GEOCOCO: REPRESENTATION-DECOUPLED CO-OCCURRENCE COMPRESSION FOR MLLM GEOSPATIAL TOKENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long visual token sequences (e.g., 10^4–10^5) from ultra-high-resolution (UHR) remote sensing imagery increase the inference cost of multimodal large language models (MLLMs). Existing token compression methods typically depend on high-dimensional MLLM features, adding planning overhead and requiring feature computation before planning. Our observation study shows that low-dimensional image descriptors provide compression guidance with comparable accuracy and lower prefill time, while spatially regional grouping further improves accuracy. We therefore propose **GeoCoCo**, a feature-decoupled, plug-and-play framework for low-overhead, region-level visual token merging. Offline, GeoCoCo quantizes image descriptors into shared basic visual symbols and hierarchically merges them using spatial adjacency co-occurrence statistics aggregated across images to learn a shared compression prior. Online, this prior guides connected-region planning from input images and aggregation of the original model's visual token features without retraining the target MLLM. Planning is independent of MLLM features, enabling parallel CPU planning and GPU visual encoding. Experiments across XLRS-Bench-lite, LRS-VQA, and MME-RealWorld-RS and six MLLMs show that GeoCoCo reduces prefill time by up to 73.1% relative to uncompressed models, achieving the best performance–efficiency trade-off among the compared compression methods. We will release the relevant code for further research.

1 INTRODUCTION

Multimodal large language models (MLLMs) are increasingly used for ultra-high-resolution (UHR) remote sensing image interpretation (Muhtar et al., 2024; Zhang et al., 2024). However, patch-based visual encoding can produce 10^4–10^5 visual tokens per image, substantially increasing computation, memory consumption, and inference latency. Reducing visual sequence length while preserving global visual coverage and task-relevant information is therefore essential for efficient and scalable remote sensing MLLM inference (Wang et al., 2026a). Visual token compression offers a promising solution to these long visual sequences (Liu et al., 2026; Huang et al., 2024; Ye et al., 2025; Yang et al., 2025a; Endo et al., 2025; Shang et al., 2025; Xing et al., 2025; Arif et al., 2025; Yu et al., 2026; Hu et al., 2026; Huang et al., 2026). By selecting tokens or aggregating redundant representations, it shortens the sequence passed to the language model without requiring image regions to be discarded before encoding (Bolya et al., 2023; Li et al., 2025b; Dong et al., 2026; Zhao et al., 2025; Yang et al., 2025d; Li et al., 2026a; Lei et al., 2026a; Ma et al., 2026b; Wang et al., 2026c; 2025b). However, existing methods typically rely on high-dimensional MLLM features for similarity computation, matching, or clustering. These operations introduce additional planning overhead and require the corresponding MLLM features to be available before compression decisions can be made.

This limitation motivates a key question: **can compression plans be generated directly from image-space information and mapped to visual tokens without relying on MLLM features?** Such a design would allow planning to proceed without waiting for these features, enabling parallel execution with visual encoding. Our observation study on XLRS-Bench-lite shows that low-dimensional image descriptors achieve task accuracy close to that obtained with MLLM features, but with lower prefill time, suggesting their potential as a low-cost alternative for compression planning. Visualizations further show that image-guided token groups tend to be spatially localized,

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.

whereas those guided by MLLM features are more dispersed. Since UHR imagery often contains contiguous, visually homogeneous regions such as cropland and water bodies, we further introduce SLIC-based regional grouping, which improves accuracy under both guidance types. These results suggest that region-level compression, which aligns with the spatial distribution of redundancy, is better suited to UHR remote sensing tasks than spatially scattered token-level compression.

These observations support using low-dimensional image descriptors for regional compression, enabling planning to run independently of MLLM features and in parallel with visual encoding. However, SLIC still requires per-image iterative clustering, introducing additional online planning overhead. To implement image-guided planning efficiently, we learn basic visual symbols offline from a remote sensing image corpus and reuse them during inference.

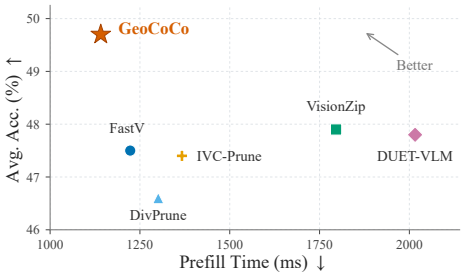


Figure 1: Accuracy–latency trade-off on XLRS-Bench-lite using Qwen2.5-VL-7B.

Building on this idea, we propose GeoCoCo, a feature-decoupled, plug-and-play framework for region-level visual token compression. GeoCoCo comprises one-time offline visual symbol learning and online regional compression. The offline stage discretizes low-dimensional image descriptors into basic visual symbols shared across images and aggregates their within-image spatial adjacency co-occurrence statistics across the corpus. Inspired by byte pair encoding (BPE) (Gage, 1994), it then hierarchically merges basic visual symbols into high-level visual symbols and constructs a symbol merging map as a shared compression prior for online region partitioning. During online inference, input patch descriptors are matched to basic visual symbols and mapped to high-level visual symbols. Spatially adjacent patches with the same high-level visual symbol are grouped into connected regions. The resulting region partition is aligned with the visual token grid to guide region-level aggregation of the original model's visual token features. Compared with existing methods for UHR visual token compression, GeoCoCo has two key characteristics: (1) **Low-cost compression planning** and (2) **Region-level structural modeling**. Extensive experiments demonstrate that GeoCoCo outperforms existing methods in the performance–efficiency trade-off, achieving the best results among the compared methods.

Our main contributions are summarized as follows:

- We show that low-dimensional image descriptors provide effective compression guidance for UHR remote sensing imagery with lower prefill time than high-dimensional MLLM features in our observation study, and region-level grouping further improves task accuracy.
- We propose GeoCoCo, which plans regional compression from image descriptors independently of MLLM features, allowing CPU planning to overlap with GPU visual encoding. It hierarchically merges basic visual symbols offline using spatial adjacency co-occurrence statistics across images to form a shared prior.
- Experiments on multiple UHR remote sensing benchmarks across diverse MLLM architectures and scales demonstrate competitive task accuracy and lower prefill time, yielding the best performance–efficiency trade-off among the compared token compression methods.

2 Observation Study

Computing relationships among MLLM features can introduce additional compression planning overhead for UHR remote sensing imagery. We therefore investigate whether image descriptors provide effective compression guidance with lower inference overhead. Motivated by grouping visualizations, we then examine whether spatially regional grouping further improves task accuracy.

Table 1: Comparison of guidance features in SparseVLM-based compression.

Guidance	Acc. (%)	Prefill (ms)
MLLM Features	**45.6**	4329
Image Descriptors	45.0	**2990**

2.1 Observation Setup

We evaluate Qwen2.5-VL-7B (Bai et al., 2025b) on XLRS-Bench-lite (Wang et al., 2025a), following the input settings and metric definitions in Sec. 4.1. To compare low-dimensional image descriptors with high-dimensional MLLM features as compression guidance, we first use SparseVLM (Zhang et al., 2025c) as a controlled framework, changing only the features used for clustering-based token recycling. All experiments use single-layer compression at the third Transformer block of the LLM with an 80% token compression ratio to make a fair comparison. The aggregated features are always the visual token hidden representations at this layer; image descriptors only guide grouping.

2.2 Image Features Provide Effective Token Compression Guidance

To investigate whether low-dimensional image descriptors provide effective compression guidance, we construct a controlled comparison using SparseVLM, which combines token pruning with clustering-based recycling. Given the broad spatial coverage of UHR images and the distribution of fine-grained information across local regions, we select the top 50% of tokens designated for pruning, ranked by importance, as recycling candidates. We compare high-dimensional MLLM features with 36-dimensional image descriptors constructed from the corresponding image patches, as defined in Sec. 3.1. Only the guidance features used for clustering are changed; both variants use the same token importance estimation, directly retained token set, recycling candidate set, cluster count, final token budget, and feature aggregation method.

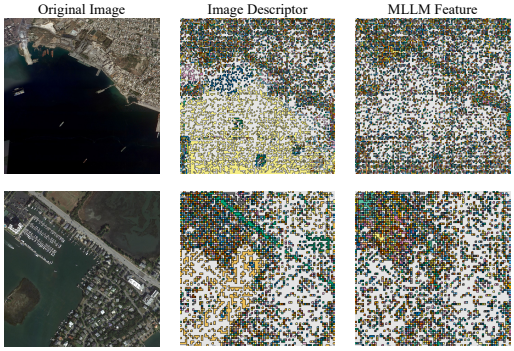


Figure 2: Token grouping guided by MLLM features and image descriptors on XLRS-Bench-lite.

Table 1 shows that image descriptors achieve 45.0% average accuracy versus 45.6% with MLLM features, while reducing prefill time from 4329 to $2990\,\mathrm{ms}$ (30.9%). The 0.6-percentage-point accuracy difference and lower prefill time support their potential as a low-cost alternative for clustering guidance in this controlled setting.

2.3 Spatially Regional Grouping Improves UHR Token Compression

As shown in Fig. 2, token groups guided by low-dimensional image descriptors are more spatially localized, whereas those guided by high-dimensional MLLM features are more dispersed, often containing spatially separated tokens within the same cluster. Since water bodies, cropland, and forests commonly form contiguous regions with relatively homogeneous appearances, we investigate whether explicitly incorporating spatial structure can better exploit local redundancy and improve compression performance.

We introduce SLIC (Achanta et al., 2012) regional grouping while maintaining the same compression location, guidance feature sources, and final token budget as in Sec. 2.2. Guidance features are arranged according to their 2D token positions, and SLIC combines feature and spatial distances to form local regions. Image-descriptor-guided planning depends only on the descriptors and positions, allowing it to run independently of high-dimensional MLLM feature computation. We execute this planning process in parallel with visual encoding, whereas planning guided by high-dimensional MLLM features must wait for the hidden representations at the compression location. As shown in Table 2, SLIC achieves average accuracies of 48.7% and 48.0% with high-dimensional MLLM features and image descriptors, respectively, improving over the corresponding SparseVLM settings by 3.1 and 3.0 percentage points. These gains across both guidance types support spatially regional compres-

Table 2: Comparison of guidance features in SLIC regional grouping.

