wise and shared across positions, and $\odot$ denotes elementwise multiplication. We scale the increment by a learned scalar s_l and add it to the current visual state:

$$\widehat{H}^{(l)}[\mathrm{vis}]_t \;=\; H^{(l)}[\mathrm{vis}]_t + s_l\,\delta_t^{(l)}, \tag{12}$$

where $t = 1, \ldots, T_v$ and $\widehat{H}^{(l)}$ denotes the hidden states after injection at LLM layer l. Only visual-token positions receive this direct residual update. Recurrent injection applies the operation at selected LLM layers, with fusion weights and gates computed from the states at each layer. This mechanism maintains access to the detail bank during intermediate processing to address visual forgetting.

4 Experiment

We present the datasets, evaluation metrics, and training settings in Sec. 4.1. We then compare captioning, scene classification, and VQA performance in Sec. 4.2, followed by analyses of feature configurations, gating, routing, and computational costs in Sec. 4.3.

4.1 Implementation Details

Fine-tuning Instruction Set Construction. We construct a remote-sensing instruction-tuning corpus following the design principles of VHM (Pang et al., 2024) to support multi-task fine-tuning. We select instruction data from VersaD-Instruct (Pang et al., 2024) and VRSBench (Li et al., 2024). We then aggregate predominantly optical RGB imagery from widely used remote-sensing benchmarks (e.g., VRSBench (Li et al., 2024), RSVQA (Lobry et al., 2020), DOTA (Xia et al., 2018), FAIR1M (Sun et al., 2022), DIOR (Li et al., 2020), NWPU-RESISC45 (Cheng et al., 2017), fMoW (Christie et al., 2018) and RSITMD (Yuan et al., 2022)). To reduce distributional bias, we adopt geography- and resolution-aware sampling. For each image, GPT-5 (Singh et al., 2025) synthesizes single- and multi-turn, verifiable instructions conditioned on scene descriptions and annotations. All examples are mapped to a unified schema consisting of an image, an instruction, and an output, spanning across six downstream tasks: captioning, VQA, visual grounding, scene classification, instruction-style QA, and detection. Specifically, we design the prompts to ensure that instructions explicitly encode key visual attributes (e.g., color, shape, absolute and relative position), keeping answers strictly grounded in the image. Finally, we unify taxonomies, filter near-duplicates within the assembled training corpus, and perform automated answer–annotation consistency checks. Table 6 summarizes the composition of the resulting 293K-sample instruction corpus.

Datasets and Metrics. Following recent work (Pang et al., 2024, Soni et al., 2025), we evaluate our model on standard remote-sensing benchmarks across three task families—visual question answering (VQA), scene classification, and image captioning—and report task-appropriate metrics. For VQA, we use RSVQA-LR (Lobry et al., 2020) and VRSBench VQA (Li et al., 2024), which contains nine perception-oriented categories. For scene classification, the primary comparison uses six datasets—AID (Xia et al., 2017), WHU-RS19 (Dai & Yang, 2010), NWPU-RESISC45 (Cheng et al., 2017), SIRI-WHU (Zhu et al., 2016, Zhao et al., 2016; 2015), EuroSAT (Helber et al., 2019), and METER-ML (Zhu et al., 2022)—and reports per-dataset Top-1 accuracy and their unweighted macro-average. The local fMoW split is reserved for diagnostic analyses. For the image captioning task, we evaluate on UCM-Captions (Qu et al., 2016), RSICD (Lu et al., 2017), RSITMD (Yuan et al., 2022), NWPU-Captions (Cheng et al., 2017) and Sydney-Captions (Qu et al., 2016) datasets, and report BLEU-4 (B-4), METEOR (MT), CIDEr (Cr), and ROUGE-L (R-L) metrics.

Experimental Details. Both training stages use $8\times$A6000 GPUs and a learning rate of 2×10^{-5}. Pretraining on VersaD (Pang et al., 2024) uses 12 examples per GPU and takes approximately 42 hours. Supervised fine-tuning (SFT) enables token-aligned windows and the feature enhancement pathway, freezes CLIP ViT-L/14@336 at 336×336 encoder inputs, and updates the MLP projector, fusion, routing and gating components, and Vicuna-v1.5 (7B). SFT uses our 293K instruction corpus, whose images are predominantly no larger than 800×800, with six examples per GPU, taking approximately 38 hours. Table 8 summarizes the main-model settings for reproducible comparison. Details of the baseline models and their categorization are provided in Appendix C.

4.2 MAIN RESULTS

Comparison on Image Captioning. To assess caption generation, we compare our method with general VLMs and remote sensing models, including GPT-5, Qwen2.5-VL, VHM, EarthDial, and RSThinker, using four metrics on five benchmarks in Table 1. For the headline comparison, we use RSICD, NWPU-Captions, and Sydney-Captions. We achieve the highest ROUGE-L scores in this subset on all three datasets, reaching 47.80, 68.32, and 72.81, respectively. On NWPU-Captions and Sydney-Captions, our CIDEr scores reach 138.18 and 254.04, compared with EarthDial's 123.60 and 120.90. On RSICD, our METEOR score is 27.46 versus RSThinker's 27.17, although CIDEr remains lower at 78.54 versus 99.83 and BLEU-4 does not lead on these three datasets. Our main strength in this comparison is higher ROUGE-L across the three benchmarks, supported by stronger CIDEr on NWPU-Captions and Sydney-Captions.

Table 1: Image captioning on five benchmarks using BLEU-4, METEOR, CIDEr, and ROUGE-L (scores $\times 100$). Ranks: 1st, 2nd, 3rd, 4th. Ours marks our model.

Model	UCM-Captions				RSICD				RSITMD				NWPU-Captions				Sydney-Captions			
	B-4	MT	Cr	R-L	B-4	MT	Cr	R-L	B-4	MT	Cr	R-L	B-4	MT	Cr	R-L	B-4	MT	Cr	R-L
Closed-source Commercial Vision-Language Models																				
Claude-Sonnet-4 (Anthropic, 2025)	20.12	20.99	30.04	13.35	11.58	13.90	24.57	10.63	20.14	17.15	19.31	9.13	28.32	21.98	32.46	13.38	19.85	20.14	27.55	12.52
Gemini-2.0 (DeepMind, 2024)	9.31	6.72	13.23	5.48	10.85	8.71	21.53	9.41	15.73	9.27	17.11	7.92	20.55	11.42	22.58	9.45	31.41	24.17	38.76	16.99
GPT-5 (Singh et al., 2025)	28.49	25.56	40.95	17.82	16.83	16.73	34.39	15.86	27.27	21.10	29.48	14.02	39.62	25.69	48.52	20.91	28.50	24.48	39.09	17.47
Open-source Vision-Language Models																				
MiniGPT-v2 (Chen et al., 2023a)	25.46	19.62	30.94	13.82	15.40	12.36	26.63	12.21	25.45	16.83	25.89	11.55	37.75	19.70	35.73	15.18	26.17	17.03	23.55	12.30
Qwen2.5-VL (Bai et al., 2025b)	27.87	21.48	35.36	17.23	17.80	13.72	32.19	14.62	27.92	17.24	24.90	12.20	38.89	21.40	42.11	17.75	28.60	18.77	31.81	16.87
Qwen3-VL-8B (Bai et al., 2025a)	3.44	14.92	26.20	28.47	2.93	14.29	14.18	22.95	1.23	12.21	11.56	20.13	3.01	14.37	16.26	24.64	3.20	15.13	13.46	30.12
Open-source Remote Sensing Vision-Language Models																				
VHM (Pang et al., 2024)	42.08	27.86	66.12	25.17	25.66	17.63	49.80	20.50	38.93	21.99	40.29	18.43	50.69	25.31	54.92	22.91	44.67	35.11	67.50	23.76
SkySenseGPT (Luo et al., 2024)	39.04	23.52	49.80	22.63	23.33	14.02	40.48	18.01	37.76	19.06	34.98	15.00	48.03	22.41	49.67	18.68	42.47	24.95	52.58	21.51
EarthDial (Soni et al., 2025)	59.77	44.08	127.70	32.43	29.09	25.20	85.82	24.19	42.09	23.92	42.56	18.35	67.14	46.17	123.60	28.96	64.04	54.91	120.90	43.75
RSThinker (Liu et al., 2025)	61.03	41.54	123.40	34.80	39.82	27.17	99.83	29.38	55.69	32.29	73.55	25.66	85.12	58.88	94.81	28.97	60.47	35.28	73.50	25.96
GeoSolver (Sun et al., 2026)	2.91	10.47	26.81	18.51	2.09	7.90	15.14	13.76	1.42	7.24	15.03	13.31	3.50	9.79	29.09	18.25	3.37	11.61	1.25	16.16
RSCoVLM (Li et al., 2026)	9.63	17.55	62.64	34.32	8.05	15.89	36.02	29.89	4.41	13.72	29.81	25.70	9.24	16.97	43.88	31.78	15.28	15.29	22.61	36.07
MF-RSVLM (Ours)	79.13	53.41	405.75	89.12	25.08	27.46	78.54	47.80	13.25	17.73	61.16	37.11	56.76	39.32	138.18	68.32	56.02	40.19	254.04	72.81

Comparison on Scene Classification. To assess scene recognition, we compare our method with general and remote sensing VLMs, including Qwen2.5-VL, GeoChat, EarthDial, and LHRS-Bot, across six datasets in Table 2. Across all displayed models, we reach 78.20% macro-average accuracy, exceeding LHRS-Bot by 2.83 percentage points. Our NWPU-RESISC45 and METER-ML accuracies are 94.29% and 74.87%, compared with LHRS-Bot's 83.94% and 69.81%, giving gains of 10.35 and 5.06 points. We also improve AID and SIRI-WHU, indicating that the average advantage is supported by several datasets. WHU-RS19 remains slightly lower at 93.10% versus 93.17%, while EuroSAT shows a larger gap at 45.30% versus 51.40%. Our main strength is improved average scene recognition, with gains over LHRS-Bot on four of the six datasets.

Table 2: Scene-classification results on six benchmarks (accuracy, %). Avg denotes the average score across the six datasets. Ranks: 1st, 2nd, 3rd, 4th. Ours marks our model.

Model	LLM	AID	WHU-RS19	NWPU	SIRI-WHU	EuroSAT	METER-ML	Avg
Open-source Vision-Language Models								
LLaVA-1.5 (Liu et al., 2023a)	Vicuna-1.5-7B	31.10	54.55	34.96	17.71	26.12	21.73	31.03
MiniGPT-v2 (Chen et al., 2023a)	LLaMA-2-7B	32.96	64.80	28.15	35.46	38.56	14.29	35.70
InstructBLIP (Dai et al., 2023)	Vicuna-7B	29.50	36.76	34.01	18.20	20.25	14.42	25.52
mPLUG-OWL2 (Ye et al., 2024)	LLaMA-2-7B	48.79	72.66	46.58	54.83	33.29	36.27	48.74
Qwen-VL (Bai et al., 2023)	Qwen-7B	55.30	72.25	42.73	54.58	26.42	38.77	48.34
Qwen2.5-VL (Bai et al., 2025b)	Qwen2.5-7B	63.07	76.60	64.35	61.54	34.25	59.82	59.94
Qwen3-VL (Bai et al., 2025a)	Qwen3-8B	66.67	88.60	68.92	67.63	41.55	60.71	65.68
InternLM-XCompose (Zhang et al., 2023a)	InternLM-7B	51.61	72.89	47.51	46.83	39.70	40.21	49.79
InternVL3.5 (Wang et al., 2025b)	Qwen3-8B	73.80	91.50	92.35	66.67	45.20	50.80	70.05
InternVL2.5 (Chen et al., 2024c)	InternLM-2.5	64.00	80.30	57.94	57.67	49.80	57.35	61.18
Phi-3.5-Vision (Research, 2024)	Phi-3.5	56.57	68.90	48.03	40.42	25.90	40.35	46.70
MiniCPM-V-2.6 (Yao et al., 2024)	Qwen2-7B	65.07	79.50	62.13	61.50	32.25	51.50	58.66
Remote-sensing Vision-Language Models								
VHM-RL (Köksal & Alatan, 2025)	Qwen2-VL-2B	70.87	88.70	73.59	62.42	31.40	61.06	64.67
GeoChat (Kuckreja et al., 2024)	Vicuna-1.5-7B	73.17	84.80	89.24	53.17	39.25	33.98	62.27
EarthDial (Soni et al., 2025)	Phi-3-mini	87.57	95.80	96.92	60.42	45.85	27.26	68.97
LHRS-Bot (Muhtar et al., 2024)	LLaMA-2-7B	91.26	93.17	83.94	62.66	51.40	69.81	75.37
RSThinker (Liu et al., 2025)	GLM-4.1V-9B	97.57	99.40	97.84	74.17	24.45	45.49	73.15
GeoSolver (Sun et al., 2026)	GeoSolver-9B	96.87	99.30	97.40	75.08	22.95	42.12	72.29
MF-RSVLM (Ours)	Vicuna-v1.5-7B	94.37	93.10	94.29	67.29	45.30	74.87	78.20

Comparison on Visual Question Answering. To assess visual understanding across question types, we compare our method with general and remote sensing VLMs, including GPT-5, Qwen2.5-VL, GLM-4.1V, VHM, and GeoChat, on VRSBench in Table 3. Across all models, we achieve the highest nine-category Avg of 65.76%, exceeding the strongest baseline, Claude-Sonnet-4 at 62.07%, by 3.69 percentage points. The corresponding margins over GLM-4.1V and VHM are 4.93 and 14.25 points. Existence accuracy reaches 90.23%, compared with GPT-5's 88.89% and VHM's 86.81%. Category

Table 3: VRSBench VQA results (accuracy, %). Avg is the unweighted mean of the nine displayed category accuracies, rounded to two decimal places. Ranks: 1st, 2nd, 3rd, 4th. Ours marks our model.Rankings cover all displayed models and are based on distinct scores within each column, in descending order; ties share the same rank. "–" denotes unavailable information.