Guidance	Acc. (%)	Prefill (ms)
MLLM Features	**48.7**(+3.1)	2021
Image Descriptors	48.0(+3.0)	**1615**

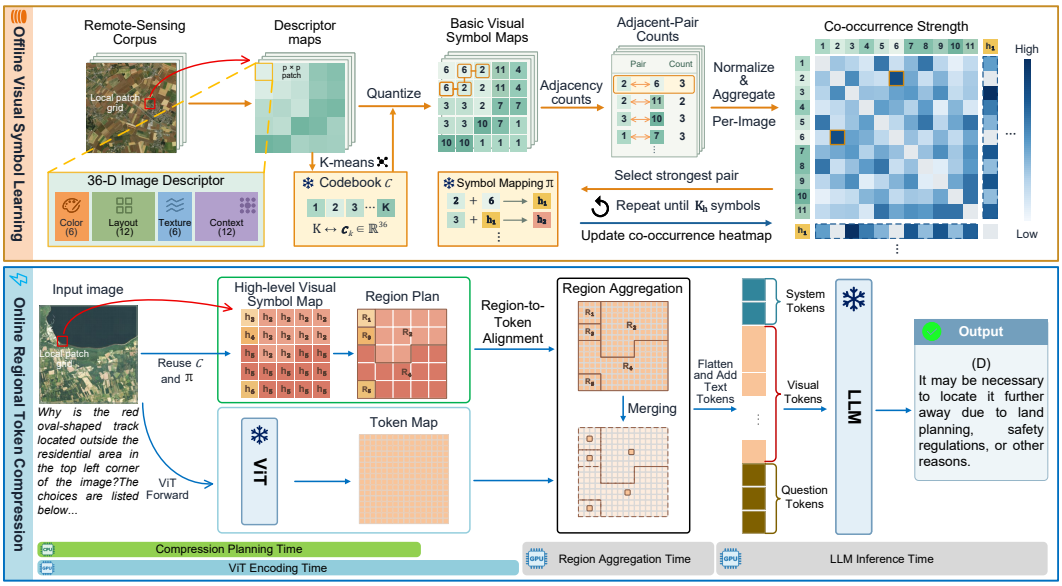


Figure 3: Overview of GeoCoCo. Offline learning constructs a codebook $\mathbf{C}$ and a symbol merging map π from image descriptors and cross-image spatial adjacency co-occurrence statistics. CPU-based regional planning reuses this prior in parallel with GPU visual encoding. The resulting regions are aligned with the visual token grid to guide feature aggregation.

sion for UHR imagery. Within the SLIC setting, image descriptors reduce prefill time from 2021 ms to 1615 ms (20.1%), with an accuracy difference of 0.7 percentage points.

3 METHOD

Based on our observations, we propose GeoCoCo, which learns a shared compression prior offline to guide regional aggregation of the original visual token features online. As illustrated in Figure 3, image-guided compression planning is independent of deep MLLM features and runs in parallel with visual encoding.

3.1 OFFLINE VISUAL SYMBOL LEARNING

Basic Visual Symbol Learning. To establish basic visual symbols shared across images, we divide each preprocessed image into N non-overlapping $p \times p$ patches $\{P_i\}_{i=1}^{N}$, retaining their 2D grid positions. Since mean color alone does not capture internal structure or surrounding appearance, we combine color distribution, intra-patch layout, texture, and neighborhood context:

$$\mathbf{x}_i = \left[\mathbf{x}_i^{\text{color}}; \mathbf{x}_i^{\text{layout}}; \mathbf{x}_i^{\text{texture}}; \mathbf{x}_i^{\text{context}}\right] \in \mathbb{R}^{36}. \tag{1}$$

All components are computed using pixel statistics and local differences without accessing MLLM features. Their detailed definitions are provided in Appendix B.1.

We standardize $\mathbf{x}_i$ dimension-wise using statistics from the offline corpus to obtain the processed descriptor $\tilde{\mathbf{x}}_i$. We then use K-means to learn a shared codebook $\mathbf{C} = \{\mathbf{c}_k\}_{k=1}^{K}$ consisting of K cluster centers. Each patch is assigned a basic visual symbol through nearest-center matching, $z_i = \arg\min_{1 \le k \le K} \|\tilde{\mathbf{x}}_i - \mathbf{c}_k\|_2^2$.

Spatial Co-occurrence Encoding. Local appearance variations may assign different basic visual symbols within a continuous visual structure, fragmenting the resulting regions. To alleviate this issue, we hierarchically merge basic visual symbols offline according to their co-occurrence statistics to construct high-level visual symbols and the map π, with the appearance distance constraint to reduce the risk of merging symbols associated with different objects only because they frequently occur next to each other.

Let $\mathcal{D}$ denote the co-occurrence learning image set, $z_i^{(I)}$ the basic visual symbol of patch i in image I, and (u_i, v_i) its row and column grid coordinates. We define four-neighbor edges and count unordered pairs of distinct symbols as

$$\mathcal{E}_I = \{(i,j) \mid i < j,\ |u_i - u_j| + |v_i - v_j| = 1\}, \tag{2}$$

$$n_I(a,b) = \sum_{(i,j)\in\mathcal{E}_I} \mathbf{1}\Big[\{z_i^{(I)}, z_j^{(I)}\} = \{a,b\}\Big], \qquad a < b, \tag{3}$$

where $i < j$ counts each edge once and $\mathbf{1}[\cdot]$ is the indicator function. To prevent images with more adjacent pairs of distinct symbols from dominating the statistics, we normalize the counts within each image and then sum them across images to obtain the initial co-occurrence strength:

$$S_0(a,b) = \sum_{I\in\mathcal{D}} \frac{n_I(a,b)}{\max\Big(1, \sum_{1\le r<s\le K} n_I(r,s)\Big)}, \tag{4}$$

where $S_0(b,a) = S_0(a,b)$.

We then iteratively merge symbols according to their co-occurrence strength, subject to an appearance distance constraint. Let $\mathcal{V}_t$ denote the set of symbols participating in merging at iteration t, and $\mathcal{B}_t(a)$ the basic symbols contained in $a \in \mathcal{V}_t$. Initially, $\mathcal{V}_0 = \{1, \dots, K\}$ and $\mathcal{B}_0(a) = \{a\}$. Using the frozen codebook, we define the maximum cross-symbol appearance distance and select the strongest eligible pair:

$$D_t(a,b) = \max_{k\in\mathcal{B}_t(a),\, l\in\mathcal{B}_t(b)} \|\mathbf{c}_k - \mathbf{c}_l\|_2^2, \tag{5}$$

$$(a^*, b^*) = \operatorname*{arg\,max}_{\substack{a,b\in\mathcal{V}_t,\ a<b \\ S_t(a,b)>0,\ D_t(a,b)\le\tau}} S_t(a,b), \tag{6}$$

where τ is the appearance distance threshold. We replace a^*, b^* with a high-level symbol h, set $\mathcal{B}_{t+1}(h) = \mathcal{B}_t(a^*) \cup \mathcal{B}_t(b^*)$, and update the co-occurrence strengths:

$$S_{t+1}(h,d) = S_t(a^*,d) + S_t(b^*,d), \qquad d \in \mathcal{V}_t \setminus \{a^*, b^*\}. \tag{7}$$

Other memberships and pairwise strengths remain unchanged. The merged pair's adjacency becomes internal and is excluded from subsequent selection; codebook centers are not updated.

Merging stops when the target number K_{h} is reached or no eligible pair remains. Let T denote the stopping iteration. The resulting number of high-level visual symbols is $|\mathcal{V}_T|$. We record the merging result as

$$\pi : \{1, \dots, K\} \to \{1, \dots, |\mathcal{V}_T|\}, \qquad \hat{z}_i = \pi(z_i). \tag{8}$$

3.2 Online Regional Token Compression

Regional Token Feature Aggregation. During online inference, GeoCoCo uses the frozen codebook $\mathbf{C}$ to map input patches to basic visual symbols and obtains high-level visual symbols through $\hat{z}_i = \pi(z_i)$. Since the same high-level visual symbol may occur at disconnected locations within an image, we form regions from four-neighbor edges connecting identical symbols:

$$\widehat{\mathcal{E}}_I = \{(i,j) \in \mathcal{E}_I \mid \hat{z}_i = \hat{z}_j\}, \qquad \{\mathcal{R}_r\}_{r=1}^{R} = \mathrm{CC}\Big(\{1, \dots, N\}, \widehat{\mathcal{E}}_I\Big). \tag{9}$$

Here, CC extracts connected components with patch indices $\{1, \dots, N\}$ as nodes and $\widehat{\mathcal{E}}_I$ as edges; R is their number, and $\mathcal{R}_r$ contains the patch indices of region r.

Since the image descriptor grid and the model's visual token grid may have different resolutions, we align region labels with the visual token grid using relative grid coordinates: majority voting over corresponding patch windows when the patch grid is finer, and nearest-neighbor label mapping otherwise. We then re-extract four-connected components within each mapped region and reindex them. Let N_{v} be the visual token count and ℓ_j the resulting component label of token j. The M_0 nonempty initial groups are

$$\mathcal{T}_m^{(0)} = \{j \in \{1, \dots, N_{\mathrm{v}}\} \mid \ell_j = m\}, \qquad m = 1, \dots, M_0. \tag{10}$$

These initial groups cover all input visual tokens and are mutually disjoint.

For a target retention ratio $\rho \in (0, 1]$, we set $M = \max\{1, \mathrm{round}(\rho N_{\mathrm{v}})\}$. If $M_0 > M$, we repeatedly merge the four-neighbor region pair with the lowest Ward cost until M regions remain (Ward Jr, 1963). This Ward cost uses processed low-dimensional image descriptors aligned with the token grid. If $M_0 < M$, we assign one output to each region and allocate the remaining budget according to region size, with exactly M outputs in total and no region receiving more outputs than its original token count. For a region assigned multiple outputs, to keep budget adjustment lightweight, we leave the region partition unchanged, replicate its mean feature, and select evenly spaced position anchors from the original sequence positions of its member tokens for subsequent position index construction. When $M_0 = M$, the initial groups remain unchanged. Details of the online compression procedure are provided in Appendix B.2.

Let $\mathcal{T}_m$ contain the original token indices contributing to output m. Outputs from the same region share $\mathcal{T}_m$ but use different anchors. Given the original model's visual token features $\mathbf{f}_j \in \mathbb{R}^{d_{\mathrm{v}}}$ before language-model processing, we compute

$$\bar{\mathbf{f}}_m = \frac{1}{|\mathcal{T}_m|} \sum_{j \in \mathcal{T}_m} \mathbf{f}_j, \qquad m = 1, \ldots, M. \tag{11}$$

For multidimensional position indices, single-output regions use the coordinate-wise mean of their members' original indices rounded to integers, whereas multiple-output regions use their selected anchors' indices. Subsequent positional offsets are adjusted accordingly. For one-dimensional indices, outputs are sorted by the mean original sequence index for single-output regions or the anchor index for multiple-output regions, then reindexed in compressed sequence order.

Parallel Execution. Compression planning depends only on image descriptors and spatial information, allowing a CPU worker to run concurrently with GPU visual processing up to the compression point without waiting for MLLM features. Once both finish, group metadata is transferred to the GPU for feature aggregation and position and sequence updates, followed by language-model prefilling. Overlapping these stages reduces the additional waiting introduced by planning without changing token grouping or aggregation.