Model	LLM	Category	Existence	Position	Quantity	Scene	Color	Image	Shape	Direction	Avg
Closed-source Vision-Language Models											
Claude-Sonnet-4 (Anthropic, 2025)	–	43.28	52.78	30.17	66.67	64.79	63.29	91.67	78.92	67.09	62.07
Gemini-2.0 (DeepMind, 2024)	–	44.03	86.11	43.97	46.00	60.56	56.96	95.83	55.94	54.09	60.39
GPT-5 (Singh et al., 2025)	–	39.55	88.89	42.24	47.33	70.42	59.49	87.50	62.33	42.98	60.08
Open-source Vision-Language Models											
MiniGPT-v2 (Chen et al., 2023a)	LLaMA-2-7B	25.37	56.25	20.69	44.00	45.07	36.71	33.33	37.88	51.15	38.94
Qwen2.5-VL (Bai et al., 2025b)	Qwen2.5-7B	37.31	75.69	37.93	44.00	67.61	63.29	91.67	51.09	33.75	55.82
InternVL3.5 (Wang et al., 2025b)	Qwen3-8B	53.68	83.14	72.65	51.62	44.51	19.01	55.27	22.35	38.57	48.98
InternVL2.5 (Chen et al., 2024c)	InternLM2.5-7B-Chat	48.10	87.03	65.44	38.63	63.47	70.45	63.06	57.91	41.09	59.46
Phi-3.5-Vision (Research, 2024)	Phi-3-mini-3.8B	40.85	79.59	62.42	37.25	54.68	61.97	70.68	55.87	35.43	55.42
MiniCPM-V-2.6 (Yao et al., 2024)	Qwen2-7B	50.06	78.87	60.50	33.89	63.43	71.72	67.32	62.19	50.52	59.83
GLM-4.1V (Hong et al., 2025)	GLM-4.1-9B	42.54	86.11	43.10	54.67	69.01	62.03	87.50	45.68	56.81	60.83
Remote-sensing Vision-Language Models											
VHM (Pang et al., 2024)	Vicuna-7B	50.75	86.81	36.21	42.67	53.52	55.70	54.17	51.51	32.29	51.51
SkySenseGPT (Luo et al., 2024)	Vicuna-7B	57.46	84.03	44.83	38.00	53.52	16.46	45.83	19.18	42.98	44.70
GeoChat (Kuckreja et al., 2024)	Vicuna-7B	62.27	86.67	73.58	32.65	49.05	26.00	42.16	22.07	44.03	48.72
EarthDial (Soni et al., 2025)	Phi-3-mini-3.8B	51.49	47.22	36.21	41.33	36.62	11.39	50.00	22.07	35.43	36.86
MF-RSVLM (Ours)	Vicuna-v1.5-7B	65.84	90.23	64.26	51.82	59.18	64.48	87.42	62.68	45.91	65.76

and Shape accuracies are 65.84% and 62.68%, also exceeding VHM's 50.75% and 51.51%. Spatial questions remain challenging, with Position at 64.26% versus GeoChat's 73.58% and Direction at 45.91%, our lowest category score. Our main strength is the highest average accuracy across all displayed models, especially on existence questions requiring reliable visual grounding.

4.3 Analysis

Comparison of Spatial Coverage and Gating. Figure 2 presents four analyses of visual coverage and update strength from left to right. The leftmost panel visualizes the coverage produced by nine coarse and 49 fine windows on a 672×672 canvas, showing how overlapping observations combine complementary spatial neighborhoods. The second panel compares window configurations, with the dual-scale setting increasing mean accuracy from 80.96% to 88.25% in the tested configuration. The third panel compares learned gates with the fixed settings $g{=}0$ and $g{=}1$, which achieve mean accuracies of 74.5%, 25.3%, and 30.9%, respectively; $g{=}0$ suppresses the detail update, whereas $g{=}1$ removes adaptive modulation while retaining the update pathway. This intervention includes the local fMoW split as a diagnostic condition. The rightmost panel reports learned-gate statistics: median values near 0.06 and 90th percentiles near 0.26 indicate that most updates are modest, with a smaller subset receiving stronger modulation. These activation summaries describe model behavior but do not by themselves establish causal importance. Together, the four panels support complementary spatial coverage and adaptive updates within the tested settings, while Appendix D.6 reports paired interventions on the selected six-dataset subset used for component diagnostics.

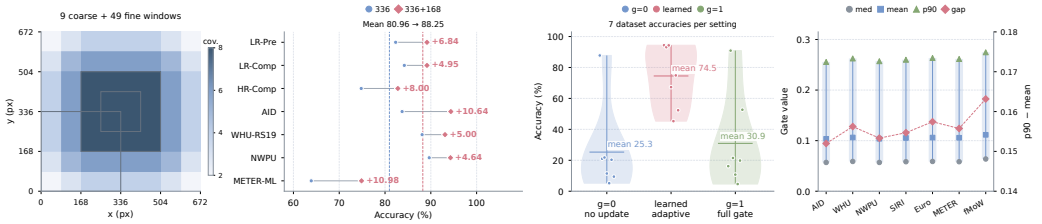

Figure 2: Spatial coverage and gating comparisons, from left to right: coverage, window settings, interventions with fixed gates, and learned gate statistics.

Feature Configurations and Efficiency. Figure 3 presents analyses of four scene-classification datasets and three VQA subtasks. In the first panel, gray, blue, and red denote ViT-24 with both scales, ViT-8/16/24 with only the 336×336 window, and ViT-8/16/24 with both scales, respectively. This panel compares joint feature-source configurations rather than a progressive depth ablation: the full configuration reaches 88.25% mean accuracy across seven evaluations, exceeding the ViT-layer-24-only and single-window-scale configurations by 5.89 and 7.29 percentage points. The second panel estimates window-sampling costs; reducing source windows from 168×168 to 84×84 increases crops from 49 to 225 and detail-branch cost by $4.59\times$. Because every crop retains a 336×336

encoder input and 576 output tokens, finer windows increase the number of encoder evaluations rather than reduce the cost of each crop. The third panel compares injection schedules with feature sources fixed: injection at layers 2/4/6/8 exceeds layer 8 alone by 4.51 points, although partial schedules do not consistently outperform the strongest single injection. The rightmost panel reports system costs, with 8.6% more peak memory and 44.4% higher latency, while the visual prefix remains at 576 tokens. The unchanged prefix length indicates that the added cost is not caused by a longer initial visual sequence, but by the additional visual processing and intermediate updates. Overall, these comparisons support complementary feature sources and the full tested injection schedule while highlighting the cost of denser spatial sampling.

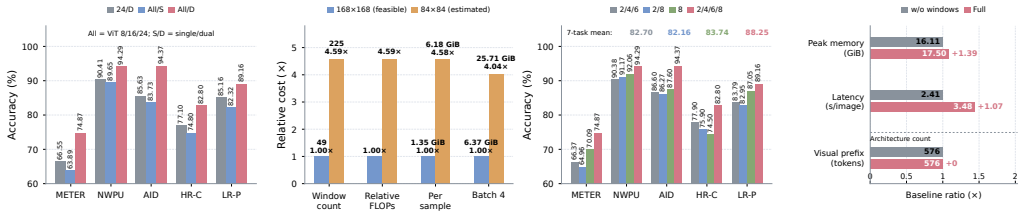

Figure 3: Feature configurations, window costs, injection schedules, and system costs, from left to right. Finer-window costs are estimated from encoder evaluations. The system panel reports peak memory, per-image inference latency, and visual-prefix length relative to the baseline configuration.

Gate Activation and Injection Effects. To assess the behavior of gated visual updates, we evaluate all 12,115 examples from six classification evaluation sets, weighting results by sample count. Table 4 reports gate means, standard deviations (SD), and layer-removal effects. Here, $\Delta \log p$ is the full-model minus ablated-model mean token log probability of the correct answer, and removal effects are displayed as $10^3 \Delta \log p$ in millinats/token. Removing the update at any tested layer reduces correct-answer support on average. Layer 4 has the largest mean gate activation (0.1608) and removal effect ($+24.922$), while layer 2 has the smallest mean activation but the second-largest effect ($+15.261$). Overall, the results show that gate magnitude alone does not determine contribution, with early-layer updates providing the strongest support for correct answers.

Table 4: Gate statistics and effects.

LLM layer	Mean	SD	$10^3 \Delta \log p$
2	0.0554	0.0028	$+15.261$
4	0.1608	0.0080	$+24.922$
6	0.1040	0.0064	$+2.643$
8	0.1001	0.0072	$+1.436$

Routing Allocation and Removal Effects. To examine the utility of routed details, we aggregate weights over the same evaluation sets and four injection layers. Each group combines four phase stacks from one window size and ViT layer. Table 5 reports mean weights and scaled removal effects, with remaining weights renormalized after removal. The ViT-layer-8 group with 336-pixel windows and the ViT-layer-24 groups with 168- and 336-pixel windows receive 97.32% of the total weight. Removing the first group reduces correct-answer support by 3.824 millinats/token. Removing the group with 168-pixel windows at ViT layer 24 reduces it by 1.741. The positive effects occur for selected groups, while the negative effect for the 336-pixel, ViT-layer-24 group shows that routing weight and marginal removal effect need not align. This concentration of routing weights, together with the removal effects, indicates that allocation and marginal utility capture distinct aspects of feature use during multimodal processing.

Table 5: Routing weights and effects.

Window	ViT layer	Weight (%)	$10^3 \Delta \log p$
336	8	47.50	$+3.824$
336	16	0.18	-0.191
336	24	23.77	-1.694
168	8	2.31	-0.547
168	16	0.19	$+0.006$
168	24	26.05	$+1.741$

5 Conclusion

We introduced MF-RSVLM, which combines multi-scale visual extraction with recurrent feature injection to preserve fine-grained structures and mitigate visual forgetting. Benchmark comparisons and component studies show task-dependent benefits and additional computational costs, characterizing the practical trade-offs of multi-feature fusion for remote-sensing vision–language understanding. Future work will focus on reducing multi-scale processing costs and improving robustness across remote-sensing datasets with different spatial resolutions and scene distributions.

AI USE STATEMENT

AI tools, including GPT-5, were used to synthesize single- and multi-turn instructions for the instruction-tuning corpus based on existing scene descriptions and annotations. AI tools also assisted with manuscript editing and presentation. They were not used to generate or modify reported experimental measurements or replace the authors' scientific judgment. The authors reviewed the AI-assisted content and take full responsibility for the manuscript.

REPRODUCIBILITY STATEMENT

We document the model architecture, training configuration, datasets, and evaluation procedures to support reproducibility. Supplementary material provides implementation details and additional experimental settings. Results obtained under different evaluation protocols or reference configurations are reported separately, with limitations in the available records explicitly noted.

REFERENCES

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7822–7831, 2021.

Anthropic. Introducing claude 4. *Anthropic blog*, 2025. URL `https://www.anthropic.com/news/claude-4`.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Miaoxin Cai, Guanqun Wang, Wei Zhang, Guangyao Zhou, Yin Zhuang, Tong Zhang, Hao Wang, He Chen, and Jun Li. Earth-onevision: Extending remote sensing multimodal large language models to more sensor modalities and tasks. *arXiv preprint arXiv:2606.10819*, 2026.

Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021.

Hongruixuan Chen, Jian Song, Chengxi Han, Junshi Xia, and Naoto Yokoya. Changemamba: Remote sensing change detection with spatiotemporal state space model. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–20, 2024a.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023a.

Keyan Chen, Bowen Chen, Chenyang Liu, Wenyuan Li, Zhengxia Zou, and Zhenwei Shi. Rsmamba: Remote sensing image classification with state space model. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024b.

Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3367–3375, 2023b.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024c.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6172–6180, 2018.

Dengxin Dai and Wen Yang. Satellite image classification via two-layer sparse coding with biased image representation. *IEEE Geoscience and remote sensing letters*, 8(1):173–176, 2010.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Google DeepMind. Gemini 2.0 flash. `https://deepmind.google/technologies/gemini/`, 2024. Large language model.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Junyi Hu, Tian Bai, Fengyi Wu, Zhenming Peng, and Yi Zhang. Pyramid sparse transformer: Enhancing multi-scale feature fusion with dynamic token selection. *arXiv preprint arXiv:2505.12772*, 2025.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021.

Aybora Köksal and A Aydın Alatan. Few-shot vision-language reasoning for satellite imagery via verifiable rewards. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6960–6969, 2025.

Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27831–27840. IEEE, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023a.

Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing*, 159:296–307, 2020.

Qingyun Li, Shuran Ma, Junwei Luo, Yi Yu, Yue Zhou, Fengxiang Wang, Xudong Lu, Xiaoxing Wang, Xin He, Yushi Chen, et al. Co-training vision-language models for remote sensing multi-task learning. *Remote Sensing*, 18(2):222, 2026.

Xiang Li, Jian Ding, and Mohamed Elhoseiny. Vrsbench: A versatile vision-language benchmark dataset for remote sensing image understanding. *Advances in Neural Information Processing Systems*, 37:3229–3242, 2024.

Yuxuan Li, Qibin Hou, Zhaohui Zheng, Ming-Ming Cheng, Jian Yang, and Xiang Li. Large selective kernel network for remote sensing object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16794–16805, 2023b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, et al. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023b. LLaVA-1.5 technical report.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024a.