4 EXPERIMENT

Table 3: Comparison of compression methods on XLRS-Bench-lite under matched token budgets.

Method	**Perception**								**Reasoning**					**Avg. Acc.**	**Avg. Tokens**	**Prefill Time**
Sub-tasks	OC	RC	OLUC	RLUC	OCC	OCL	OMS	OSR	AD	ECR	RP	RCCD	CCR	(%)		(ms)
Compression Method on InternVL3-8B																
+FastV *(ECCV'24)*	30.0	36.0	13.0	73.0	41.4	31.1	65.0	27.4	75.0	80.0	39.0	18.3	46.0	44.3	3.3k(↓80%)	454
+VisionZip *(CVPR'25)*	26.7	39.0	9.0	73.0	43.1	32.8	65.0	24.6	73.0	77.0	38.0	15.0	48.0	43.4	3.3k(↓80%)	852
+GeoCoCo	35.0	38.0	18.0	75.5	42.8	35.0	66.7	25.0	77.0	80.0	32.0	16.7	46.0	**45.2**	3.3k(↓80%)	**359**
Compression Method on Qwen2.5-VL-7B																
+FastV *(ECCV'24)*	33.3	43.0	30.0	80.5	41.1	39.6	66.7	35.0	69.0	72.0	26.0	36.7	44.0	47.5	3.3k(↓80%)	1223
+VisionZip *(CVPR'25)*	36.7	41.0	35.0	80.5	39.5	40.9	66.7	34.6	69.0	74.0	26.0	35.0	44.0	47.9	3.3k(↓80%)	1796
+DivPrune *(CVPR'25)*	36.7	38.0	32.0	78.5	38.9	39.4	66.7	34.8	67.0	73.0	25.0	35.0	41.0	46.6	3.3k(↓80%)	1301
+DUET-VLM *(CVPR'26)*	33.3	41.0	34.0	77.5	40.5	41.8	66.7	36.8	68.0	73.0	27.0	38.3	43.0	47.8	3.3k(↓80%)	2016
+IVC-Prune *(ICLR'26)*	28.3	41.0	25.0	77.5	37.6	38.4	65.0	32.0	71.0	76.0	31.0	51.7	41.0	47.4	3.3k(↓80%)	1367
+GeoCoCo	31.7	43.0	42.0	76.5	39.8	40.3	66.7	34.8	69.0	80.0	31.0	50.0	41.0	**49.7**	3.3k(↓80%)	**1141**
Compression Method on Qwen2.5-VL-72B																
+FastV *(ECCV'24)*	33.3	45.0	32.0	79.0	47.8	44.0	65.0	32.4	69.0	73.0	36.0	43.3	46.0	49.7	3.3k(↓80%)	2810
+VisionZip *(CVPR'25)*	33.3	44.0	41.0	79.5	46.5	42.9	65.0	34.8	70.0	75.0	35.0	41.7	43.0	50.1	3.3k(↓80%)	3096
+DivPrune *(CVPR'25)*	26.7	43.0	37.0	80.0	42.4	43.0	66.7	35.0	72.0	72.0	35.0	36.7	42.0	48.6	3.3k(↓80%)	2809
+DUET-VLM *(CVPR'26)*	33.3	48.0	39.0	79.5	47.0	43.8	65.0	33.6	68.0	74.0	37.0	41.7	40.0	50.0	3.3k(↓80%)	4011
+IVC-Prune *(ICLR'26)*	26.7	46.0	26.0	72.5	45.6	41.4	66.7	34.2	71.0	74.0	38.0	35.0	45.0	47.8	3.3k(↓80%)	3371
+GeoCoCo	35.0	49.0	35.0	76.5	43.8	43.4	66.7	32.6	73.0	75.0	35.0	41.7	46.0	**50.2**	3.3k(↓80%)	**2560**
Compression Method on GeoLLaVA-8K																
+FastV *(ECCV'24)*	23.3	41.0	46.0	50.5	43.8	31.6	65.0	37.4	53.0	60.0	65.0	46.7	49.0	47.1	13.8k(↓96%)	2294
+VisionZip *(CVPR'25)*	23.3	39.0	37.0	49.0	44.1	30.0	65.0	38.6	36.0	38.0	62.0	46.7	47.0	42.8	13.8k(↓96%)	4821
+GeoCoCo	26.7	40.0	48.0	66.5	43.0	32.9	65.0	34.6	63.0	78.0	64.0	50.0	50.0	**50.9**	13.8k(↓96%)	**1722**

4.1 EXPERIMENTAL SETUP

Evaluation Protocol. We evaluate GeoCoCo on XLRS-Bench-lite (Wang et al., 2025a), LRS-VQA (Luo et al., 2025), and MME-RealWorld-RS (Zhang et al., 2025b).We use Qwen2.5-VL-7B (Bai et al., 2025b) as the primary backbone and assess generality across architectures and

scales using Qwen2.5-VL-72B, Qwen3-VL-8B (Bai et al., 2025a), InternVL3-8B (Zhu et al., 2025), InternVL3.5-8B (Wang et al., 2025c), and the remote sensing model GeoLLaVA-8K (Wang et al., 2026a). Baselines include FastV (Chen et al., 2024), VisionZip (Yang et al., 2025c), DivPrune (Alvar et al., 2025), DUET-VLM (Singh et al., 2026), and IVC-Prune (Sun et al., 2026b). We report macro-average accuracy over the 13 XLRS-Bench-lite subtasks and over question categories within each LRS-VQA subset, and overall sample accuracy on MME-RealWorld-RS. Efficiency metrics include average retained visual token count, token compression ratio, Prefill Time, and E2E Latency. Prefill Time spans visual encoding after input preprocessing through completion of the first LLM forward pass; E2E Latency spans sample reading through complete answer text generation. Both are mean wall-clock times in ms/sample.

Implementation Details. Offline learning uses images from the remote sensing training corpus released with GeoLLaVA-8K (Wang et al., 2026a), without using their question–answer annotations. We set $p = 14$, $K = 48$, and a target of $K_{\mathrm{h}} = 32$. We set τ to the 30th percentile of squared codebook-center distances for distinct basic visual symbol pairs observed as neighbors. We evaluate using LMMs-Eval (Zhang et al., 2025a). Unless otherwise stated, we set `max_pixels` to 12,845,056 for Qwen models and `max_patches` to 64 for InternVL models. For GeoLLaVA-8K, we replace its native compression module. Detailed information are provided in Appendix A and B.4.

Table 4: Comparison at 50% compression rate XLRS-Bench-lite using Qwen2.5-VL-7B.

Method	Acc. (%)	Prefill (ms)
Uncompressed	47.4	1924
FastV	47.0	1441
VisionZip	47.2	1543
DivPrune	46.4	1966
DUET-VLM	47.5	2134
IVC-Prune	47.2	1783
GeoCoCo	**49.5**	**1399**

4.2 Main Results

Table 5: Comparison with representative MLLMs and plug-in generality of GeoCoCo across backbones ranging from 7B to 72B parameters.

Method	**Perception**								**Reasoning**					**Avg. Acc**	**Avg. Tokens**	**Prefill Time**
Sub-tasks	OC	RC	OLUC	RLUC	OCC	OCL	OMS	OSR	AD	ECR	RP	RCCD	CCR	(%)		(ms)
Remote Sensing MLLMs																
GeoChat	16.7	29.0	2.0	23.0	21.1	16.8	35.0	24.2	33.0	43.0	10.0	–	21.0	22.9	–	–
ZoomEarth	–	–	–	–	–	–	–	–	–	–	–	–	–	40.2	–	–
GeoEyes	38.3	40.0	24.0	73.5	59.5	66.1	68.3	32.2	77.0	80.0	56.0	40.0	50.0	**54.2**	–	–
UHR-BAT	21.7	33.0	50.0	55.5	43.5	33.8	65.0	44.8	62.0	71.0	54.0	46.7	51.0	48.6	–	–
GeoLLaVA-8K	26.7	38.0	49.0	69.0	41.6	31.6	65.0	35.0	67.0	78.0	66.0	50.0	52.0	51.5	13.8k (↓96%)	1770
+GeoCoCo	26.7	40.0	48.0	66.5	43.0	32.9	65.0	34.6	63.0	78.0	64.0	50.0	50.0	50.9	13.8k (↓96%)	1722
Open-source MLLMs																
InternVL3-8B	33.3	40.0	9.0	77.0	39.1	32.8	65.0	26.4	74.0	79.0	42.0	25.0	55.0	46.0	16.1k	1201
+GeoCoCo	35.0	38.0	18.0	75.5	42.8	35.0	66.7	25.0	77.0	80.0	32.0	16.7	46.0	45.2	3.3k (↓80%)	**359**
InternVL3.5-8B	36.7	44.0	46.0	71.0	35.0	25.6	66.7	37.6	73.0	76.0	49.0	46.7	50.0	50.6	16.1k	1336
+GeoCoCo	33.3	41.0	47.0	76.0	34.4	24.5	66.7	37.8	73.0	78.0	51.0	38.3	47.0	49.9	3.3k (↓80%)	389
Qwen2.5-VL-7B	33.3	40.0	31.0	77.0	40.6	40.5	66.7	36.2	68.0	72.0	27.0	38.3	45.0	47.4	16.1k	1924
+GeoCoCo	31.7	43.0	42.0	76.5	39.8	40.3	66.7	34.8	69.0	80.0	31.0	50.0	41.0	49.7	3.3k (↓80%)	1141
Qwen3-VL-8B	21.7	50.0	26.0	81.5	46.6	43.1	66.7	30.4	74.0	79.0	37.0	43.3	51.0	50.0	12.6k	2348
+GeoCoCo	21.7	42.0	27.0	82.5	44.3	42.8	66.7	29.2	71.0	80.0	41.0	43.3	52.0	49.5	2.5k (↓80%)	1606
Qwen2.5-VL-72B	33.3	47.0	39.0	80.0	45.3	42.1	65.0	34.0	71.0	74.0	37.0	43.3	42.0	50.2	16.1k	9523
+GeoCoCo	35.0	49.0	35.0	76.5	43.8	43.4	66.7	32.6	73.0	75.0	35.0	41.7	46.0	50.2	3.3k (↓80%)	2560

Comparison with Existing Methods. At matched per-backbone token budgets, GeoCoCo achieves the highest average accuracy and lowest prefill time among the compared methods on all four backbones (Table 3). It exceeds the strongest baseline by 0.9 and 3.8 percentage points on InternVL3-8B and GeoLLaVA-8K, respectively; on Qwen2.5-VL-72B, it is 0.1 points more accurate and $536\,\mathrm{ms}$ faster than VisionZip. At 50% token compression on Qwen2.5-VL-7B (Table 4), GeoCoCo leads in both accuracy (49.5%, 2.0 points above DUET-VLM) and prefill time ($1399\,\mathrm{ms}$, 27.3% below the uncompressed model).

Plug-in Generality across MLLMs. Table 5 evaluates GeoCoCo's plug-in generality, with remote sensing MLLMs included for reference. Without updating MLLM parameters, 80% token compression on Qwen and InternVL changes average accuracy by -0.8 to $+2.3$ percentage points while reducing prefill time by 31.6%–73.1%. Qwen2.5-VL-72B maintains 50.2% accuracy while reducing prefill time from 9523 to $2560\,\mathrm{ms}$. For GeoLLaVA-8K, replacing native compression at a matched token budget yields 50.9% accuracy, compared with the original model's 51.5%.

Table 6: Comparison of GeoCoCo and existing methods on LRS-VQA and MME-RealWorld-RS.

Method / Dataset	MME-RW MME-RW-RS	LRS-VQA LRS-FAIR	LRS-VQA LRS-Bridge	LRS-VQA LRS-STAR	Avg. Acc (%)	Avg. Tokens	Prefill Time (ms)
Qwen2.5-VL-7B	43.7	19.7	36.3	26.4	**31.5**	15.3k	1895
+FastV	37.5	18.6	34.9	26.1	29.3	3.1k (↓80%)	1193
+VisionZip	41.7	19.2	35.4	26.3	30.6	3.1k (↓80%)	1538
+GeoCoCo	39.6	21.1	33.1	26.3	30.0	3.1k (↓80%)	**1127**
InternVL3-8B	50.0	22.5	37.4	26.9	34.2	13.4k	1067
+FastV	49.4	22.1	37.1	26.9	33.9	2.7k (↓80%)	426
+VisionZip	42.0	21.8	35.7	27.4	31.7	2.7k (↓80%)	802
+GeoCoCo	50.9	22.9	39.0	26.8	**34.9**	2.7k (↓80%)	**328**
InternVL3.5-8B	48.2	25.4	35.5	27.6	**34.2**	13.9k	1259
+FastV	45.6	25.3	33.9	27.7	33.1	2.8k (↓80%)	449
+VisionZip	40.0	24.8	33.5	26.8	31.3	2.8k (↓80%)	827
+GeoCoCo	48.3	25.6	35.5	27.0	34.1	2.8k (↓80%)	**351**

Generalization across Benchmarks. Table 6 compares methods on MME-RealWorld-RS and LRS-VQA at an 80% token compression ratio. Across the three backbones, GeoCoCo improves average accuracy over FastV by 0.7–1.0 percentage points and achieves the lowest prefill time. It also outperforms VisionZip in average accuracy on both InternVL models. On Qwen2.5-VL-7B, its average accuracy is 0.6 percentage points below VisionZip, but its prefill time is 26.7% lower. These results extend GeoCoCo's favorable performance–efficiency trade-off beyond XLRS-Bench-lite.

Table 7: Acc. and latency on XLRS-Bench-lite at 80% token compression using Qwen2.5-VL-7B.

Metrics	FastV	VisionZip	DivPrune	DUET-VLM	IVC-Prune	GeoCoCo
Prefill Time (ms)	1223	1796	1301	2016	1367	**1141**
E2E Latency (ms)	3548	4557	4391	5179	4580	**3294**
Avg. Acc. (%)	47.5	47.9	46.6	47.8	47.4	**49.7**

Efficiency Analysis. As shown in Table 7, GeoCoCo achieves the lowest prefill time ($1141\,\mathrm{ms}$) and end-to-end latency ($3294\,\mathrm{ms}$) among the compared compression methods. Relative to FastV, the fastest baseline, it reduces these latencies by 6.7% and 7.2%, respectively, while improving average accuracy from 47.5% to 49.7%.

4.3 Ablation Studies

Component ablation. Table 8a shows that using the full descriptor instead of color alone for SLIC raises accuracy from 47.9% to 48.5%, but increases prefill time from 1137 to $1503\,\mathrm{ms}$. Replacing SLIC with basic visual symbol grouping yields 49.0% at $1124\,\mathrm{ms}$; co-occurrence-guided grouping reaches 49.7% at $1141\,\mathrm{ms}$. Relative to full-descriptor SLIC, GeoCoCo gains 1.2 accuracy points and cuts prefill time by 24.1%.

Table 8: Ablations on XLRS-Bench-lite with Qwen2.5-VL-7B at 80% token compression. Component rows 1–4 use SLIC (Sec. 2.3). T_{plan}, T_{vis}, and T_{apply} denote planning, visual encoding and connector execution, and plan application. Further ablations appear in Appendix C.4.

(a) Component ablation.

#	Color	Layout	Texture	Context	Basic Symbol	High-level Symbol	Avg. Acc.	Prefill Time
1	✓	–	–	–	–	–	47.9	1137
2	✓	✓	–	–	–	–	48.4	1313
3	✓	✓	✓	–	–	–	48.5	1428
4	✓	✓	✓	✓	–	–	48.5	1503
5	✓	✓	✓	✓	✓	–	49.0	**1124**
6	✓	✓	✓	✓	✓	✓	**49.7**	1141

(b) Parallel execution.

Metric	Serial	Parallel
T_{plan}	610	650
T_{vis}	942	960
T_{apply}	14	13
Prefill Time	1724	**1141**
E2E Latency	3964	**3294**

Effect of parallel execution. Table 8b compares the execution times of serial and parallel CPU–GPU execution under the same compression configuration. Parallel CPU–GPU execution reduces prefill time from 1724 to $1141\,\mathrm{ms}$ (33.8%) and end-to-end latency from 3964 to $3294\,\mathrm{ms}$ (16.9%). Decoupling compression planning from deep visual features allows most CPU planning time to overlap with GPU visual encoding, reducing additional waiting on the inference path.

4.4 Qualitative Analysis

Figure 4 compares the basic and high-level visual symbol maps. Some neighboring farmland patches with similar appearances initially receive different basic visual symbols, resulting in fragmented labels. Spatial co-occurrence encoding maps these symbols to shared high-level visual symbols, forming larger connected regions with consistent labels, while some visually distinct areas retain different symbols.

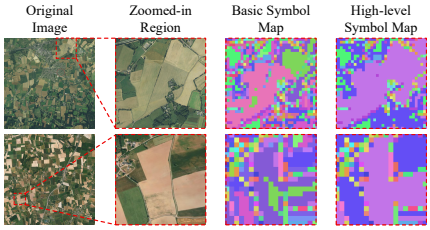


Figure 4: Visualization of spatial co-occurrence encoding in GeoCoCo.

5 Related Works

As MLLMs are applied to UHR remote sensing imagery (Kuckreja et al., 2024; Muhtar et al., 2024; Zhang et al., 2024; Zhan et al., 2025; Pang et al., 2025; Soni et al., 2025; Li et al., 2026b; 2025c), a single image can produce tens of thousands of visual tokens, increasing computational cost and GPU memory consumption. Existing approaches address these costs through region-selection-based zoom-in search or visual token compression.

Zoom-in methods inspect task-relevant regions through high-resolution cropping or iterative visual search (Wu & Xie, 2024; Li et al., 2025a; Su et al., 2026; Li et al., 2026c; Zhang et al., 2026; Zheng et al., 2026; Qi et al., 2025). Subsequently, Khayatkhoei et al. (2025) uses internal relevance estimates to automatically crop question-relevant regions. In UHR remote sensing scenarios, Liu et al. (2025) uses active visual perception to progressively locate task-relevant regions in large-scale geospatial imagery. Wang et al. (2026b) further learns task-oriented visual search policies, allowing models to actively select and zoom into local evidence regions based on the question.

Visual token compression reduces the inference overhead of MLLMs by shortening visual sequences (Li et al., 2026d; Lei et al., 2026b; Gao et al., 2026; Sun et al., 2026a; Xu et al., 2026; Yang et al., 2025b; Han et al., 2026; Song et al., 2026; Li et al., 2026e; Hyun et al., 2025; Zhu et al., 2026). Chen et al. (2024) uses attention from shallow language model layers to identify and prune redundant visual tokens. Yang et al. (2025c) retains informative visual tokens before they enter the language model and constructs complementary representations by aggregating redundant tokens. Alvar et al. (2025) selects representative tokens based on visual representation diversity. Singh et al. (2026) adopts a two-stage compression strategy that combines visual-side redundant token merging with text-guided layer-wise pruning within the language model. Sun et al. (2026b) retains both implicit visual coordinate tokens and semantically relevant foreground tokens to preserve spatial information and semantic content. In UHR remote sensing scenarios, extremely long visual sequences further amplify visual token redundancy and compression decision overhead (Luo et al., 2025). Dang et al. (2026) use query-guided multiscale selection to retain important tokens and pool redundant ones within regions. Ma et al. (2026a) combine task-adaptive resolution routing with token selection based on task relevance, wavelet-derived spatial structure, and local feature redundancy. In contrast, GeoCoCo constructs regions from image descriptors using a shared co-occurrence prior and aggregates all tokens within each resulting region, rather than selecting individual tokens. Planning is independent of MLLM features and runs in parallel with visual encoding.

6 Conclusion

In this work, we demonstrate the potential of low-dimensional image features for low-cost compression guidance and introduce GeoCoCo, an image-guided, plug-and-play framework for region-level visual token compression in UHR remote sensing imagery. GeoCoCo learns a shared compression prior by hierarchically merging basic visual symbols using spatial adjacency co-occurrence statistics aggregated across images. This prior guides connected-region aggregation of the original visual tokens without retraining the target MLLM. Decoupling compression planning from MLLM features allows CPU planning to overlap with GPU visual encoding. Experiments across remote sensing benchmarks and diverse MLLM architectures and scales demonstrate competitive accuracy with lower prefill time, offering the best performance–efficiency trade-off among the compared visual token compression methods.

AI USE STATEMENT

OpenAI Codex assisted with code implementation, manuscript polishing, and LaTeX formatting. The authors fully reviewed the AI-assisted content and code, checked references and analyses against the original sources and experimental records, and accept responsibility for the final manuscript, implementation, and reported results.

REPRODUCIBILITY STATEMENT

We describe GeoCoCo in Sec. 3 and provide detailed descriptor definitions and offline and online algorithms in Appendices B.1 and B.2. Dataset descriptions and evaluation protocols are provided in Appendix A. Model-specific integration and experimental configurations are in Appendices B.3 and B.4, respectively. We will release our code for further research.

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1773–1781, 2025.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *Proceedings of the International Conference on Learning Representations*, 2023.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Proceedings of the European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Yunkai Dang, Minxin Dai, Yuekun Yang, Zhangnan Li, Wenbin Li, Feng Miao, and Yang Gao. Uhr-bat: Budget-aware token compression vision-language model for ultra-high-resolution remote sensing. In *Forty-third International Conference on Machine Learning*, 2026.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. Mmtok: Multimodal coverage maximization for efficient inference of vlms. In *International Conference on Learning Representations*, volume 2026, pp. 48075–48099, 2026.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22826–22835. IEEE, 2025.