Jiaqi Liu, Lang Sun, Ronghao Fu, and Bo Yang. Towards faithful reasoning in remote sensing: A perceptually-grounded geospatial chain-of-thought for vision-language models. *arXiv preprint arXiv:2509.22221*, 2025.

Xu Liu and Zhouhui Lian. Rsunivlm: A unified vision language model for remote sensing via granularity-oriented mixture of experts. *arXiv preprint arXiv:2412.05679*, 2024.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

Sylvain Lobry, Diego Marcos, Jesse Murray, and Devis Tuia. Rsvqa: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12):8555–8566, 2020.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Siqi Lu, Junlin Guo, James R Zimmer-Dauphinee, Jordan M Nieusma, Xiao Wang, Steven A Wernke, Yuankai Huo, et al. Vision foundation models in remote sensing: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 2025.

Xiaoqiang Lu, Binqiang Wang, Xiangtao Zheng, and Xuelong Li. Exploring models and data for remote sensing image caption generation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4):2183–2195, 2017.

Junwei Luo, Zhen Pang, Yongjun Zhang, Tingzhu Wang, Linlin Wang, Bo Dang, Jiangwei Lao, Jian Wang, Jingdong Chen, Yihua Tan, and Yansheng Li. Skysensegpt: A fine-grained instruction tuning dataset and model for remote sensing vision-language understanding. *arXiv preprint arXiv:2406.10100*, 2024.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. *Advances in Neural Information Processing Systems*, 37:23464–23487, 2024.

Dilxat Muhtar, Zhenshi Li, Feng Gu, Xueliang Zhang, and Pengfeng Xiao. LHRS-Bot: Empowering remote sensing with VGI-enhanced large multimodal language model. In *European Conference on Computer Vision (ECCV)*, 2024.

Mubashir Noman, Noor Ahsan, Muzammal Naseer, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, and Fahad Shahbaz Khan. Cdchat: A large multimodal model for remote sensing change description. In *IGARSS 2025-2025 IEEE International Geoscience and Remote Sensing Symposium*, pp. 7033–7037. IEEE, 2025.

Chao Pang, Xingxing Weng, Jiang Wu, Jiayu Li, Yi Liu, Jiaxing Sun, Weijia Li, Shuai Wang, Litong Feng, Gui-Song Xia, and Conghui He. VHM: Versatile and honest vision language model for remote sensing image analysis. *arXiv preprint arXiv:2403.20213*, 2024.

Bo Qu, Xuelong Li, Dacheng Tao, and Xiaoqiang Lu. Deep semantic understanding of high resolution remote sensing image. In *2016 International conference on computer, information and telecommunication systems (Cits)*, pp. 1–5. IEEE, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Microsoft Research. Phi-3.5-vision technical overview. Technical report, Microsoft, 2024. URL `https://www.microsoft.com/en-us/research/project/phi-3/`. Model card and documentation available online.

Akashah Shabbir, Mohammed Zumri, Mohammed Bennamoun, Fahad S Khan, and Salman Khan. Geopixel: Pixel grounding large multimodal model in remote sensing. *arXiv preprint arXiv:2501.13925*, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Sagar Soni, Akshay Dudhane, Hiyam Debary, Mustansar Fiaz, Muhammad Akhtar Munir, Muhammad Sohail Danish, Paolo Fraccaro, Campbell D Watson, Levente J Klein, Fahad Shahbaz Khan, and Salman Khan. EarthDial: Turning multi-sensory earth observations to interactive dialogues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Hai-Long Sun, Zhun Sun, Houwen Peng, and Han-Jia Ye. Mitigating visual forgetting via take-along visual conditioning for multi-modal long cot reasoning. *arXiv preprint arXiv:2503.13360*, 2025.

Lang Sun, Ronghao Fu, Zhuoran Duan, Haoran Liu, Xueyan Liu, and Bo Yang. Geosolver: Scaling test-time reasoning in remote sensing with fine-grained process supervision. *arXiv preprint arXiv:2603.09551*, 2026.

Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:116–130, 2022.

Gabriel Tseng, Anthony Fuller, Marlena Reil, Henry Herzog, Patrick Beukema, Favyen Bastani, James R Green, Evan Shelhamer, Hannah Kerner, and David Rolnick. Galileo: Learning global & local features of many remote sensing modalities. *arXiv preprint arXiv:2502.09356*, 2025.

Fengxiang Wang, Mingshuo Chen, Yueying Li, Di Wang, Haotian Wang, Zonghao Guo, Zefan Wang, Boqi Shan, Long Lan, Yulin Wang, Hongzhen Wang, Wenjing Yang, Bo Du, and Jing Zhang. GeoLLaVA-8k: Scaling remote-sensing multimodal large language models to 8k resolution. *arXiv preprint arXiv:2505.21375*, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

Xingxing Weng, Chao Pang, and Gui-Song Xia. Vision-language modeling meets remote sensing: Models, datasets, and perspectives. *IEEE Geoscience and Remote Sensing Magazine*, 13(3): 276–323, September 2025. ISSN 2473-2397. doi: 10.1109/mgrs.2025.3572702.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 2247–2256. IEEE, 2023.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3974–3983, 2018.

Aoran Xiao, Weihao Xuan, Junjue Wang, Jiaxing Huang, Dacheng Tao, Shijian Lu, and Naoto Yokoya. Foundation models for remote sensing and earth observation: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 2025.

Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pp. 270–279, 2010.

Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Shuai Bai, et al. Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21744–21754, 2025.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2841–2858, 2023.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 13040–13051, 2024.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11704–11715, 2026.

Zhiqiang Yuan, Wenkai Zhang, Kun Fu, Xuan Li, Chubo Deng, Hongqi Wang, and Xian Sun. Exploring a fine-grained multiscale method for cross-modal remote sensing image retrieval. *arXiv preprint arXiv:2204.09868*, 2022.

Yang Zhan, Zhitong Xiong, and Yuan Yuan. Skyeyegpt: Unifying remote sensing vision-language tasks via instruction tuning with large language model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 221:64–77, 2025.

Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Haodong Duan, Songyang Zhang, Shuangrui Ding, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023a.

Wei Zhang, Miaoxin Cai, Tong Zhang, Yin Zhuang, and Xuerui Mao. EarthGPT: A universal multi-modal large language model for multi-sensor image comprehension in remote sensing domain. *arXiv preprint arXiv:2401.16822*, 2024.

Xin Zhang, Chen Liu, Degang Yang, Tingting Song, Yichen Ye, Ke Li, and Yingze Song. Rfaconv: Innovating spatial attention and standard convolutional operation. *arXiv preprint arXiv:2304.03198*, 2023b.

Bei Zhao, Yanfei Zhong, Gui-Song Xia, and Liangpei Zhang. Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 54(4):2108–2123, 2015.

Bei Zhao, Yanfei Zhong, Liangpei Zhang, and Bo Huang. The fisher kernel coding framework for high spatial resolution scene classification. *Remote Sensing*, 8(2):157, 2016.

Bryan Zhu, Nicholas Lui, Jeremy Irvin, Jimmy Le, Sahil Tadwalkar, Chenghao Wang, Zutao Ouyang, Frankie Y Liu, Andrew Y Ng, and Robert B Jackson. Meter-ml: a multi-sensor earth observation benchmark for automated methane source mapping. *arXiv preprint arXiv:2207.11166*, 2022.

Qiqi Zhu, Yanfei Zhong, Bei Zhao, Gui-Song Xia, and Liangpei Zhang. Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 13(6):747–751, 2016.

APPENDIX CONTENTS

A Datasets and Evaluation Protocols 17
A.1 Training Corpus and Input Channels 17
A.2 Main-Table Evaluation Protocols 18

B Method and Implementation Details 18
B.1 Visual Pipeline 18
B.2 Sampling and Alignment Details 21
B.3 Training Strategy and Hyperparameters 23

C Additional Experimental Comparisons 24
C.1 Remote-Sensing Benchmark Results 24
C.2 Archived Additional Remote-Sensing Tasks 25
C.3 Comparison with Recent General-Purpose VLMs 25
C.4 General-Domain Transfer 26

D Ablations and Diagnostic Analyses 28
D.1 Feature Sources and Injection Schedules 28
D.2 Component Comparisons and Reference Settings 28
D.3 Additional Injection-Schedule Results 29
D.4 Gate Intervention and Activation Statistics 30
D.5 Full-Set Gate and Routing Diagnostics 31
D.6 Visual Dependence and Robustness Diagnostics 32
D.7 Efficiency and Scalability 33

E Qualitative Case Studies 34

A Datasets and Evaluation Protocols

A.1 Training Corpus and Input Channels

Training inputs and cross-task provenance. Table 6 summarizes the sources and instance counts of our six-task supervised fine-tuning (SFT) corpus. Among 208,441 audited source images, 206,938 (99.28%) are RGB and 1,503 (0.72%) are RGBA. We discard the alpha channel, so every model input has three RGB channels. Following recent remote-sensing vision–language studies (Pang et al., 2024, Li et al., 2024), we use the official task-specific splits and maintain image-level separation between training and evaluation data across tasks and datasets. To avoid inferring acquisition platforms from dataset names, Table 6 now reports the upstream imagery collection or sensor/platform used by the selected instances. VRSBench traces to DOTA-v2 and DIOR (Li et al., 2024). The three RSVQA entries are explicitly from RSVQA-LR and use Sentinel-2 imagery; RSVQA-HR, which uses USGS aerial orthophotos, is not included in this corpus (Lobry et al., 2020). The listed 720×720 RGB METER-ML instances correspond to the NAIP aerial product rather than its Sentinel-1 or Sentinel-2 products (Zhu et al., 2022). UCMerced-LandUse uses USGS aerial orthoimagery (Yang & Newsam, 2010). The DOTA-v2 and DIOR images used for VRSBench evaluation do not occur in the pretraining or SFT data. The NWPU-RESISC45 images used for NWPU-Captions evaluation are also excluded from training. Therefore, the reported evaluation sets have no image-level overlap with the training corpus.

Table 6: Overview of the fine-tuning corpus, grouped by task. Imagery source identifies the upstream collection or documented sensor/platform for the instances selected in this corpus, rather than assigning a platform type from the dataset name. The corpus spans six tasks—captioning, VQA, visual grounding, classification, instruction-based QA, and detection—with 293,002 labeled instances. Size denotes the task-specific instance count for each dataset. "–" indicates that metadata are not provided.

Task	Dataset	Size	Imagery Source	Image Size
Captioning	VRSBench	20,264	DOTA-v2/DIOR	512×512
VQA	VRSBench (VQA subset)	85,813	DOTA-v2/DIOR	512×512
	RSVQA-LR (comparison)	4,750	Sentinel-2 satellite	256×256
	RSVQA-LR (presence)	4,750	Sentinel-2 satellite	256×256
	RSVQA-LR (rural–urban)	500	Sentinel-2 satellite	256×256
Visual Grounding	DIOR-RSVG	53,982	DIOR (Google Earth)	800×800
	VRSBench (grounding)	36,313	DOTA-v2/DIOR	512×512
Classification	NWPU-RESISC45	4,995	Google Earth/public RS	256×256
	fMoW	4,969	Commercial satellite	679×563
	UCMerced-LandUse	2,100	USGS aerial orthoimagery	256×256
	METER-ML	1,497	NAIP aerial (RGB)	720×720
	RSITMD	484	Google Earth/RSICD	256×256
Instruction-based QA	fMoW (instruction)	1,872	Commercial satellite	679×563
Detection	DOTA-train	13,128	Multi-platform aerial	512×512
	DOTA-val	3,243	Multi-platform aerial	512×512
	DOTA-train (mini)	200	Multi-platform aerial	512×512
	DOTA-test	11,679	Multi-platform aerial	512×512
	FAIR1M	12,079	GF-2/Google Earth	569×568
	DIOR	6,880	Google Earth	800×800
	FAIR1M-train	6,662	GF-2/Google Earth	569×568
	FAIR1M-train (mini)	2,072	GF-2/Google Earth	569×568
	crowdAI (counting)	2,000	Satellite tiles	300×300
	crowdAI (buildings)	10,000	Satellite tiles	300×300
	Mts-WH	200	IKONOS satellite	512×512
	BANDON	200	Off-nadir aerial	2048×2048
	DeepGlobe LandCover	2,000	DigitalGlobe satellite	512×512
	FBP (GF-2 chips)	370	GF-2 satellite	512×512

A.2 Main-Table Evaluation Protocols

The main tables group methods by model family. Green, blue, yellow, and peach cells denote the four highest distinct scores in descending order. Tied scores receive the same color. Lavender identifies our model, with rank colors taking precedence in ranked cells. A dash (–) indicates that the corresponding result could not be obtained.

Classification evaluation protocols. The main-model classification results provide the common full-configuration reference. The reference classification accuracies are 94.37% on AID, 93.10% on WHU-RS19, 94.29% on NWPU-RESISC45, 67.29% on SIRI-WHU, 45.30% on EuroSAT, and 74.87% on METER-ML, with a macro-average of 78.20%. Tables 19, 20, and 22 use these values for the complete configuration wherever the corresponding datasets are reported. Table 12 reports the MF-RSVLM RSVQA reference used in the corresponding comparison.

Captioning evaluation protocols. The captioning results, metrics, and ranking interpretation follow Table 1. Appendix C.1.1 provides supplementary caption-output statistics only. VQA comparisons use a common nine-category average. Table 3 reports category-level accuracy over nine categories. Avg is the unweighted mean of the nine displayed category accuracies, and the ranking highlights cover all displayed models. Under this common calculation, Claude-Sonnet-4 and GeoChat have Avg values of 62.07 and 48.72, respectively.