Philip Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, 1994.

Tianxiao Gao, Shanwei Zhao, Shuo Fang, Shiai Zhu, and Chenguang Ma. Quietprune: Query-guided early token pruning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3553–3562, 2026.

Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Junjie Chen, Honggang Chen, Donglin Wang, Qingsen Yan, and Siteng Huang. Filter, correlate, compress: Training-free token reduction for mllm acceleration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4601–4609, 2026.

Lianyu Hu, Liqing Gao, Fanhua Shang, Liang Wan, and Wei Feng. illava: An image is worth fewer than 1/3 input tokens in large multimodal models. In *International Conference on Learning Representations*, volume 2026, pp. 73724–73737, 2026.

Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European conference on computer vision*, pp. 214–230. Springer, 2024.

Mouxiao Huang, Borui Jiang, Dehua Zheng, Hailin Hu, Kai Han, and Xinghao Chen. Ppe: Positional preservation embedding for token compression in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 67656–67676, 2026.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video llms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23990–24000. IEEE, 2025.

Mahyar Khayatkhoei, Prateek Chhikara, Filip Ilievski, et al. Mllms know where to look: Training-free perception of small visual details with multimodal llms. In *International Conference on Learning Representations*, volume 2025, pp. 68194–68213, 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 27831–27840, 2024.

Jingyu Lei, Gaoang Wang, and Der-Horng Lee. Core: Compact object-centric representations as a new paradigm for token merging in lvlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39593–39605, 2026a.

Lei Lei, Jie Gu, Xiaokang Ma, Chu Tang, Jingmin Chen, and Tong Xu. Task-related token compression in multimodal large language models from an explainability perspective. In *International Conference on Learning Representations*, volume 2026, pp. 112724–112746, 2026b.

Ao Li, Yuxiang Duan, Jinghui Zhang, Congbo Ma, Yutong Xie, Gustavo Carneiro, Mohammad Yaqub, and Hu Wang. Transprune: Token transition pruning for efficient large vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39529–39538, 2026a.

Daixun Li, Zirui Li, Sibo He, Jiayun Tian, Mingxiang Cao, Weiying Xie, Yunke Wang, Xin Zhang, Yusi Zhang, Yunsong Li, Chang Xu, and Leyuan Fang. Geocot: Towards reliable remote sensing reasoning with manifold perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20575–20585, June 2026b.

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9098–9108. IEEE, 2025a.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133(10):6794–6812, 2025b.

Yangfu Li, Hongjian Zhan, Jiawei Chen, Yuning Gong, Qi Liu, and Yue Lu. Deepscan: A training-free framework for visually grounded reasoning in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19155–19164, 2026c.

Yangfu Li, Hongjian Zhan, Tianyi Chen, Qi Liu, Yu-jie Xiong, and Yue Lu. Why 1+ 1 $<$ 1 in visual token pruning: Beyond naive integration via multi-objective balanced covering. *Advances in Neural Information Processing Systems*, 38:71999–72039, 2026d.

Yueying Li, Fengxiang Wang, Yan Li, Mingshuo Chen, Mengying Zhao, and Long Lan. Semantic-geometric dual compression: Training-free visual token reduction for ultra-high-resolution remote sensing understanding. *arXiv preprint arXiv:2604.11122*, 2026e.

Zhenshi Li, Dilxat Muhtar, Feng Gu, Yanglangxing He, Xueliang Zhang, Pengfeng Xiao, Guangjun He, and Xiaoxiang Zhu. Lhrs-bot-nova: Improved multimodal large language model for remote sensing vision-language interpretation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 227:539–550, 2025c.

Ruixun Liu, Bowen Fu, Jiayi Song, Kaiyu Li, Wanchen Li, Lanxuan Xue, Hui Qiao, Weizhan Zhang, Deyu Meng, and Xiangyong Cao. Zoomearth: Active perception for ultra-high-resolution geospatial vision-language tasks. *arXiv preprint arXiv:2511.12267*, 2025.

Xuyang Liu, Ziming Wang, Junjie Chen, Yuhang Han, Yingyao Wang, Jiale Yuan, Jun Song, Siteng Huang, and Honggang Chen. Global compression commander: Plug-and-play inference acceleration for high-resolution large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 7350–7358, 2026.

Junwei Luo, Yingying Zhang, Xue Yang, Kang Wu, Qi Zhu, Lei Liang, Jingdong Chen, and Yansheng Li. When large vision-language model meets large remote sensing imagery: Coarse-to-fine text-guided token pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9206–9217, 2025.

Kexin Ma, Jing Xiao, Bowen Xing, Liang Liao, and Chia-Wen Lin. Sa-gem: Scale-adaptive and geospatial evidence-modulated token pruning for efficient remote sensing large vision-language models. *arXiv preprint arXiv:2608.15075*, 2026a.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Zhen Song, Jie Chen, and Hairong Zheng. Apet: Approximation-error guided token compression for efficient vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26306–26316, 2026b.

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*, pp. 440–457. Springer, 2024.

Chao Pang, Xingxing Weng, Jiang Wu, Jiayu Li, Yi Liu, Jiaxing Sun, Weijia Li, Shuai Wang, Litong Feng, Gui-Song Xia, et al. Vhm: Versatile and honest vision language model for remote sensing image analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6381–6388, 2025.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Xu Bin, Lei Hou, Juanzi Li, Yuxiao Dong, et al. Cogcom: A visual language model with chain-of-manipulations reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 11090–11110, 2025.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. Duet-vlm: Dual stage unified efficient token reduction for vlm training and inference. *arXiv preprint arXiv:2602.18846*, 2026.

Jiafei Song, Fengwei Zhou, Jin Qu, Wenjin Jason Li, Tong Wu, Gengjian Xue, Zhikang Zhao, Daomin Wei, Yichao Lu, and Bailin Na. Evocomp: Learning visual token compression for multimodal large language models via semantic-guided evolutionary labeling. *arXiv preprint arXiv:2604.17087*, 2026.

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, et al. Earthdial: Turning multi-sensory earth observations to interactive dialogues. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14303–14313. IEEE, 2025.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. *Advances in Neural Information Processing Systems*, 38:8222–8251, 2026.

Guohao Sun, Yufei Wang, Sizhuo Ma, Yuege Xie, Yuting Cheng, Zhiqiang Tao, and Jian Wang. If-prune: Information-flow guided token pruning for efficient vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3522–3531, 2026a.

Zhichao Sun, Yidong Ma, Gang Liu, Yibo Chen, Xu Tang, Yao Hu, and Yongchao Xu. Ivc-prune: Revealing the implicit visual coordinates in lvlms for vision token pruning. *arXiv preprint arXiv:2602.03060*, 2026b.

Fengxiang Wang, Hongzhen Wang, Zonghao Guo, Di Wang, Yulin Wang, Mingshuo Chen, Qiang Ma, Long Lan, Wenjing Yang, Jing Zhang, et al. Xlrs-bench: Could your multimodal llms understand extremely large ultra-high-resolution remote sensing imagery? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14325–14336, 2025a.

Fengxiang Wang, Mingshuo Chen, Yueying Li, Di Wang, Haotian Wang, Zonghao Guo, Zefan Wang, Shan Boqi, Long Lan, Yulin Wang, et al. Geollava-8k: scaling remote-sensing multimodal large language models to 8k resolution. *Advances in Neural Information Processing Systems*, 38: 159185–159218, 2026a.

Fengxiang Wang, Mingshuo Chen, Yueying Li, Yajie Yang, Yifan Zhang, Long Lan, Xue Yang, Hongda Sun, Yulin Wang, Di Wang, et al. Geoeyes: On-demand visual focusing for evidence-grounded understanding of ultra-high-resolution remote sensing imagery. *arXiv preprint arXiv:2602.14201*, 2026b.

Haicheng Wang, Zhemeng Yu, Gabriele Spadaro, Chen Ju, Victor Quétu, Shuai Xiao, and Enzo Tartaglione. Folder: Accelerating multi-modal large language models with enhanced performance. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23614–23625. IEEE, 2025b.

Huanyu Wang, Jushi Kai[1], Haoli Bai, Lu Hou, Bo Jiang, Ziwei He, and Zhouhan Lin. Fourier compressor: Frequency-domain visual token compression for vision-language models. In *Computer Vision–ECCV 2026: 19th European Conference, Malmö, Sweden, September 8–12, 2026, Proceedings, Part IX*, pp. 284. Springer Nature, 2026c.

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*, 2025c.

Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Tong Xu, Hailong Shi, and Xingyu Gao. SCoRe: Salience-coverage reduction for vision token pruning in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24686–24695, 2026.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19803–19813, 2025a.

Longrong Yang, Dong Shen, Chaoxiang Cai, Kaibing Chen, Fan Yang, Tingting Gao, Di Zhang, and Xi Li. Libra-merging: Importance-redundancy and pruning-merging trade-off for acceleration plug-in in large vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9402–9412, 2025b.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025c.

Sihan Yang, Runsen Xu, Chenhang Cui, Tai Wang, Dahua Lin, and Jiangmiao Pang. Vflowopt: A token pruning framework for lmms with visual information flow-guided optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23924–23934. IEEE, 2025d.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24972–24982, 2025.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. Visiontrim: Unified vision token compression for training-free mllm acceleration. *arXiv preprint arXiv:2601.22674*, 2026.

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.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, 2025a.

Peirong Zhang, Yidan Zhang, Luxiao Xu, Jinliang Lin, Zonghao Guo, Fengxiang Wang, Xue Yang, Kaiwen Wei, and Lei Wang. Geovis: Geospatially rewarded visual search for remote sensing visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14335–14345, 2026.

Wei Zhang, Miaoxin Cai, Tong Zhang, Yin Zhuang, and Xuerui Mao. Earthgpt: A universal multimodal large language model for multisensor image comprehension in remote sensing domain. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–20, 2024.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, volume 2025, pp. 89655–89701, 2025b.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the Forty-second International Conference on Machine Learning*, 2025c.

Shiyu Zhao, Zhenting Wang, Felix Juefei-Xu, Xide Xia, Miao Liu, Xiaofang Wang, Mingfu Liang, Ning Zhang, Dimitris N Metaxas, and Licheng Yu. Accelerating multimodal large language models by searching optimal vision token reduction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29869–29879. IEEE, 2025.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, and Chao Shen. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Qihui Zhu, Tao Zhang, Yuchen Wang, Zijian Wen, Mengjie Zhang, Shuangwu Chen, Xiaobin Tan, Jian Yang, Yang Liu, Zhenhua Dong, et al. Hawk: Head importance-aware visual token pruning in multimodal models. *arXiv preprint arXiv:2604.07812*, 2026.