B Method and Implementation Details

B.1 Visual Pipeline

Token-aligned dual-scale windows separate source field of view from encoder resolution. Figure 4 illustrates how the 672×672 detail canvas is decomposed into complementary coarse and fine source views. The 336×336 and 168×168 source windows specify different spatial fields of view, but every crop is resized to the same 336×336 vision-encoder input, corresponding to $1\times$ and $2\times$ local magnification. Both window grids use 50% overlap and share the same token lattice, preserving consistent coordinates during stitching while allowing coarse windows to retain broader structures and fine windows to emphasize small objects and local boundaries.

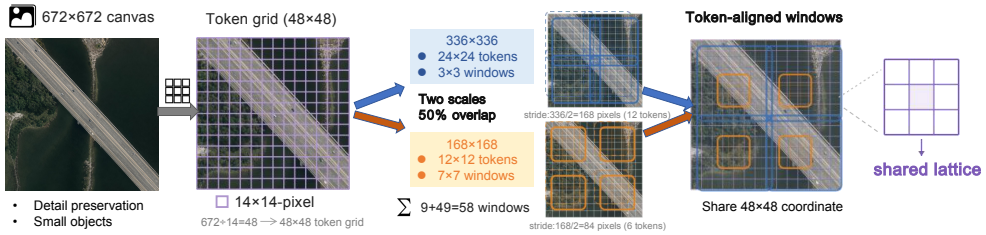

Figure 4: Token-aligned dual-scale overlapping windowing on a 672×672 canvas. The 336×336 and 168×168 source windows use 50% overlap on a shared 48×48 token lattice, producing 9 coarse and 49 fine crops; all crops are encoded at 336×336.

Phase subsampling converts stitched features into aligned detail sequences. After local features are stitched onto the shared canvas, they are reorganized into spatially aligned detail sequences, as illustrated in Figure 5. The four phase offsets form complementary checkerboard partitions, so each sequence preserves a regular sampling pattern while matching the length of the global visual stream. Applying this operation to two window scales and three vision transformer (ViT) layers produces a compact bank that retains both cross-scale and cross-layer information.

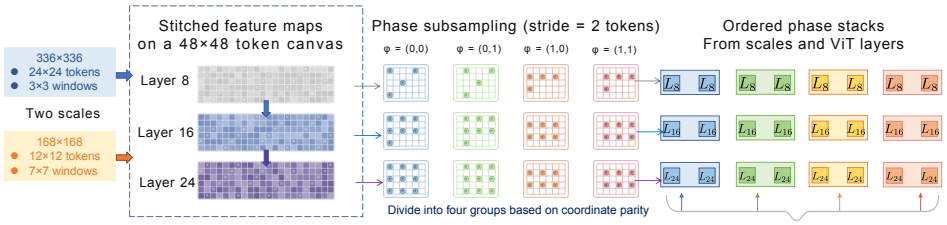

Figure 5: Construction of phase-subsampled detail stacks. Features from ViT layers $\{8, 16, 24\}$ are stitched on a 48×48 canvas and partitioned into four $f{=}2$ phase groups, yielding 24 aligned stacks across two window scales.

Layer-dependent routing selects visual details from the feature bank. Figure 6 shows how the model selects useful visual information from the detail bank at each injection layer. The current visual stream provides a router query, while every detail stack provides a corresponding key. Their similarity scores produce layer-dependent weights, allowing the model to emphasize the most relevant scale, ViT depth, and phase without applying expensive token-level attention over all stacks.

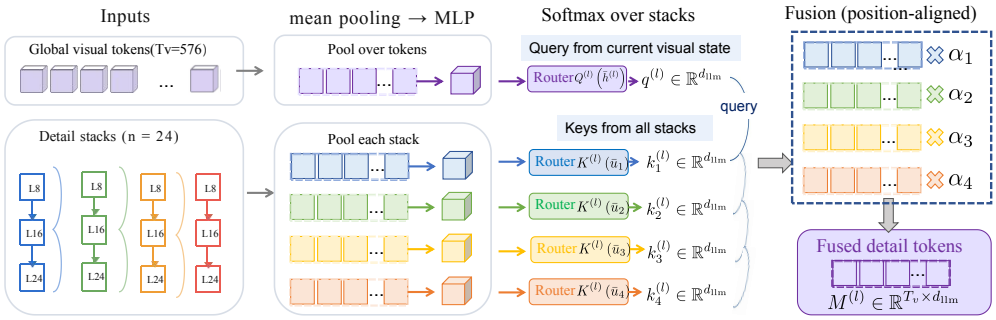

Figure 6: Adaptive detail-stack fusion at LLM layer l. Router scores derived from the current visual stream weight the 24 aligned stacks and produce fused detail tokens $M^{(l)}$ without full token-level cross-attention.

Gated residual updates refresh visual tokens at selected LLM layers. Figure 7 presents the final interaction between the fused detail tokens and the evolving large language model (LLM) representation. At each selected layer, a position- and channel-wise gate controls how much new detail is added through the residual pathway. The direct residual update is restricted to visual-token positions, while subsequent language-model processing can propagate the updated information. This repeated access to the detail bank is designed to address deep-layer visual forgetting.

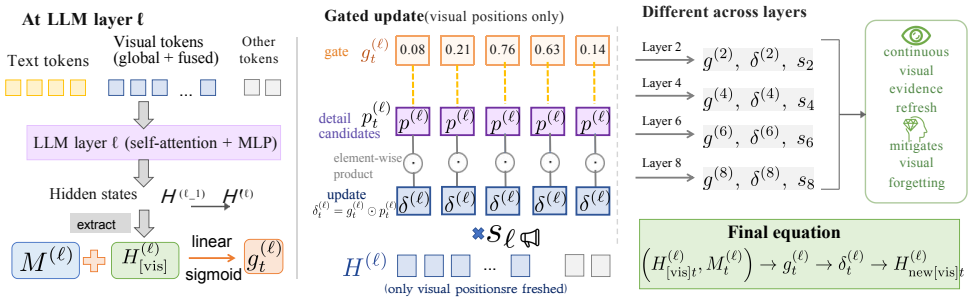

Figure 7: Layer-wise gated visual injection. At LLM layers $\{2, 4, 6, 8\}$, a position- and channel-wise gate modulates the fused detail update before residual addition to visual tokens, repeatedly refreshing visual information.

The comparison separates input processing from intermediate visual interaction. Table 7 compares twelve published architectures with MF-RSVLM across seven factors. LLaVA-1.5 (Liu et al., 2023b) uses a fixed-resolution view, while LLaVA-NeXT (Liu et al., 2024a) and DeepStack-L (Meng et al., 2024) encode global and grid-cropped views. Qwen2.5-VL (Bai et al., 2025b) and Qwen3-VL (Bai et al., 2025a) use dynamic native-resolution encoding with adjacent-feature merging. MiniCPM-V 2.5 (Yao et al., 2024) compresses adaptive slices using learned queries, and standard InternVL3.5 (Wang et al., 2025b) applies pixel shuffle before MLP projection. The table abbreviates these bridges as Merge, Queries, and Shuffle. BLIP-2 (Li et al., 2023a) uses a querying transformer (Q-Former), while OpenFlamingo (Awadalla et al., 2023) uses a Perceiver resampler. These alternatives differ in how they compress and expose visual evidence to the language model.

Table 7: Architecture comparison with models as rows and design factors as columns. RS denotes remote sensing. The "ViT levels" column distinguishes single-depth from multi-depth feature extraction. The "Image views" column describes spatial processing per image. "Stack routing" denotes LLM-state-conditioned weighting of a separate detail bank. The entries "None" and "No" refer to the specified module, not to the model's overall capabilities.

Model	Domain	Image views	Token bridge	ViT levels	LLM visual access	Stack routing	Update gate
BLIP-2 (Li et al., 2023a)	General	Fixed	Q-Former	Single	Input	No	None
LLaVA-1.5 (Liu et al., 2023b)	General	Fixed	MLP	Single	Input	No	None
LLaVA-NeXT (Liu et al., 2024a)	General	Grid	MLP	Single	Input	No	None
Qwen2.5-VL (Bai et al., 2025b)	General	Native	Merge	Single	Input	No	None
Qwen3-VL (Bai et al., 2025a)	General	Native	Merge	Multiple	Input + Add	No	None
MiniCPM-V 2.5 (Yao et al., 2024)	General	Slices	Queries	Single	Input	No	None
InternVL3.5 (Wang et al., 2025b)	General	Tiles	Shuffle	Single	Input	No	None
OpenFlamingo (Awadalla et al., 2023)	General	Fixed	Perceiver	Single	Cross-attn	No	Scalar
DeepStack-L (Meng et al., 2024)	General	Grid	MLP	Single	Input + Add	No	None
GeoChat (Kuckreja et al., 2024)	RS	Fixed	MLP	Single	Input	No	None
VHM (Pang et al., 2024)	RS	Fixed	Linear + MLP	Multiple	Input	No	None
EarthDial (Soni et al., 2025)	RS	Tiles	MLP	Single	Input	No	None
MF-RSVLM (Ours)	RS	Overlap	MLP	Multiple	Input + Add	Yes	Position + channel

Multilevel features, visual access, and gating are distinct design choices. GeoChat (Kuckreja et al., 2024) extends its fixed view to 504×504 pixels, while VHM (Pang et al., 2024) combines features from encoder layers 8, 16, and 24 before input-level projection. EarthDial (Soni et al., 2025) adds adaptive tiling and fusion for multi-spectral and temporal inputs. In the table, Input means delivery through the initial visual sequence, which remains available to subsequent self-attention. Add denotes additional residual injection within the LLM, as used by DeepStack-L and Qwen3-VL. OpenFlamingo instead introduces cross-attention at multiple decoder layers, abbreviated as Cross-attn, with learned scalar gates. Although its attention depends on decoder states, OpenFlamingo does not perform

the stack routing defined here. Thus, the comparison distinguishes intermediate interaction from multi-level encoder extraction and distinguishes scalar gating from position- and channel-dependent modulation.

MF-RSVLM combines a spatially organized detail bank with adaptive gated reuse. Our Overlap setting uses two source-window scales with a shared encoder resolution and three encoder depths to construct 24 phase-sampled stacks. At LLM layers $\{2, 4, 6, 8\}$, current visual states determine stack weights, and a position- and channel-wise gate modulates the fused residual update. This combination retains access to local evidence without token-level cross-attention over the entire bank. Stack routing excludes language-expert routing and input-resolution selection, and the gate column excludes ordinary gated feed-forward layers. The comparison concerns the cited architectures, including the standard InternVL3.5 and language-side DeepStack-L variants. The table compares architectural design choices. Downstream performance is reported in Tables 12 and 13, while Table 21 reports component settings.

B.2 Sampling and Alignment Details

Phase offsets partition the feature canvas into regularly sampled stacks. Let $I^{(s,\ell)}$ be the stitched feature map on a $T_{\text{base}} \times T_{\text{base}}$ token canvas for scale s and encoder block ℓ, where T_{base} is the number of tokens per side. Given a fixed positive integer downsampling factor f (default $f{=}2$), we enumerate phase offsets $\mathbf{r} = (a, b)$ with $a, b \in \{0, \ldots, f{-}1\}$ and define the sampling set

$$\Omega_{\mathbf{r}} \;=\; \{(u, v) \in \{0, \ldots, T_{\text{base}} - 1\}^2 \mid u \equiv a \pmod f,\; v \equiv b \pmod f\}. \tag{13}$$

For a batched feature tensor stored in batch–height–width–channel order, this corresponds to taking $I^{(s,\ell)}[:, a{::}f, b{::}f, :]$ in code. We omit the batch index in the equations. We form the detail stack for offset $\mathbf{r}$ by rasterizing $I^{(s,\ell)}$ on $\Omega_{\mathbf{r}}$ in row-major order (from top-left to bottom-right):

$$D_{\mathbf{r}}[t] \;=\; I^{(s,\ell)}[u_t, v_t], \qquad (u_t, v_t) \in \Omega_{\mathbf{r}} \text{ in raster order.} \tag{14}$$

Let $L_{\mathbf{r}} = |\Omega_{\mathbf{r}}|$ be the raw length of the stack. For $\mathbf{r} = (a, b)$, this length is $\lceil (T_{\text{base}} - a)/f \rceil \lceil (T_{\text{base}} - b)/f \rceil$. When f divides T_{base}, every phase stack has length $(T_{\text{base}}/f)^2$. We pad or truncate each stack as needed to match the global stream length T_v.

The detail bank combines every selected scale, encoder layer, and phase. Collecting the stacks across all selected scales, encoder layers, and phase offsets yields $\mathcal{B} = \{D_k\}$ with

$$N_{\text{stack}} \;=\; |\mathcal{S}| \cdot |\mathcal{L}| \cdot f^2, \tag{15}$$

where $\mathcal{S}$ is the set of sliding-window scales (two in our setup) and $\mathcal{L}$ is the set of encoder blocks selected for detail extraction (three blocks). At this stage, each $D_k \in \mathbb{R}^{T_v \times d_{\text{vit}}}$ preserves local spatial coherence within its own sampling grid (horizontal or vertical neighbors on the sampling grid are f canvas tokens apart), and all stacks remain at the vision-tower width d_{vit}. Global and detail sequences are projected to the LLM embedding space in a later step. When $f{=}2$, the token grid is partitioned into four disjoint "checkerboard" phases:

$$(0, 0),\; (0, 1),\; (1, 0),\; (1, 1), \tag{16}$$

where the token at coordinate (u, v) is assigned to group $(u \bmod 2,\; v \bmod 2)$. Thus, each (s, ℓ) produces 4 phase stacks of similar length (after padding/truncation, all have length T_v). Consequently, with $|\mathcal{S}| = 2$ scales and $|\mathcal{L}| = 3$ blocks, the four phase partitions yield a total of $N_{\text{stack}} = 2 \times 3 \times 4 = 24$ detail stacks.

Source-window sizes and strides are defined on a common token grid. Here, window size refers to the spatial field of view cropped from the detail canvas, not to the vision encoder's input resolution. We explain how to define these source-window sizes for a more general sliding-window mechanism. Let $X \in \mathbb{R}^{B \times C \times H \times W}$ be the input image and $\mathcal{C} \in \mathbb{R}^{B \times C \times S \times S}$ the high-resolution canvas obtained by resizing X to $S \times S$. Assuming that the patch size p divides S, a vision patch of side length p pixels induces a token grid with

$$T_{\text{base}} \;=\; S/p \;\in\; \mathbb{N} \tag{17}$$

tokens per side. We denote by w the base window side used by the encoder input (e.g., $w{=}336$). For a scale s with nominal window side $\hat{w}_s = \beta_s\, w$ ($\beta_s{>}0$), we derive the snapped window size and the

corresponding stride so that windows at all scales are aligned on the token grid and support 50% overlap.

Snapping window widths to even divisors supports aligned integer strides. Directly rounding $\hat{w}_s/p$ to the nearest integer may prevent exact boundary coverage or yield a non-integer token stride for 50% overlap. To guarantee cross-scale alignment and integer strides, we restrict the number of tokens per side to a nonempty admissible set

$$\mathcal{D} \subseteq \{d \in \mathbb{N} \mid d \mid T_{\text{base}},\ d \text{ is even},\ d \geq d_{\min}\}. \tag{18}$$

Divisibility aligns the window and canvas grids, while even token widths allow integer strides with exact 50% overlap. Given the nominal token width $x = \hat{w}_s/p$, we define the snap operator

$$t_s = \operatorname{snap}\big(x, \mathcal{D}\big) = \arg\min_{d \in \mathcal{D}} |x - d|, \tag{19}$$

breaking ties toward the larger d (to avoid under-coverage).

Each snapped token width determines the pixel window and its half-window stride. With t_s fixed, the actual window side and the strides are

$$w_s = t_s\, p, \qquad \tau_s = \frac{t_s}{2}, \qquad \Delta_s = \tau_s\, p = \frac{t_s}{2}\, p, \tag{20}$$

where w_s is in pixels, τ_s is the token-grid stride (exact 50% overlap), and Δ_s is the pixel stride. By construction, $\tau_s, \Delta_s \in \mathbb{N}$. This overlap supports Hann-weighted blending between adjacent windows.

Aligned window starts cover the canvas up to its boundary. Let $\mathcal{J}_s = \{0, \tau_s, 2\tau_s, \ldots, T_{\text{base}} - t_s\}$ be the set of token-grid start positions per side (1D) for scale s. Since $t_s \,|\, T_{\text{base}}$ and $\tau_s = t_s/2$,

$$T_{\text{base}} - t_s = 2\Big(\frac{T_{\text{base}}}{t_s} - 1\Big)\tau_s \in \mathbb{Z}_{\geq 0} \cdot \tau_s, \tag{21}$$

so the last window ends exactly at the canvas boundary. For two scales $s \neq s'$ with $t_s, t_{s'} \in \mathcal{D}$, all start positions are integer multiples of the corresponding token strides τ_s and $\tau_{s'}$. Since the window widths are also integers in token units, all window boundaries align with the shared token grid, providing consistent coordinates for cross-scale aggregation. When $t_s \,|\, T_{\text{base}}$ and $\tau_s = t_s/2$,

$$N_{\text{side}}(s) = 1 + \frac{T_{\text{base}} - t_s}{\tau_s} = 2 \cdot \frac{T_{\text{base}}}{t_s} - 1. \tag{22}$$

Hence the total number of windows at scale s is $N_{\text{side}}(s)^2$.

Boundary-safe Hann blending prevents zero normalization weights. For a window containing t_s tokens per side, we sample the one-dimensional Hann function at token-cell centers rather than at its zero-valued endpoints:

$$h_s[n] = \sin^2\left(\frac{\pi(n + \frac{1}{2})}{t_s}\right), \qquad n = 0, \ldots, t_s - 1. \tag{23}$$

Thus, $h_s[n] > 0$ at every sampled token. For a window (i, j) covering canvas coordinate (u, v), the separable two-dimensional weight is

$$W_{i,j}[u, v] = h_s[u - i\tau_s] \cdot h_s[v - j\tau_s]. \tag{24}$$

The implementation initializes a feature accumulator $A^{(s,\ell)}$ and a scalar weight accumulator $Z^{(s)}$ to zero and applies the following updates for every window and covered canvas position:

$$\begin{aligned} A^{(s,\ell)}[u, v] &\mathrel{+}= W_{i,j}[u, v] \cdot \widetilde{H}^{(s,\ell)}_{i,j}[u - i\tau_s,\, v - j\tau_s], \\ Z^{(s)}[u, v] &\mathrel{+}= W_{i,j}[u, v], \\ I^{(s,\ell)}[u, v] &= \frac{A^{(s,\ell)}[u, v]}{\max\big(Z^{(s)}[u, v], \varepsilon_{\text{num}}\big)}. \end{aligned} \tag{25}$$

Here, $\varepsilon_{\text{num}} > 0$ is a small numerical safeguard for the working precision. Because the center-sampled Hann weights are strictly positive and the aligned windows cover every canvas position, $Z^{(s)}[u, v]$ is positive throughout the canvas; the clamp protects only against finite-precision underflow. In pseudocode, the final normalization is implemented as `I = A / Z.clamp_min(eps)`.

Window offsets map local tokens to shared canvas coordinates. Let (i, j) index a window in token units by its top-left coordinate $(i\tau_s,\ j\tau_s)$. A local token $(u', v') \in \{0, \ldots, t_s - 1\}^2$ inside this window maps to the global token coordinate

$$(u, v) \ = \ (i\tau_s + u',\ j\tau_s + v'). \tag{26}$$

Equivalently, in pixel coordinates, window corners are at $(i\Delta_s,\ j\Delta_s)$ and the window side is $w_s = t_s p$.

Naive rounding does not guarantee exact boundary coverage and integer strides. Let $x = \hat{w}_s/p$ and $t_s^{\text{round}} = \text{round}(x)$. Rounding does not guarantee either that (i) $t_s^{\text{round}} \mid T_{\text{base}}$ (the chosen sufficient condition for exact boundary coverage) or that (ii) t_s^{round} is even (the condition for an integer token stride at 50% overlap). Snapping to $\mathcal{D}$ enforces these grid constraints and gives integer window starts and overlap strides for the subsequent aggregation procedure.

The adopted configuration produces nine coarse and 49 fine windows. The following example illustrates the adopted window sizes and strides. Let $S{=}672$ and $p{=}14$, so that $T_{\text{base}}{=}48$. Choose $\mathcal{D} = \{8, 12, 16, 24, 48\}$ (even divisors with $d_{\min}{=}8$). For $\hat{w}_s \in \{168, 336\}$:

- $\hat{w}_s{=}336$: $x{=}336/14{=}24 \Rightarrow t_s{=}24$, $\tau_s{=}12$, $w_s{=}336$, $\Delta_s{=}168$, $N_{\text{side}}{=}2 \cdot 48/24 - 1{=}3$.
- $\hat{w}_s{=}168$: $x{=}12 \Rightarrow t_s{=}12$, $\tau_s{=}6$, $w_s{=}168$, $\Delta_s{=}84$, $N_{\text{side}}{=}2 \cdot 48/12 - 1{=}7$.

Both window grids align with the same token lattice, providing shared coordinates for Hann-weighted blending across overlapping windows.

Window granularity determines the token count on the aligned feature grid. At each scale s, the number of windows is $N_{\text{win}}(s) = (2T_{\text{base}}/t_s - 1)^2$ and each window contributes t_s^2 tokens per selected encoder layer. Thus, the raw token count per scale–layer pair is

$$N_{\text{tok}}(s, \ell) \ = \ \left(2\frac{T_{\text{base}}}{t_s} - 1\right)^2 t_s^2 \ = \ \left(2T_{\text{base}} - t_s\right)^2, \tag{27}$$

which decreases with larger t_s (coarser windows). This explains the common design of mixing a coarse (t_s large) and a fine (t_s small) scale to balance detail and cost.

B.3 Training Strategy and Hyperparameters

Pretraining and supervised fine-tuning use different parameter-update settings. Figure 8 summarizes the two training stages. The multilayer perceptron (MLP) projects visual features into the language-model embedding space. Table 8 consolidates the model configuration and optimization settings. Both global and per-GPU batch sizes are reported.

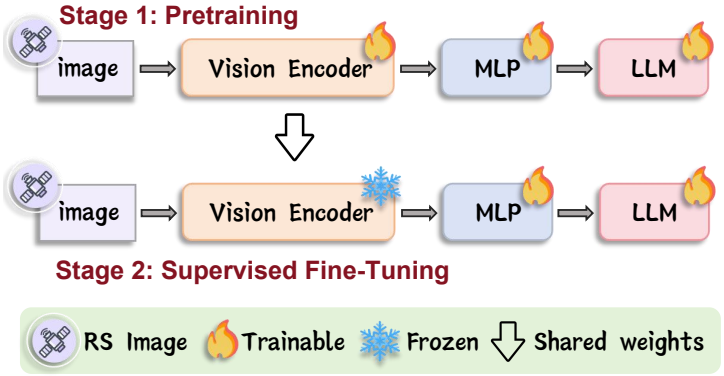

Figure 8: Two-stage training strategy for MF-RSVLM. In Stage 1 (pretraining), a CLIP-based vision encoder extracts features from remote-sensing images, and an MLP projects these features into the LLM embedding space. The vision encoder, MLP projector, and LLM are all trainable. In Stage 2 (supervised fine-tuning), the vision encoder is frozen, while the MLP projector, fusion module, routing module, gating module, and LLM are optimized on the instruction-tuning corpus.

Table 8: Hyperparameters and experimental settings for MF-RSVLM.

Category	Parameter	Value
Model Architecture		
Vision Encoder	Backbone	CLIP ViT-L/14@336
Language Model	Backbone	Vicuna-v1.5 (7B)
Resolution	Global Input Size	336×336
	High-Res Canvas Size	672×672
Multi-Scale	Sliding Window Sizes	336×336, 168×168
	ViT Extraction Layers	$\{8, 16, 24\}$
	LLM Injection Layers	$\{2, 4, 6, 8\}$
	Downsampling Factor (f)	2
Stage 1: Pretraining		
Data	Dataset Size	$\sim$1.4M pairs
Training	Epochs	1
	Global Batch Size	96 (12 per GPU)
	Learning Rate	2×10^{-5}
	Warmup Ratio	0.03
	Weight Decay	0
Optimization	Trainable Modules	CLIP ViT-L/14@336, MLP projector, Vicuna-v1.5 (7B)
Resources	Hardware	$8\times$ NVIDIA A6000 (48GB)
	Training Time	$\sim$42 hours
Stage 2: Supervised Fine-Tuning (SFT)		
Data	Dataset Size	293,002 instances
Training	Epochs	2
	Global Batch Size	48 (6 per GPU)
	Learning Rate	2×10^{-5}
	Warmup Ratio	0.03
	Weight Decay	0
Optimization	Trainable Modules	MLP projector, fusion, routing, gating, Vicuna-v1.5 (7B)
	Frozen Modules	Vision Encoder
Resources	Hardware	$8\times$ NVIDIA A6000 (48GB)
	Training Time	$\sim$38 hours

C Additional Experimental Comparisons

C.1 Remote-Sensing Benchmark Results

C.1.1 Complete Caption Metrics and Output Profiles

Caption outputs vary in length despite a shared prompt. Table 9 repeats the Qwen3-VL-8B metrics used in the main comparison for all five displayed test splits. Table 10 summarizes 14,889 captions from the three listed models as a supplementary output-format diagnostic. The captioning results and their interpretation remain those reported in Table 1.

Table 9: Complete Qwen3-VL-8B caption results for the five datasets displayed in Table 1 (scores $\times 100$). N is the number of images. All four metrics are reported for each dataset.

Dataset	N	BLEU-4	METEOR	CIDEr	ROUGE-L
UCM	210	3.44	14.92	26.20	28.47
RSICD	1,093	2.93	14.29	14.18	22.95
RSITMD	452	1.23	12.21	11.56	20.13
NWPU	3,150	3.01	14.37	16.26	24.64
Sydney	58	3.20	15.13	13.46	30.12

Table 10: Output-form statistics for 14,889 completed captions from the three listed models. The Short column counts captions containing at most three words. The Multi-terminator column counts captions containing multiple sentence-ending punctuation marks.

Model	N	Mean words	Median words	Short	Multi-terminator
Qwen3-VL-8B	4,963	15.14	15	0	0
RSCoVLM-7B	4,963	10.05	10	0	0
GeoSolver	4,963	7.45	2	3,761	391

C.2 Archived Additional Remote-Sensing Tasks

Additional remote-sensing evaluations cover detection, numerical estimation, grounding, polygon prediction, and multilabel classification. Table 11 summarizes the completed historical evaluation. Grounding precision is 19.55% overall and 0.15%, 2.16%, and 34.88% for small, medium, and large objects, respectively, with higher recorded precision for larger objects. We report each task using its recorded metric and unit. Some numerical scorers exclude unparseable answers. For object counting, mean absolute percentage error (MAPE) excludes cases with zero-valued targets. Raw N counts prediction rows. Average CP denotes average classwise precision, not image-level accuracy. The inventory documents completed outputs.