A Datasets and Evaluation Protocols

A.1 XLRS-Bench-lite

XLRS-Bench (Wang et al., 2025a) is designed for UHR remote sensing image understanding, covering visual question answering, image captioning, and visual grounding. We use the officially released XLRS-Bench-lite, which contains $3{,}080$ question–answer samples across 13 L3 subtasks, comprising 8 perception subtasks and 5 reasoning subtasks. Table 9 lists their names, abbreviations, and sample counts.

Table 9: Subtask composition and sample counts of the XLRS-Bench-lite evaluation used in this work.

Subtask	**Abbreviation**	**QA Pairs**
Perception		
Overall Counting	OC	60
Regional Counting	RC	100
Overall Land Use Classification	OLUC	100
Regional Land Use Classification	RLUC	200
Object Classification	OCC	800
Object Color	OCL	800
Object Motion State	OMS	60
Object Spatial Relationship	OSR	500
Reasoning		
Anomaly Detection and Interpretation	AD	100
Environmental Condition Reasoning	ECR	100
Route Planning	RP	100
Regional Counting with Change Detection	RCCD	60
Counting with Complex Reasoning	CCR	100
Total		3,080

Evaluation metric. Following the aggregation protocol provided by the official XLRS-Bench-lite project, we compute accuracy for each L3 subtask and report the arithmetic mean across all 13 subtasks as average accuracy (Avg. Acc). Let n_t and c_t denote the number of samples and correctly answered samples in subtask t, respectively. The metric is defined as

$$\text{Avg. Acc} = \frac{1}{13}\sum_{t=1}^{13}\frac{c_t}{n_t} \times 100. \tag{12}$$

A.2 LRS-VQA

LRS-VQA (Luo et al., 2025) is a benchmark for open-ended visual question answering on large-format remote sensing imagery. It contains $1{,}657$ images and $7{,}333$ question–answer pairs, with image side lengths reaching $27{,}328$ pixels. The benchmark includes three subsets: LRS-FAIR,

Table 10: Composition of the LRS-VQA benchmark.

Subset	Source Dataset	QA Pairs
LRS-FAIR	FAIR1M-1.0	2,272
LRS-Bridge	GLH-Bridge	1,062
LRS-STAR	STAR	3,999
Total		7,333

Table 11: Subtask composition of MME-RealWorld-RS.

Subtask	QA Pairs
Object Counting	1,255
Color Recognition	1,226
Spatial Relationship Understanding	1,257
Total	3,738

LRS-Bridge, and LRS-STAR. Their question–answer counts are provided in Table 10. The questions cover eight categories: counting, color, category, shape, state, reasoning, urban–rural classification, and object background recognition.

Unlike the multiple-choice format of XLRS-Bench-lite, LRS-VQA provides no candidate answers and requires models to respond with a single word or short phrase. Following the official evaluation script, we determine answer correctness using exact string matching and WordNet-based semantic similarity.

Evaluation metric. We report Average Accuracy (AA) separately for LRS-FAIR, LRS-Bridge, and LRS-STAR. For each subset, we first compute accuracy within each question category and then take the arithmetic mean across categories. Formally, let $\mathcal{C}_s$ denote the set of question categories containing evaluation samples in subset s, and let $n_{s,t}$ and $c_{s,t}$ denote the number of samples and correctly answered samples in category t, respectively. Then,

$$\mathrm{AA}_s = \frac{1}{|\mathcal{C}_s|} \sum_{t \in \mathcal{C}_s} \frac{c_{s,t}}{n_{s,t}} \times 100. \tag{13}$$

A.3 MME-REALWORLD-RS

MME-RealWorld (Zhang et al., 2025b) evaluates multimodal understanding of high-resolution real-world scenes, covering domains such as remote sensing, autonomous driving, surveillance, charts, and optical character recognition. We use only its remote sensing subset, MME-RealWorld-RS, which contains 1,298 images and 3,738 question–answer pairs. This subset covers three perception subtasks: object counting, color recognition, and spatial relationship understanding. Their sample counts are provided in Table 11.

Evaluation metric. Following the aggregation protocol in the official MME-RealWorld evaluation implementation, we report overall accuracy across all question–answer samples in the remote sensing subset. Let n_t and c_t denote the number of samples and correctly answered samples in remote sensing subtask t, respectively. The metric is defined as

$$\mathrm{Acc}_{\mathrm{RS}} = \frac{\sum_{t=1}^{3} c_t}{\sum_{t=1}^{3} n_t} \times 100, \qquad \sum_{t=1}^{3} n_t = 3{,}738. \tag{14}$$

A.4 OFFLINE LEARNING CORPUS

We use the remote sensing training images released with GeoLLaVA-8K (Wang et al., 2026a). The released corpus was constructed with image deduplication within its source datasets and removal of

overlaps with evaluation datasets. Our offline learning uses low-dimensional image descriptors and spatial adjacency co-occurrence statistics to construct the basic visual symbol codebook and symbol merging map, without using question–answer annotations. The learned codebook, merging map, and normalization statistics remain frozen during evaluation, and the target MLLM parameters are unchanged.

B Detailed Algorithms and Implementation

This section provides implementation details and algorithms for GeoCoCo, together with its integration into different MLLMs.

B.1 Offline Visual Symbol Learning

B.1.1 Image descriptor construction.

RGB pixel values are scaled to $[0, 1]$ before descriptor extraction. Let $\mu(\cdot)$ and $\sigma(\cdot)$ denote channel-wise means and population standard deviations over spatial dimensions, respectively, and let $[\cdot;\cdot]$ denote concatenation along the feature dimension. The four components in Eq. equation 1 are defined below.

Color. The color component consists of the means and standard deviations of the RGB channels within each patch:

$$\mathbf{x}_i^{\text{color}} = [\mu(P_i); \sigma(P_i)] \in \mathbb{R}^6. \tag{15}$$

The means describe the overall color, while the standard deviations capture within-patch color variation.

Layout. We divide P_i into 2×2 subpatches, denoted by $P_{i,1}, \ldots, P_{i,4}$ in top-left, top-right, bottom-left, and bottom-right order. The layout component records the deviation of each subpatch's mean RGB values from those of the entire patch:

$$\mathbf{x}_i^{\text{layout}} = [\mu(P_{i,1}) - \mu(P_i); \mu(P_{i,2}) - \mu(P_i); \mu(P_{i,3}) - \mu(P_i); \mu(P_{i,4}) - \mu(P_i)] \in \mathbb{R}^{12}. \tag{16}$$

The fixed concatenation order preserves coarse within-patch color layout, capturing spatial arrangements not represented by overall color statistics.

Texture. Let I_R, I_G, and I_B denote the three channels of the preprocessed RGB image. Its grayscale representation is

$$Y = 0.299 I_R + 0.587 I_G + 0.114 I_B. \tag{17}$$

For interior pixels, horizontal and vertical gradients are computed using central differences:

$$\begin{aligned} G_x(u, v) &= \frac{Y(u, v+1) - Y(u, v-1)}{2}, \\ G_y(u, v) &= \frac{Y(u+1, v) - Y(u-1, v)}{2}, \end{aligned} \tag{18}$$

where (u, v) denotes the pixel's row and column coordinates. One-sided differences are used at image boundaries. The four directional gradient responses and gradient magnitude within patch P_i are

$$\begin{aligned} \mathbf{G}_i &= \left[G_x;\, G_y;\, \frac{G_x + G_y}{\sqrt{2}};\, \frac{G_x - G_y}{\sqrt{2}} \right] \Big|_{P_i}, \\ M_i &= \sqrt{G_x^2 + G_y^2} \Big|_{P_i}, \end{aligned} \tag{19}$$

where $\cdot|_{P_i}$ restricts the responses to the spatial extent of P_i. The texture component is

$$\mathbf{x}_i^{\text{texture}} = [\mu(|\mathbf{G}_i|); \mu(M_i); \sigma(M_i)] \in \mathbb{R}^6. \tag{20}$$

The first four entries are the mean absolute gradient responses along the four directions, and the remaining two are the mean and standard deviation of gradient magnitude. Together, they characterize the direction, strength, and variability of local intensity changes.

Algorithm 1 Offline Visual Symbol Learning of GeoCoCo

Require: Image corpus $\mathcal{I}$; co-occurrence set $\mathcal{D} \subseteq \mathcal{I}$; patch size p; basic symbol count K; target high-level symbol count K_{h}; appearance distance threshold τ
Ensure: Codebook $\mathbf{C}$; symbol merging map π; offline normalization statistics $\boldsymbol{\mu}^{\mathrm{off}}$ and $\boldsymbol{\sigma}^{\mathrm{off}}$

$\mathcal{X} \leftarrow \mathrm{ExtractDescriptors}(\mathcal{I}, p)$
$(\widetilde{\mathcal{X}}, \boldsymbol{\mu}^{\mathrm{off}}, \boldsymbol{\sigma}^{\mathrm{off}}) \leftarrow \mathrm{Standardize}(\mathcal{X})$
$\mathbf{C} \leftarrow \mathrm{KMeans}(\mathrm{Sample}(\widetilde{\mathcal{X}}), K)$
for $I \in \mathcal{D}$ **do**
  $z_i^{(I)} \leftarrow \arg\min_{1 \le k \le K} \left\| \tilde{\mathbf{x}}_i^{(I)} - \mathbf{c}_k \right\|_2^2, \quad \forall i$
  $n_I(a,b) \leftarrow \sum_{(i,j)\in\mathcal{E}_I} \mathbf{1}\big[\{z_i^{(I)}, z_j^{(I)}\} = \{a,b\}\big], \quad \forall\, 1 \le a < b \le K$
end for
$S_0(a,b) \leftarrow S_0(b,a) \leftarrow \sum_{I\in\mathcal{D}} \dfrac{n_I(a,b)}{\max\Big(1, \sum_{1\le r<s\le K} n_I(r,s)\Big)}, \quad \forall\, 1 \le a < b \le K$
$\mathcal{V}_0 \leftarrow \{1, \dots, K\}, \qquad \mathcal{B}_0(a) \leftarrow \{a\}\ \forall a \in \mathcal{V}_0, \qquad t \leftarrow 0$
while $|\mathcal{V}_t| > K_{\mathrm{h}}$ **do**
  $\mathcal{P}_t \leftarrow \{(a,b) \mid a,b \in \mathcal{V}_t,\ a < b,\ S_t(a,b) > 0,\ D_t(a,b) \le \tau\}$
  if $\mathcal{P}_t = \varnothing$ **then**
    break
  end if
  $(a^*, b^*) \leftarrow \arg\max_{(a,b)\in\mathcal{P}_t} S_t(a,b)$
  $h \leftarrow K + t + 1$
  $\mathcal{B}_{t+1} \leftarrow \mathcal{B}_t, \qquad S_{t+1} \leftarrow S_t$
  $\mathcal{B}_{t+1}(h) \leftarrow \mathcal{B}_t(a^*) \cup \mathcal{B}_t(b^*)$
  $S_{t+1}(h,d) \leftarrow S_{t+1}(d,h) \leftarrow S_t(a^*,d) + S_t(b^*,d), \quad \forall d \in \mathcal{V}_t \setminus \{a^*, b^*\}$
  $\mathcal{V}_{t+1} \leftarrow (\mathcal{V}_t \setminus \{a^*, b^*\}) \cup \{h\}$
  $t \leftarrow t + 1$
end while
Enumerate final symbols as $\mathcal{V}_t = \{h_1, \dots, h_{|\mathcal{V}_t|}\}$
$\pi(k) \leftarrow r, \quad \forall r = 1, \dots, |\mathcal{V}_t|, \quad \forall k \in \mathcal{B}_t(h_r)$
return $\mathbf{C}, \pi, \boldsymbol{\mu}^{\mathrm{off}}, \boldsymbol{\sigma}^{\mathrm{off}}$