Table 11: Complete auxiliary task inventory from a single completed historical run. Raw N is the stored prediction-row count. Metric labels and units follow the available source records.

Task	Raw N	Recorded metric	Value
DOTA object presence	348	Legacy perception (%)	81.03
DOTA absolute position	112	Legacy perception (%)	26.79
DOTA relative position	242	Legacy perception (%)	22.31
DOTA object counting	120	MAE (objects), MAPE (%)	6.2667, 37.94
Image type	20	Legacy perception (%)	95.00
DOTA/FAIR1M color	200	Legacy perception (%)	85.00
DOTA/FBP ground sample distance	360	MAE (m/pixel)	0.301306
DIOR-RSVG grounding	7,500	Recorded precision (%)	19.55
crowdAI building polygons	200	Mean IoU, CIoU (%)	79.42, 70.72
DOTA object measurement	200	MAE (annotation unit: m)	16.00046
GID multilabel land cover	6,240	Average CP (%)	43.55
FBP multilabel land cover	6,240	Average CP (%)	75.04

C.3 Comparison with Recent General-Purpose VLMs

The reported comparison with Qwen3-VL shows dataset-dependent gains. Table 12 compares MF-RSVLM with Qwen3-VL on four scene-classification datasets and three RSVQA subtasks. The classification values match the corresponding results in Table 2. MF-RSVLM exceeds Qwen3-VL by 27.70 and 25.37 percentage points on AID and NWPU, respectively. Qwen3-VL remains ahead on SIRI-WHU by 0.34 percentage points. The reported RSVQA-LR scores are higher than those of Qwen3-VL by 14.63 percentage points on Comparison and 20.63 percentage points on Presence. We adopt the RSVQA results in this table as the common MF-RSVLM reference: 82.80% on HR Comparison and 89.16% on both LR Comparison and LR Presence. Results are reported under the respective evaluation protocols.

Table 12: Reported accuracy (%) comparison with Qwen3-VL (8B). HR-C, LR-C, and LR-P denote RSVQA-HR Comparison, RSVQA-LR Comparison, and RSVQA-LR Presence. Classification values for both methods match Table 2.

Method	**AID**	**NWPU**	**SIRI**	**RS19**	**HR-C**	**LR-C**	**LR-P**
Qwen3-VL (8B)	66.67	68.92	67.63	88.60	76.70	74.53	68.53
MF-RSVLM	94.37	94.29	67.29	93.10	82.80	89.16	89.16

MF-RSVLM achieves higher accuracy than InternVL3.5 in eleven of the twelve reported evaluations. To examine task-level performance and differences associated with multi-scale inputs,

we compare MF-RSVLM with InternVL3.5 and a variant without the multi-scale branch. Table 13 reports six VRSBench categories and six scene-classification datasets. MF-RSVLM achieves higher accuracy than InternVL3.5 on all six classification datasets and five VQA categories, but trails it on Position by 8.39 percentage points. Relative to the no-multi-scale variant, the largest VQA improvements occur on Existence and Position, at 55.35 and 15.55 percentage points, respectively. All six classification scores also improve. The table reports configuration-level performance differences across the evaluated tasks.

Table 13: Accuracy (%) comparison with InternVL3.5 and the MF-RSVLM variant without the multi-scale branch on VRSBench VQA and scene classification. Higher accuracy ranks first.

Method	VRSBench VQA						Scene Classification					
	Existence	Position	Scene	Color	Shape	Direction	AID	RS19	NWPU	SIRI	EuroSAT	METER
without multi-scale	34.88	48.71	56.70	57.94	56.85	36.89	87.07	91.20	91.62	62.12	38.85	68.67
InternVL3.5 (Wang et al., 2025b)	83.14	72.65	44.51	19.01	22.35	38.57	73.80	91.50	92.35	66.67	45.20	50.80
MF-RSVLM	90.23	64.26	59.18	64.48	62.68	45.91	94.37	93.10	94.29	67.29	45.30	74.87

C.4 General-Domain Transfer

We evaluate zero-shot transfer on four general-domain multimodal benchmarks. Table 14 reports accuracy on MMStar, MMBench, ConBench, and MME-RealWorld-Lite, using 1,500, 4,329, 3,000, and 1,919 samples, respectively. Accuracies range from 24.75% to 33.01%, with the model attaining its highest reported accuracy on MMBench. The evaluation reports benchmark-specific zero-shot results. Appendix C.4.1 reports a separate local MMMU-Pro evaluation, which is not pooled with these four benchmark results.

Table 14: Preliminary zero-shot transfer to general-domain multimodal benchmarks.

Dataset	Samples	Accuracy (%)
MMStar	1,500	25.13
MMBench	4,329	33.01
ConBench	3,000	32.03
MME-RealWorld-Lite	1,919	24.75

C.4.1 MMMU-Pro Transfer Across Subject Areas

The local MMMU-Pro evaluation provides a subject-level transfer profile. Table 15 reports results for all thirty subjects in the completed local evaluation, with four answer options per question. The evaluation comprises 1,730 samples, with 367 answers scored as correct (21.21%). Rates range from 10.00% in Physics to 36.54% in Literature, with only 50–60 samples per subject. Subject-level rates provide a descriptive transfer profile across the thirty evaluated subjects. Overall accuracy is 21.21% against a 25% uniform-guessing reference. The evaluation retains its original answer-extraction decisions.

Table 15: All thirty subject areas in the completed local MMMU-Pro evaluation. N denotes the number of evaluated samples. Counts of correct answers reproduce stored evaluation decisions, rates are percentages.

Subject	N	Correct	Rate (%)
Accounting	58	18	31.03
Agriculture	60	10	16.67
Architecture and Engineering	60	11	18.33
Art	53	10	18.87
Art Theory	55	12	21.82
Basic Medical Science	52	18	34.62
Biology	59	10	16.95
Chemistry	60	10	16.67
Clinical Medicine	59	14	23.73
Computer Science	60	11	18.33
Design	60	15	25.00
Diagnostics and Laboratory Medicine	60	8	13.33
Economics	59	9	15.25
Electronics	60	12	20.00
Energy and Power	58	8	13.79
Finance	60	14	23.33
Geography	52	14	26.92
History	56	17	30.36
Literature	52	19	36.54
Manage	50	14	28.00
Marketing	59	8	13.56
Materials	60	15	25.00
Math	60	11	18.33
Mechanical Engineering	59	11	18.64
Music	60	9	15.00
Pharmacy	57	20	35.09
Physics	60	6	10.00
Psychology	60	10	16.67
Public Health	58	11	18.97
Sociology	54	12	22.22
Overall	1,730	367	21.21

C.4.2 General-Domain Consistency and Answer Extraction

Question-format and answer-extraction breakdowns provide context for overall accuracy. Tables 16, 17, and 18 provide breakdowns of the completed general-domain evaluations in Table 14. The evaluated ConBench set contains 860 complete groups with all three question formats, 140 groups with two formats, and 140 single questions, totaling 3,000 questions. All-format accuracy is 1.86% (16/860 complete groups). This all-format consistency rate uses the 860 complete groups as its denominator, not the 1,000 questions in each individual format. The MME-RealWorld-Lite evaluation yields 475 correct answers among 1,919 questions (24.75%). By question type, 297 of 1,169 perception questions (25.41%) and 178 of 750 reasoning questions (23.73%) are answered correctly. Table 17 provides a separate breakdown by domain. The answer-extraction procedure maps 1,385 outputs through letter extraction, 251 through exact matching, and 168 through containment matching. The remaining 115 unmapped outputs (5.99%) are retained in the evaluation denominator. These extraction counts describe output mapping. These breakdowns characterize question-format sensitivity and answer extraction in the evaluated checkpoint.

Table 16: ConBench by question format. Every format contains 1,000 questions, the total is 961/3,000 correct (32.03%).

Question format	Correct / N	Accuracy (%)
Multiple choice	308 / 1,000	30.80
Yes/no	527 / 1,000	52.70
Open Q/A	126 / 1,000	12.60

Table 17: MME-RealWorld-Lite subtask breakdown. The five rows partition all 1,919 samples, 475 are correct (24.75%).

Subtask	Correct / N	Accuracy (%)
Autonomous Driving	191 / 750	25.47
Diagram and Table	47 / 200	23.50
Monitoring	109 / 469	23.24
OCR with Complex Context	92 / 350	26.29
Remote Sensing	36 / 150	24.00

Table 18: MME-RealWorld-Lite answer-extraction breakdown for all 1,919 outputs. Counts describe extraction routes, not numbers of correct answers. Unmapped outputs remain in the evaluation denominator.

Extraction route	Answer count
Letter	1,385
Exact match	251
Containment	168
Unmapped	115
Total	1,919

D Ablations and Diagnostic Analyses

D.1 Feature Sources and Injection Schedules

Comparison of ViT Layers and Window Scales. To examine feature depth and spatial scale, we compare three extraction configurations with LLM injection at layers 2, 4, 6, and 8. Table 19 uses the main-model classification results for the complete configuration. This configuration uses ViT layers 8, 16, and 24 with both window scales and achieves a mean accuracy of 88.25%, compared with 82.36% for ViT layer 24 with both window scales and 80.96% for the three ViT layers with only the 336×336 window scale. The corresponding differences are 5.89 and 7.29 percentage points, respectively, with improvements on all seven benchmarks. Mean accuracy is the unweighted average of the four classification and three RSVQA subtask accuracies. Appendix B explains alignment and stack construction.

Comparison of LLM Injection Layers. To investigate where visual updates are most useful, we vary the LLM injection schedule while fixing ViT layers 8, 16, and 24 and using both the 336×336 and 168×168 window scales. Table 20 compares single-site and multi-site schedules across seven benchmark columns. Injection at layers 2, 4, 6, and 8 reaches 88.25% mean accuracy, exceeding the layer-8-only schedule by 4.51 percentage points and the schedule using layers 2, 4, and 6 by 5.55 percentage points. However, all three tested partial multi-site schedules have lower mean accuracy than the strongest single-site setting, which uses layer 8. The four-site schedule achieves the highest mean accuracy among the tested configurations on the seven benchmarks listed here. Appendix D.6 complements these configuration comparisons with leave-one-injection-out diagnostics, which probe the frozen checkpoint without independently retraining the architectures.

D.2 Component Comparisons and Reference Settings

Component comparisons use the main-model full configuration as a common reference. Table 21 reports full-configuration minus variant accuracy differences. All six classification differences for the

Table 19: Accuracy (%) for different ViT feature-extraction layers and window scales, with LLM injection fixed at layers 2, 4, 6, and 8. Checkmarks indicate selected layers and window scales. Mean is the unweighted average of the seven benchmark accuracies. Classification scores for the complete configuration match the main-model results in Table 2.

ViT layers			Window Size		Benchmarks							
8	**16**	**24**	336×336	168×168	**METER-ML**	**NWPU**	**WHU-RS19**	**AID**	**HR-Comp**	**LR-Comp**	**LR-Pre**	**Mean**
		✓	✓	✓	66.55	90.41	88.80	85.63	77.10	82.84	85.16	82.36
✓	✓	✓	✓		63.89	89.65	88.10	83.73	74.80	84.21	82.32	80.96
✓	✓	✓	✓	✓	74.87	94.29	93.10	94.37	82.80	89.16	89.16	88.25

Table 20: Accuracy (%) for different LLM injection schedules, with ViT layers 8, 16, and 24 and both window scales fixed. Checkmarks indicate selected injection layers. Mean is the unweighted average of the seven benchmark accuracies. The full four-site reference is shared with Table 19 and uses the main-model classification scores.

LLM layers				Benchmarks							
2	**4**	**6**	**8**	**METER-ML**	**NWPU**	**WHU-RS19**	**AID**	**HR-Comp**	**LR-Comp**	**LR-Pre**	**Mean**
✓				65.84	91.90	89.00	85.00	74.90	85.58	86.11	82.62
	✓			65.49	90.19	89.40	85.53	74.90	84.00	85.79	82.19
		✓		64.78	89.84	88.80	83.03	76.90	83.26	85.37	81.71
			✓	70.09	92.06	89.70	87.60	74.50	85.16	87.05	83.74
✓			✓	64.96	91.17	88.50	86.27	75.90	85.37	82.95	82.16
✓	✓	✓		66.37	90.38	88.60	86.60	77.90	85.26	83.79	82.70
	✓	✓	✓	66.19	89.14	89.40	86.10	77.00	83.89	84.32	82.29
✓	✓	✓	✓	74.87	94.29	93.10	94.37	82.80	89.16	89.16	88.25

multi-scale branch can be reproduced from the main-model and no-multi-scale rows in Table 13. For the cross-layer, window-scale, and injection contrasts, the differences on the seven shared benchmarks can be reproduced from the full reference and the corresponding variants in Tables 19 and 20. For recurrent injection, the SIRI and EuroSAT differences follow the full reference and layer-8-only scores in Table 22: $67.29 - 64.42 = 2.87$ and $45.30 - 41.80 = 3.50$ percentage points, respectively. Each Mean entry is the unweighted average of the nine displayed differences.

Table 21: Full-configuration minus variant accuracy differences, in percentage points, grouped by the available reference comparisons. Positive values favor the corresponding full configuration, negative values favor the variant. Mean is the unweighted average of the nine displayed differences.