Context. We arrange $\mu(P_i)$ according to their positions in the 2D patch grid. Let $B_i^{(s)}$ denote the $s \times s$ neighborhood of these vectors centered at patch i. The context component is

$$\mathbf{x}_i^{\mathrm{context}} = \Big[\mu(B_i^{(3)}); \sigma(B_i^{(3)}); \mu(B_i^{(7)}); \sigma(B_i^{(7)})\Big] \in \mathbb{R}^{12}. \tag{21}$$

Each scale contributes three channel means and three standard deviations. Replicate padding is applied at grid boundaries. The two scales capture neighborhood appearance over different spatial extents, complementing the statistics within each patch.

The resulting descriptors are standardized dimension-wise using offline corpus statistics, which remain frozen during online inference.

B.1.2 Offline Learning Algorithm

Algorithm 1 summarizes offline visual symbol learning. Let $\mathcal{I}$ denote the complete offline image corpus and $\mathcal{D} \subseteq \mathcal{I}$ the co-occurrence learning set. The active symbol set $\mathcal{V}_t$ and membership sets $\mathcal{B}_t(a)$ follow the definitions in Sec. 3.1.

B.2 Online Regional Token Compression

B.2.1 Region-to-Token Grid Alignment

The image patch grid used for compression planning may differ in resolution from the model's visual token grid. We therefore map the patch-level region partition $\{\mathcal{R}_r\}_{r=1}^{R}$ to the target visual token grid using their relative spatial coordinates.

Let H_{p} and W_{p} denote the numbers of rows and columns in the image patch grid, and let H_{v} and W_{v} denote those in the visual token grid. When $H_{\mathrm{p}} \geq H_{\mathrm{v}}$ and $W_{\mathrm{p}} \geq W_{\mathrm{v}}$, we map the row and column boundaries of each target grid cell to the image patch grid by scaling them by $H_{\mathrm{p}}/H_{\mathrm{v}}$ and $W_{\mathrm{p}}/W_{\mathrm{v}}$, respectively, and rounding down to integer boundaries. These boundaries define the corresponding patch window for each visual token.

Let Ω_j denote the set of patch indices within this window. The region label of visual token j is assigned by majority voting:

$$\tilde{\ell}_j = \underset{r \in \{1,\ldots,R\}}{\arg\max} \, |\Omega_j \cap \mathcal{R}_r| \,, \tag{22}$$

where $\tilde{\ell}_j$ denotes the region label before reindexing. When the two grids have identical dimensions, this operation reduces to a one-to-one label mapping. When $H_{\mathrm{p}} < H_{\mathrm{v}}$ or $W_{\mathrm{p}} < W_{\mathrm{v}}$, we instead apply nearest-neighbor resampling using the same row and column coordinate ratios, so that each visual token inherits the region label of its corresponding image patch.

After label mapping, we re-extract four-connected components on the target token grid and consecutively reindex them to obtain ℓ_j and construct the initial groups in Eq. equation 10. Spatially disconnected parts sharing the same mapped label are thus treated as separate initial groups.

B.2.2 Online Inference Algorithm

Algorithm 2 summarizes the online visual token compression procedure. CPU-based regional planning runs concurrently with GPU-based visual processing up to the model-specific compression point. Once both branches finish, visual features are aggregated on the GPU, and token order and position indices are updated to produce the compact visual sequence.

Ward-based budget adjustment. When $M_0 > M$, we follow the Ward merging criterion used in scikit-learn, restricting merges to regions sharing a four-neighbor edge on the visual token grid. For adjacent regions A and B, the merging cost is

$$\Delta(A, B) = \frac{|A||B|}{|A| + |B|} \, \|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|_2^2 \,, \tag{23}$$

where $|A|$ and $|B|$ count the original visual tokens in each region, and $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are the means of their token-grid-aligned, standardized image descriptors. We repeatedly merge the adjacent pair with the smallest cost until M regions remain.

B.3 Integration with Different MLLMs

GeoCoCo keeps the target MLLM parameters frozen and generates regional compression plans directly from input images. This section describes model-specific adaptations in compression placement, image tile organization, and position index handling.

Qwen2.5-VL. We preserve the original visual encoding and spatial merging processes and apply region-level feature aggregation before visual tokens enter the language model. For multi-image inputs, each image is planned independently, and no token group spans different images. After aggregation, we update the visual-token placeholders, attention mask, and multi-axis position indices. For each image span, we maintain a cumulative offset on each positional axis based on the change in that axis's maximum index before and after compression, and apply it to subsequent tokens.

Qwen3-VL. To accommodate its multi-level visual feature injection, we apply the same regional compression plan to the final visual features and all DeepStack features after their respective visual connectors. Each branch is aggregated independently using identical token groups and output order, preserving spatial correspondence across feature levels and the original injection paths. Visual position indices are updated according to the compression plan following Qwen2.5-VL.

InternVL3 and InternVL3.5. We use the same integration strategy for both models, preserving their dynamic tiling strategy and aggregating visual features before language-model processing. For each original image, we generate a region partition from image descriptors and align it with the

Algorithm 2 Online Regional Token Compression of GeoCoCo

Require: Image I; frozen codebook $\mathbf{C}$ and mapping π; normalization statistics $\boldsymbol{\mu}^{\mathrm{off}}, \boldsymbol{\sigma}^{\mathrm{off}}$; patch size p; patch-grid size $(H_{\mathrm{p}}, W_{\mathrm{p}})$; token-grid size $(H_{\mathrm{v}}, W_{\mathrm{v}})$; original position indices; target token count M
Ensure: Compressed visual tokens $\{\bar{\mathbf{f}}_m\}_{m=1}^{M}$ and updated position indices
Execute the CPU and GPU branches concurrently
CPU branch: Regional compression planning
$\{\mathbf{x}_i\}_{i=1}^{N} \leftarrow \mathrm{ExtractDescriptors}(I, p)$
$\tilde{\mathbf{x}}_i \leftarrow \mathrm{Standardize}(\mathbf{x}_i; \boldsymbol{\mu}^{\mathrm{off}}, \boldsymbol{\sigma}^{\mathrm{off}}), \quad \forall i$
$z_i \leftarrow \arg\min_{1 \le k \le K} \|\tilde{\mathbf{x}}_i - \mathbf{c}_k\|_2^2, \qquad \hat{z}_i \leftarrow \pi(z_i), \quad \forall i$
$\widehat{\mathcal{E}}_I \leftarrow \{(i,j) \in \mathcal{E}_I \mid \hat{z}_i = \hat{z}_j\}$
$\{\mathcal{R}_r\}_{r=1}^{R} \leftarrow \mathrm{CC}(\{1, \ldots, N\}, \widehat{\mathcal{E}}_I)$
$(\{\ell_j\}_{j=1}^{N_{\mathrm{v}}}, M_0) \leftarrow \mathrm{AlignRegions}(\{\mathcal{R}_r\}_{r=1}^{R}, (H_{\mathrm{p}}, W_{\mathrm{p}}), (H_{\mathrm{v}}, W_{\mathrm{v}}))$
$\mathcal{T}_m^{(0)} \leftarrow \{j \in \{1, \ldots, N_{\mathrm{v}}\} \mid \ell_j = m\}, \quad m = 1, \ldots, M_0$
$\mathcal{A} \leftarrow \varnothing$
if $M_0 > M$ **then**
    $\{\tilde{\mathbf{x}}_j^{\mathrm{v}}\}_{j=1}^{N_{\mathrm{v}}} \leftarrow \mathrm{AlignDescriptors}(\{\tilde{\mathbf{x}}_i\}_{i=1}^{N}, (H_{\mathrm{p}}, W_{\mathrm{p}}), (H_{\mathrm{v}}, W_{\mathrm{v}}))$ ▷ For Ward costs
    $\{\mathcal{T}_m\}_{m=1}^{M} \leftarrow \mathrm{AdjacentWardMerge}(\{\mathcal{T}_m^{(0)}\}_{m=1}^{M_0}, \{\tilde{\mathbf{x}}_j^{\mathrm{v}}\}_{j=1}^{N_{\mathrm{v}}}, (H_{\mathrm{v}}, W_{\mathrm{v}}), M)$
else if $M_0 < M$ **then**
    $(\{\mathcal{T}_m\}_{m=1}^{M}, \mathcal{A}) \leftarrow \mathrm{AllocateOutputsAndAnchors}(\{\mathcal{T}_m^{(0)}\}_{m=1}^{M_0}, M)$
else
    $\mathcal{T}_m \leftarrow \mathcal{T}_m^{(0)}, \quad m = 1, \ldots, M$
end if
GPU branch: Visual computation up to the compression point
$\{\mathbf{f}_j\}_{j=1}^{N_{\mathrm{v}}} \leftarrow \mathrm{EncodeToCompressionPoint}(I)$
Synchronization and aggregation
Wait until both branches complete
Transfer grouping metadata $\{\mathcal{T}_m\}_{m=1}^{M}$ and anchor metadata $\mathcal{A}$ to the GPU
$\bar{\mathbf{f}}_m \leftarrow \frac{1}{|\mathcal{T}_m|} \sum_{j \in \mathcal{T}_m} \mathbf{f}_j, \quad m = 1, \ldots, M$
Update token order and position indices using $\{\mathcal{T}_m\}_{m=1}^{M}$, $\mathcal{A}$, and the original position indices
Update the corresponding sequence metadata
return $\{\bar{\mathbf{f}}_m\}_{m=1}^{M}$ and updated position indices

full visual token grid induced by the dynamic tiling layout. The aligned label map is then cropped to individual tiles, where connected components are recomputed. Aggregation is restricted to each local tile. The global thumbnail is retained without compression. Given the total output token budget M, we reserve N_{th} tokens for the thumbnail and allocate the remaining $M - N_{\mathrm{th}}$ tokens to the local tiles, where N_{th} denotes the thumbnail token count. The compressed local-tile features and unchanged thumbnail features are concatenated in the original order to replace the corresponding visual token spans. Text tokens and image delimiters are preserved, while attention masks and contiguous one-dimensional position indices are updated.