Factor	Comparison	Scene classification						VQA			Mean
		AID	RS19	NWPU	SIRI	EuroSAT	METER	HR-C	LR-C	LR-P	
(a) Six classification differences match the main-model comparison											
Multi-scale branch	Full vs. without multi-scale	7.30	1.90	2.67	5.17	6.45	6.20	2.60	1.37	0.74	3.82
(b) Full-configuration differences use the main-model reference											
Cross-layer visual features	Full vs. ViT-24	8.74	4.30	3.88	8.75	1.50	8.32	5.70	6.32	4.00	5.72
Effective visual coverage	Full vs. 336-only windows	10.64	5.00	4.64	10.75	1.35	10.98	8.00	4.95	6.84	7.02
Recurrent visual injection	Full vs. injection-8	6.77	3.40	2.23	2.87	3.50	4.78	8.30	4.00	2.11	4.22
(c) Training-data-scale comparison											
Training-data scale	Full vs. half-data	5.44	1.30	2.23	0.66	-0.45	7.44	1.90	0.95	1.48	2.33

D.3 Additional Injection-Schedule Results

The highest-scoring injection schedule varies across the additional tasks. Table 22 reports the schedules from Table 20 on two additional classification datasets, SIRI-WHU and EuroSAT, and two RSVQA subtasks, HR Presence and LR Rural. The reported scores span 58.79–67.29%, 36.65–45.30%, 51.40–60.20%, and 76.00–91.00%, respectively. The four-site schedule achieves the highest SIRI-WHU and EuroSAT accuracies, at 67.29% and 45.30%. Injection at layer 6 only

yields the highest HR Presence score, while injection at layer 8 only yields the highest LR Rural score. Together with Table 20, these results highlight the four-site schedule's leading classification performance alongside task-specific advantages of individual injection depths.

Table 22: Additional results (%) for seven alternative injection schedules and the four-site reference. The four-site classification scores match the main-model results in Table 2. HR Presence and LR Rural denote the corresponding RSVQA-HR and RSVQA-LR subtasks. Higher scores rank first.

LLM injection layers	SIRI-WHU	EuroSAT	HR Presence	LR Rural
2	61.42	36.65	55.50	80.00
4	60.33	38.05	55.20	76.00
6	63.83	39.65	60.20	80.00
8	64.42	41.80	51.40	91.00
2, 8	58.79	41.20	52.60	79.00
2, 4, 6	59.58	39.45	56.60	88.00
4, 6, 8	63.38	41.60	53.10	79.00
2, 4, 6, 8 (reference)	67.29	45.30	54.80	80.00

D.4 Gate Intervention and Activation Statistics

Learned gates outperform both constant settings on six of the seven datasets. We replace the learned gate with zero or one during inference and evaluate seven scene-classification datasets (Table 23). Setting the gate to zero suppresses the gated detail update, setting it to one removes adaptive gate modulation while retaining the residual addition and learned layer-wise scale. On AID, accuracy falls from 94.37% with the learned gate to 87.77% with gate zero and 16.20% with gate one. On WHU-RS19, gate one retains 90.90%, whereas gate zero yields 21.00%, compared with 93.10% for the learned gate. The learned gate performs best on all six datasets other than fMoW. On the local fMoW split, gate one scores 52.70%, slightly above the learned gate's 52.35%. This split is included only as a diagnostic. The intervention results support the benefit of adaptive modulation in the evaluated model.

Table 23: Accuracy (%) under inference-time gate interventions on seven scene-classification datasets. The local fMoW split is reported for diagnostic purposes only. Higher accuracy ranks first.

Gate setting	**AID**	**RS19**	**NWPU**	**SIRI**	**EuroSAT**	**METER**	**fMoW**
Learned gate	94.37	93.10	94.29	67.29	45.30	74.87	52.35
Gate $= 0$	87.77	21.00	21.81	11.46	5.20	20.35	9.45
Gate $= 1$	16.20	90.90	21.49	10.75	4.65	19.82	52.70

Learned gates have nonuniform activations with similar dataset-level summaries. Table 24 summarizes gate activations separately for each evaluation dataset, reporting the mean, variance, median, 90th percentile, and observed range. Mean activations lie between 0.10361 and 0.11154, while medians are lower, between 0.05689 and 0.06372. The 90th percentiles range from 0.2556 to 0.2747. These summaries indicate predominantly small gate values with an upper tail of larger activations and a broadly similar distribution across datasets. Gate values describe modulation coefficients, not residual-update magnitudes, which also depend on the projected detail features and the learned layer-wise scale.

Table 24: Learned gate activation statistics across the evaluated classification datasets. Count denotes the approximate number of scalar gate elements, reported in billions, rather than the number of images.

Metric	AID	RS19	NWPU	SIRI	EuroSAT	METER	fMoW
Mean	0.10361	0.10626	0.10377	0.10531	0.10606	0.10575	0.11154
Variance	0.01628	0.01661	0.01639	0.01643	0.01676	0.01654	0.01723
Count ($\times 10^9$)	28.3	9.4	29.7	22.6	18.9	5.3	18.9
Minimum	0.00011	0.00008	0.00008	0.00008	0.00008	0.00010	0.00009
Median	0.05689	0.05884	0.05689	0.05835	0.05884	0.05835	0.06372
90th percentile	0.2556	0.2625	0.2571	0.2600	0.2635	0.2615	0.2747
Maximum	1.0	1.0	1.0	1.0	1.0	1.0	1.0

D.5 Full-Set Gate and Routing Diagnostics

Evaluation scope and aggregation. We extend the gate and routing diagnostics to all 12,115 evaluation examples from AID, EuroSAT, METER-ML, NWPU-RESISC45, SIRI-WHU, and WHU-RS19, without subsampling. Each example is scored under 11 conditions: the intact model, four leave-one-injection-layer-out conditions, and six leave-one-routing-group-out conditions. This yields 133,265 scores, with no missing or invalid entries in the completed result summary. Full-set aggregates are weighted by sample count rather than assigning equal weight to each dataset. Tables 25 and 26 report all statistics from this evaluation; the main-text tables provide compact summaries of allocation statistics and removal effects.

Removal-effect definition. For the correct answer $y^\star$, let $\bar{\ell}(y^\star)$ denote its mean token log probability. We report $\Delta \log p = \bar{\ell}_{\text{full}}(y^\star) - \bar{\ell}_{\text{ablated}}(y^\star)$, averaged across evaluation examples, in nats/token. For readability, both the main-text and appendix tables report $10^3 \Delta \log p$, converting the unit to millinats/token without changing the underlying measurements. A positive gap means that removal reduces support for the correct answer; a negative gap means that removal increases it. These gaps are not changes in prediction accuracy. Layer removal disables the injection at that LLM layer. Each routing group comprises the four phase stacks associated with one window size and encoder layer; group removal renormalizes the remaining routing weights. Note that this full-minus-ablated sign convention is opposite to the intervention-minus-full convention in the earlier 120-image leave-one-injection-out table. The different evaluation scope and aggregation also prevent direct numerical comparison with those subset results.

Table 25: Full-set gate statistics and injection-layer removal effects. SD denotes standard deviation and P90 the 90th percentile. Gate statistics are dimensionless; the last column reports $10^3 \Delta \log p$ in millinats/token, with positive values indicating reduced correct-answer support after removal.

LLM layer	Mean	SD	Median	P90	$10^3 \Delta \log p$
2	0.055392	0.002789	0.055016	0.058936	$+15.261$
4	0.160820	0.007959	0.160121	0.171116	$+24.922$
6	0.104015	0.006350	0.104139	0.111830	$+2.643$
8	0.100145	0.007225	0.099252	0.110097	$+1.436$

Gated visual updates change correct-answer support across layers. Removing any of the four tested injections changes correct-answer support on average, with the largest mean gaps at layers 4 and 2. Layer 4 has the largest mean gate activation and mean removal effect ($+24.922$ millinats/token). Layer 2 has the smallest mean activation but the second-largest removal effect ($+15.261$), showing that gate magnitude alone does not determine contribution. Layers 4 and 2 also retain the two largest mean removal effects under equal weighting of the six datasets, although this ordering varies across individual datasets. These statistics characterize layer-wise sensitivity under the frozen checkpoint.

Routing weights and removal effects vary across detail groups. The groups with 336 pixel windows at ViT layer 8 and with 168 and 336 pixel windows at ViT layer 24 jointly receive 97.32% of the mean routing weight. The first group is most frequently top-ranked (48.26%), followed by the two groups from ViT layer 24 (27.10% and 22.90%). Removing the group with 336 pixel windows at ViT layer 8 reduces correct-answer support by 3.824 millinats/token, while removing the group with 168 pixel windows at ViT layer 24 reduces it by 1.741. The positive effects occur for selected

Table 26: Full-set routing statistics and group-removal effects. Window sizes are in pixels; mean weights, medians, and Top-1 frequencies are percentages, and SD is in percentage points (pp).

Window	ViT layer	Mean (%)	SD (pp)	Median (%)	Top-1 (%)	$10^3 \Delta \log p$
336	8	47.50	7.39	49.96	48.26	$+3.824$
336	16	0.18	0.86	0.01	0.03	-0.191
336	24	23.77	4.31	25.05	22.90	-1.694
168	8	2.31	7.13	0.03	1.71	-0.547
168	16	0.19	0.32	0.05	0.00	$+0.006$
168	24	26.05	4.22	24.91	27.10	$+1.741$

groups, while the negative effect for the 336-pixel, ViT-layer-24 group shows that routing weight and marginal removal effect need not align. For the 336-pixel, ViT-layer-24 group, the full-minus-removal gap is -1.694 millinats/token despite a mean routing weight of 23.77%. Routing allocation and marginal removal effects therefore capture different aspects of feature use under renormalization. The group with 168 pixel windows at ViT layer 8 has a mean weight of 2.31% but a median of 0.03%, further showing variation in feature utilization. These summaries describe average log-probability effects, not accuracy changes.

D.6 Visual Dependence and Robustness Diagnostics

Paired interventions characterize a frozen checkpoint on fixed evaluation subsets. The full-model reference retains all learned gates and injection sites. The evaluation uses 120 images (20 each from AID, EuroSAT, METER-ML, NWPU-RESISC45, SIRI-WHU, and WHU-RS19). Secondary robustness and prompt tests use 60 images (10 per dataset). UCM and fMoW are excluded. Accuracy uses strict normalized exact match with equal weight per task. Confidence intervals (CIs) are computed using a task-stratified paired bootstrap with 10,000 replicates. Accuracy changes are measured in percentage points and computed before rounding. Leave-one-injection-out and secondary confidence intervals are pointwise and unadjusted for multiple comparisons. Table 27 reports image-replacement and leave-one-injection-out diagnostics on the 120-image set. No refresh disables injections while retaining detail stacks. Detail off removes the stacks entirely. Under learned-full inference, replacing each image with a preselected same-task image from a different class reduces the gold-answer mean token log probability, denoted by $\bar{\ell}(y^\star)$ in the table, where $y^\star$ is the reference answer, yielding a task-equal gap of 5.053 (95% CI: 4.517–5.626). The image-replacement gap is 4.944 (4.412–5.510) for both no-refresh and detail-off inference at the reported precision. All three conditions show sensitivity to image identity. Leave-one-injection-out changes are small. Only the unadjusted layer-6 interval excludes zero, with an estimated change of -0.0055. These comparisons diagnose the frozen checkpoint and are not retrained architectural ablations.

Table 27: Same-task image-replacement sensitivity and leave-one-injection-out diagnostics on 120 images. The visual gap is the correct-image score minus the swapped-image score. In panel (b), changes are measured as intervention minus full model.

(a) Correct-image versus swapped-image response

Inference condition	Correct $\bar{\ell}(y^\star)$	Swapped $\bar{\ell}(y^\star)$	Visual gap	95% CI
Full model	-0.565	-5.618	5.053	[4.517, 5.626]
No refresh	-0.612	-5.556	4.944	[4.412, 5.510]
Detail off	-0.612	-5.556	4.944	[4.412, 5.510]

(b) Leave one trained injection site out

Removed injection	$\Delta\bar{\ell}(y^\star)$ vs. full	95% CI
Layer 2	-0.0038	[-0.0353, 0.0258]
Layer 4	-0.0225	[-0.0498, 0.0014]
Layer 6	-0.0055	[-0.0095, -0.0018]
Layer 8	-0.0002	[-0.0039, 0.0032]

Input and prompt changes produce different responses on the 60-image subset. Table 28 presents secondary diagnostics on a fixed 60-image subset, comprising 10 images from each of six datasets.

Pre-downsampling is applied before standard preprocessing and does not change the encoder input size. Relative to the 65.00% clean macro accuracy, the geometric, JPEG-compression, and pre-downsampling variants have unchanged or higher observed accuracy. The second Gaussian-blur setting changes accuracy by -3.33 percentage points, with a confidence interval that includes zero. Consistent with the main-model configuration in Table 8, the input pathway uses a 336×336 global view together with a 672×672 detail canvas; these are standard components rather than an out-of-distribution input setting. Alternative aerial and land-use prompts yield changes of $+3.33$ and $+6.67$ points, respectively. Only the land-use prompt has a pointwise 95% interval that excludes zero ($[1.67, 13.33]$), these secondary intervals are not adjusted for multiple comparisons. The changes measure test-time sensitivity, not gains from training-time augmentation.

Table 28: Secondary robustness, input, and prompt diagnostics on 60 images. Each change is measured against the corresponding within-study reference before rounding. Intervals are pointwise 95% CIs without multiple-comparison adjustment.

Study	Variant	Accuracy (%)	Paired change (percentage points)	95% CI
Robustness	Clean	65.00	–	–
	Rotate 90°	70.00	+5.00	[0.00, 10.00]
	Rotate 180°	66.67	+1.67	[0.00, 5.00]
	Rotate 270°	66.67	+1.67	[-3.33, 6.67]
	Horizontal flip	66.67	+1.67	[0.00, 5.00]
	Vertical flip	70.00	+5.00	[0.00, 10.00]
	Pre-downsample to 224 pixels	66.67	+1.67	[0.00, 5.00]
	Pre-downsample to 168 pixels	66.67	+1.67	[0.00, 5.00]
	JPEG quality 70	65.00	0.00	[0.00, 0.00]
	JPEG quality 40	66.67	+1.67	[0.00, 5.00]
	Gaussian blur (setting 1)	65.00	0.00	[-5.00, 5.00]
	Gaussian blur (setting 2)	61.67	-3.33	[-10.00, 1.67]
Input pathway	336×336 global input (default)	65.00	–	–
	672×672 detail canvas (default)	65.00	0.00	[0.00, 0.00]
Prompt	Original wording	65.00	–	–
	Aerial wording	68.33	+3.33	[0.00, 8.33]
	Land-use wording	71.67	+6.67	[1.67, 13.33]

D.7 Efficiency and Scalability

Sliding windows add more relative runtime overhead than peak-memory overhead. To quantify the system-level cost of multi-scale processing, we compare the full model with the variant without sliding windows on A6000 GPUs. Table 29 reports SFT duration, per-image inference time, and peak memory in gibibytes (GiB), excluding model loading from inference. Sliding windows increase SFT time from 25 to 38 hours and inference time from 2.41 to 3.48 seconds per image, corresponding to increases of 52.0% and 44.4%. Peak memory rises from 16.11 to 17.50 GiB, an increase of approximately 8.6%. Runtime therefore grows more sharply than peak memory in this setup. These measurements quantify the overhead of sliding-window processing in this setup and show why deployment decisions must consider latency as well as memory capacity.

Table 29: Training time, per-image inference time, and peak memory on A6000 GPUs. Lower values indicate lower cost.

Setting	SFT time	Inference	Peak memory
Full model	38 h	3.48 s	17.50 GiB
without sliding windows	25 h	2.41 s	16.11 GiB

A100 latency across feature configurations. To examine inference cost on a second hardware platform, we compare the full model, the no-multi-scale variant, and the ViT-24-only variant on A100 GPUs. Table 30 reports average per-image latency across seven evaluation datasets, excluding one-time model loading. The unweighted means of the seven dataset-level latencies are 0.4595,

0.4391, and 0.2097 seconds for the full model, no-multi-scale variant, and ViT-24-only variant, respectively. The full model is therefore slower on average, although it is slightly faster than the no-multi-scale variant on AID. The ViT-24-only variant is faster on every listed dataset. These results show configuration-dependent inference overhead, with runtime differences that vary across the tested inputs. Comparisons remain within this A100 setup because hardware and latency summaries differ from the other reported measurements.

Table 30: Mean per-image inference time on A100 GPUs, in seconds, excluding one-time model loading. Mean is the unweighted average across the seven listed datasets. Lower values indicate lower cost.

Setting	AID	RS19	NWPU	SIRI	EuroSAT	UCM	fMoW	Mean
Full model	0.4299	0.4429	0.4874	0.4243	0.4171	0.4408	0.5743	0.4595
without multi-scale	0.4494	0.4170	0.4448	0.4174	0.4164	0.4272	0.5015	0.4391
ViT-24 only	0.1905	0.2004	0.2213	0.1848	0.1812	0.2041	0.2853	0.2097

E Qualitative Case Studies

Selected examples illustrate accurate answers to object-level questions. Figure 9 presents qualitative comparisons across three VQA subtasks, with models ordered from top to bottom: Phi-3.5-Vision (Research, 2024), MiniCPM-V-2.6 (Yao et al., 2024), GeoChat (Kuckreja et al., 2024), and MF-RSVLM. For the category question, MF-RSVLM describes the scene as "mountainous terrain," consistent with the reference label "Terrain." The comparison models describe a desert landscape or identify the windmill rather than the natural feature. For the existence question, MF-RSVLM answers yes and identifies two bridges, whereas all three comparison models report only one. For the counting question, MF-RSVLM identifies four vehicles, matching the reference, while the comparison models report three, an unspecified number, or five. These selected examples illustrate semantic agreement with the reference answers rather than exact label-string matching.

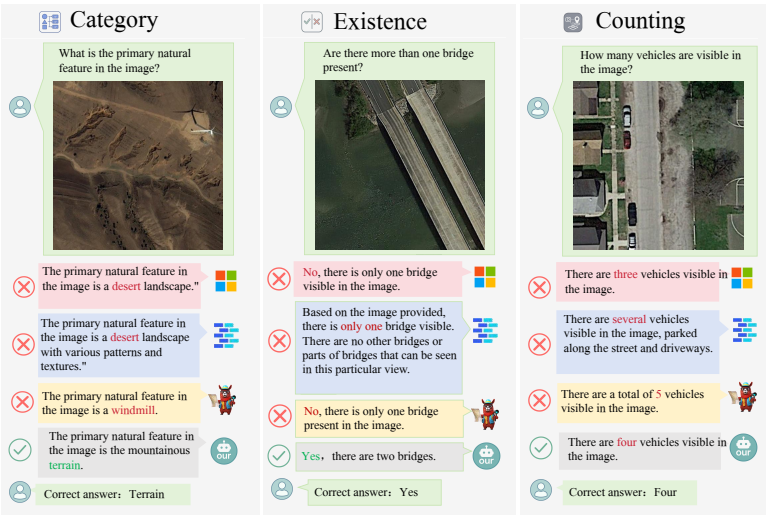

Figure 9: Qualitative comparison of Phi-3.5-Vision, MiniCPM-V-2.6, GeoChat, and MF-RSVLM on category, existence, and counting questions.

Multi-turn examples illustrate responses across object attributes, spatial relations, and scene properties. Figures 10–14 show questions about object attributes, spatial relations, counts, and scene properties across several scene types. Several responses agree semantically with the displayed references, including the overpass category and the absence of water in the highway examples. In the sports-field example, the model answers "Ground track field" for the facility on the right, whereas the reference is "Golf field." These examples provide a qualitative view of responses to multiple questions about the same image.

Category: Single-image Multi-turn VQA

Turn 1 (Object Category)
Human: What is the main structure crossing the highway?
Model (ours): overpass
Ground Truth: overpass

Turn 2 (Object Position)
Human: Which part of the image is the overpass located?
Model (ours): Bottom
Ground Truth: bottom

Turn 3 (Object Existence)
Human: Does the image contain any bodies of water?
Model (ours): No, there are no bodies of water visible in the image.
Ground Truth: No

Turn 4 (Image Property)
Human: Is the image in color or grayscale?
Model (ours): The image is in color.
Ground Truth: color

Turn 5 (Object Color)
Human: What color is predominant in the non-paved areas of the image?
Model (ours): Green
Ground Truth: green



Category: Single-image Multi-turn VQA

Turn 1 (Object Category)
Human: What type of structure is visible in the image?
Model (ours): overpass
Ground Truth: overpass

Turn 2 (Object Position)
Human: What is the position of the overpass in the image?
Model (ours): The overpass is located at the bottom of the image.
Ground Truth: bottom

Turn 3 (Object Existence)
Human: Does the image include any roads?
Model (ours): Yes, the image shows a section of a highway interchange.
Ground Truth: Yes

Turn 4 (Object Direction)
Human: Is the overpass running east-west or north-south?
Model (ours): The overpass is running east-west.
Ground Truth: east-west



Figure 10: Examples of single-image multi-turn VQA on highway scenes.

Category: Single-image Multi-turn VQA



Turn 1 (Image Property)
Human: Was the image provided by satellite imaging?
Model (ours): Yes
Ground Truth: Yes

Turn 2 (Object Existence)
Human: Does the image feature any buildings or trees?
Model (ours): No, the image does not show any buildings or trees.
Ground Truth: No

Turn 3 (Object Existence)
Human: Is there only one windmill present in the image?
Model (ours): Yes, there is only one windmill visible in the image.
Ground Truth: Yes

Turn 4 (Object Existence)
Human: Is the windmill surrounded by other structures?
Model (ours): No, the windmill is the only structure visible in the image.
Ground Truth: No

Category: Single-image Multi-turn VQA



Turn 1 (Object Quantity)
Human: How many vehicles are visible near the toll station?
Model (ours): There are two vehicles visible near the toll station.
Ground Truth: 2

Turn 2 (Object Position)
Human: Where is the toll station located within the image?
Model (ours): Bottom
Ground Truth: bottom

Turn 3 (Object Existence)
Human: Is there vegetation visible in the image?
Model (ours): Yes, there is vegetation visible in the image.
Ground Truth: Yes

Figure 11: Examples of single-image multi-turn VQA on a solitary windmill and a toll station.

Category: Single-image Multi-turn VQA



Turn 1 (Object Category)
Human: What is the main structure visible in the center of the image?
Model (ours): The main structure visible in the center of the image is a large train station.
Ground Truth: train station

Turn 2 (Object Category)
Human: What feature identifies the object as a train station?
Model (ours): The presence of multiple parallel railway tracks and a large building complex with a blue roof, which is likely the station building.
Ground Truth: tracks and platforms

Turn 3 (Object Quantity)
Human: How many train stations are visible in the image?
Model (ours): There is one train station visible in the image.
Ground Truth: 1

Turn 4 (Object Existence)
Human: Are there any other prominent structures besides the train station?
Model (ours): Yes, there are many residential buildings and factories in the image.
Ground Truth: Yes

Turn 5 (Object Size)
Human: Does the train station extend across the entire image?
Model (ours): No, the train station is not the entire image. It occupies a large area in the center of the image, but there are other areas visible around it.
Ground Truth: No

Category: Single-image Multi-turn VQA



Turn 1 (Object Category)
Human: What is the object located at the bottom of the image?
Model (ours): The object located at the bottom of the image is a large, white, circular storage tank.
Ground Truth: Storage tank

Turn 2 (Object Color)
Human: What is the dominant color of the natural landscape in the image?
Model (ours): The dominant color of the natural landscape in the image is green, indicating the presence of vegetation.
Ground Truth: Green

Turn 3 (Object Existence)
Human: Is there any body of water visible in the image?
Model (ours): No, there is no body of water visible in the image.
Ground Truth: No

Figure 12: Examples of single-image multi-turn VQA on complex infrastructure scenes.

Category: Single-image Multi-turn VQA

Turn 1 (Object Category)
Human: What is the object located at the bottom-left of the image?
Model (ours): The object located at the bottom-left of the image is a windmill.
Ground Truth: windmill

Turn 2 (Object Existence)
Human: Is there more than one windmill visible in the image?
Model (ours): No, there is only one windmill visible in the image.
Ground Truth: No

Turn 3 (Scene Type)
Human: What type of landscape is visible in the image?
Model (ours): The image shows a barren, rocky landscape.
Ground Truth: Barren

Turn 4 (Object Existence)
Human: Are there any roads visible in the image?
Model (ours): Yes, there is a road visible in the image.
Ground Truth: Yes

Turn 5 (Object Existence)
Human: Does the windmill cast a shadow?
Model (ours): Yes, the windmill casts a shadow on the ground.
Ground Truth: Yes



Category: Single-image Multi-turn VQA

Turn 1 (Object Category)
Human: What is the main recreational facility visible in the image?
Model (ours): baseball field
Ground Truth: baseball field

Turn 2 (Object Existence)
Human: Does the image show any buildings or houses?
Model (ours): No, the image does not show any buildings or houses.
Ground Truth: No

Turn 3 (Object Existence)
Human: Is there any body of water visible in the image?
Model (ours): No, there is no body of water visible in the image.
Ground Truth: No



Figure 13: Examples of single-image multi-turn VQA on natural and recreational scenes.

Category: Single-image Multi-turn VQA



Turn 1 (Object Color)
Human: What is the main color of the large open space in the image?
Model (ours): Green
Ground Truth: Green

Turn 2 (Object Existence)
Human: Is there a residential area visible in the image?
Model (ours): Yes, there is a residential area visible in the image.
Ground Truth: Yes

Turn 3 (Object Category)
Human: What sports facility is located on the left side of the image?
Model (ours): Ground track field
Ground Truth: Ground track field

Turn 4 (Object Category)
Human: What type of field occupies the right side of the image?
Model (ours): Ground track field
Ground Truth: Golf field

Turn 5 (Object Existence)
Human: Are there buildings present near the sports facilities?
Model (ours): Yes, there are several buildings near the sports facilities.
Ground Truth: Yes

Category: Single-image Multi-turn VQA



Turn 1 (Scene Type)
Human: What type of area is visible in the image?
Model (ours): Suburban
Ground Truth: suburban

Turn 2 (Object Quantity)
Human: How many vehicles are visible?
Model (ours): There are three vehicles visible in the image.
Ground Truth: 3

Turn 3 (Object Position)
Human: Where is the largest vehicle located?
Model (ours): Top-left
Ground Truth: top-left

Turn 4 (Object Existence)
Human: Is there a vehicle located in the center of the image?
Model (ours): Yes, there is a red vehicle located in the center of the image.
Ground Truth: Yes

Turn 5 (Object Existence)
Human: Does the image show any vehicles near the top edge?
Model (ours): Yes, there are two vehicles near the top edge of the image.
Ground Truth: Yes

Figure 14: Examples of single-image multi-turn VQA on sports and suburban scenes.