GeoLLaVA-8K. GeoLLaVA-8K already incorporates background token pruning (BTP) and anchored token selection (ATS). We replace these native compression modules with GeoCoCo rather than applying additional compression to their outputs. We preserve the original high-resolution tiling and visual encoding pipeline, generate regional compression plans directly from each input tile's pixel information. The aggregated features then pass through the original projector into the language model. Subsequent visual sequence assembly and one-dimensional position index construction follow the original model pipeline.

B.4 Experimental Configurations

For offline descriptor extraction and online GeoCoCo planning, the image long-side limit is set to 2048 pixels. K-means runs for at most 50 iterations with a random seed of 0. The co-occurrence learning set $\mathcal{D}$ contains $1{,}024$ images sampled without replacement from the offline corpus, also using a random seed of 0. The SLIC-based experiments use at most 10 iterations, 9 candidate centers per token, and a spatial weight of 0.1. For the cross-benchmark comparison in Table 6, Avg. Acc. is

the unweighted mean of the MME-RealWorld-RS accuracy and the three LRS-VQA subset scores. We reproduce the baselines based on their official implementations and evaluate them using LMMs-Eval (Zhang et al., 2025a), with a batch size of 1 per inference process. The experiments ran on a server with Intel Xeon Platinum 8470Q CPU and NVIDIA RTX PRO 6000 GPU, all experiments are repeated three times, and the reported results are averaged over the three runs.

C Additional Experiments

This section provides detailed results omitted from the main text and additional experiments with different symbol merging thresholds, token compression ratios, and input resolutions.

C.1 Detailed Results

Table 12 provides the full per-subtask results at a 50% token compression ratio on XLRS-Bench-lite using Qwen2.5-VL-7B. GeoCoCo achieves the highest average accuracy and lowest prefill time among the compared methods.

Table 12: Comparison of token compression methods at compression ratios of 50% on Qwen2.5-VL-7B.

Method	**Perception**								**Reasoning**					**Avg. Acc↑**	**Avg. Tokens↓**	**Prefill time↓**
Sub-tasks	OC	RC	OLUC	RLUC	OCC	OCL	OMS	OSR	AD	ECR	RP	RCCD	CCR	(%)	(tokens)	(ms)
Compression Method on Qwen2.5-VL-7B																
+FastV	35.0	39.0	32.0	77.0	39.1	40.5	66.7	35.4	68.0	73.0	26.0	35.0	44.0	47.0	8.0k (↓50%)	1441
+VisionZip	35.0	38.0	35.0	77.5	38.6	40.8	66.7	36.4	69.0	74.0	24.0	36.7	42.0	47.2	8.0k (↓50%)	1543
+DivPrune	31.7	38.0	31.0	76.5	40.3	39.6	66.7	36.4	68.0	71.0	26.0	36.7	41.0	46.4	8.0k (↓50%)	1966
+DUET-VLM	33.3	41.0	33.0	76.5	40.1	41.1	66.7	36.6	68.0	74.0	27.0	36.7	44.0	47.5	8.0k (↓50%)	2134
+IVC-Prune	20.0	39.0	24.0	72.5	42.1	38.5	66.7	34.8	71.0	78.0	31.0	53.3	42.0	47.2	8.0k (↓50%)	1783
+GeoCoCo	31.7	41.0	39.0	81.5	40.3	37.8	66.7	35.8	69.0	78.0	31.0	48.3	43.0	**49.5**	8.0k (↓50%)	**1399**

C.2 Different Symbol Merging Thresholds and Token Compression Ratios

Table 13 reports results with different symbol merging thresholds and token compression ratios. At an 80% token compression ratio, setting τ to the 10th, 20th, and 30th percentiles yields average accuracies of 49.2%, 49.1%, and 49.7%, respectively, with approximately 3.3k retained visual tokens in each setting.

With the default threshold $\tau = Q_{30\%}$, average accuracy ranges from 49.1% to 49.7% across the tested token compression ratios of 50%–90%. At 90% token compression, GeoCoCo retains approximately 1.6k visual tokens and achieves 49.1% average accuracy. Both 70% and 80% token compression achieve 49.7%.

Table 13: GeoCoCo results with different symbol merging thresholds and token compression ratios on XLRS-Bench-lite using Qwen2.5-VL-7B. The default configuration uses $\tau = Q_{30\%}$ and 80% token compression. $Q_{q\%}$ denotes the qth percentile of squared codebook-center distances over distinct basic visual symbol pairs observed as neighbors.

Settings	**Perception**								**Reasoning**					**Avg. Acc↑**	**Avg. Tokens↓**
Sub-tasks	OC	RC	OLUC	RLUC	OCC	OCL	OMS	OSR	AD	ECR	RP	RCCD	CCR		
GeoCoCo (default)	31.7	43.0	42.0	76.5	39.8	40.3	66.7	34.8	69.0	80.0	31.0	50.0	41.0	**49.7**	3.3k
Effect of the quantile threshold τ															
$\tau = Q_{10\%}$	33.3	39.0	38.0	76.0	38.9	36.9	66.7	33.8	71.0	80.0	30.0	55.0	41.0	49.2	3.3k
$\tau = Q_{20\%}$	31.7	39.0	38.0	77.0	40.0	38.0	66.7	34.0	73.0	79.0	31.0	51.7	39.0	49.1	3.3k
Effect of the token compression ratio															
ratio $= 90\%$	31.7	42.0	36.0	76.0	39.8	39.0	66.7	34.4	69.0	78.0	31.0	53.3	42.0	49.1	1.6k
ratio $= 70\%$	30.0	42.0	42.0	76.0	40.3	38.9	66.7	34.8	69.0	79.0	32.0	51.7	44.0	**49.7**	4.9k
ratio $= 60\%$	31.7	40.0	38.0	76.5	41.1	38.1	66.7	35.4	68.0	79.0	32.0	53.3	44.0	49.5	6.6k
ratio $= 50\%$	31.7	41.0	39.0	81.5	40.3	37.8	66.7	35.8	69.0	78.0	31.0	48.3	43.0	49.5	8.0k

C.3 Different Input Resolutions

Table 14 compares GeoCoCo with the uncompressed model under six maximum input pixel budgets, with GeoCoCo using an 80% token compression ratio throughout. GeoCoCo improves average accuracy by 2.2–3.7 percentage points across all six settings, reaching 50.5% at the two largest pixel budgets. Prefill time decreases at the four larger budgets but increases at the two smallest budgets despite the accuracy gains. This pattern is consistent with shorter GPU visual encoding being unable to fully hide CPU planning overhead, while compressing already short visual sequences offers smaller downstream computational savings. These results align with GeoCoCo's design for UHR remote sensing imagery, where longer visual encoding provides a larger window for overlapping CPU planning and longer visual sequences offer greater computational savings through token compression.

Table 14: Generalization across different input image resolutions on Qwen2.5-VL-7B on XLRS-Bench-lite.

Settings	**Perception**								**Reasoning**					**Avg. Acc↑**	**Avg. Tokens↓**	**Prefill Time↓**
Sub-tasks	OC	RC	OLUC	RLUC	OCC	OCL	OMS	OSR	AD	ECR	RP	RCCD	CCR			
max_pixels = 2,048 × 28 × 28																
Qwen2.5-VL-7B	28.3	39.0	29.0	72.5	32.1	38.0	66.7	36.2	66.0	73.0	23.0	36.7	47.0	45.2	2.1k	**189**
+GeoCoCo	23.3	43.0	33.0	73.5	33.6	37.8	66.7	34.0	70.0	79.0	31.0	46.7	44.0	**47.4**	0.4k (↓80%)	467
max_pixels = 4,096 × 28 × 28																
Qwen2.5-VL-7B	25.0	37.0	29.0	75.5	37.3	39.0	66.7	35.8	68.0	74.0	27.0	36.7	46.0	45.9	4.2k	**342**
+GeoCoCo	33.3	46.0	33.0	70.5	37.0	37.0	66.7	35.6	69.0	80.0	31.0	48.3	40.0	**48.3**	0.8k (↓80%)	590
max_pixels = 8,192 × 28 × 28																
Qwen2.5-VL-7B	33.3	36.0	28.0	78.5	38.6	39.5	66.7	36.6	67.0	73.0	26.0	36.7	42.0	46.3	8.2k	871
+GeoCoCo	26.7	43.0	36.0	79.5	38.6	37.8	66.7	35.0	68.0	81.0	32.0	46.7	41.0	**48.6**	1.6k (↓80%)	**684**
max_pixels = 16,384 × 28 × 28																
Qwen2.5-VL-7B	33.3	40.0	31.0	77.0	40.6	40.5	66.7	36.2	68.0	72.0	27.0	38.3	45.0	47.4	16.1k	1924
+GeoCoCo	31.7	43.0	42.0	76.5	39.8	40.3	66.7	34.8	69.0	80.0	31.0	50.0	41.0	**49.7**	3.3k (↓80%)	**1141**
max_pixels = 32,768 × 28 × 28																
Qwen2.5-VL-7B	35.0	36.0	30.0	79.5	42.6	41.6	66.7	34.4	67.0	71.0	27.0	36.7	45.0	47.1	30.9k	4954
+GeoCoCo	36.7	40.0	47.0	79.0	42.6	38.6	66.7	30.8	69.0	78.0	32.0	50.0	46.0	**50.5**	6.2k (↓80%)	**2765**
max_pixels = 65,536 × 28 × 28																
Qwen2.5-VL-7B	36.7	36.0	30.0	75.0	43.6	40.9	66.7	33.4	67.0	73.0	28.0	36.7	41.0	46.8	54.0k	9501
+GeoCoCo	40.0	39.0	43.0	72.0	40.6	39.9	66.7	32.0	70.0	78.0	36.0	55.0	44.0	**50.5**	10.8k (↓80%)	**6736**

C.4 Additional Ablations

Table 15: Effect of token budget adjustment on XLRS-Bench-lite using Qwen2.5-VL-7B.

Settings	**Perception**								**Reasoning**					**Avg. Acc↑**	**Avg. Tokens↓**	**Prefill Time↓**
Sub-tasks	OC	RC	OLUC	RLUC	OCC	OCL	OMS	OSR	AD	ECR	RP	RCCD	CCR			
GeoCoCo	31.7	43.0	42.0	76.5	39.8	40.3	66.7	34.8	69.0	80.0	31.0	50.0	41.0	**49.7**	3.3k (↓80%)	**1141**
w/o Token Budget	30.0	41.0	41.0	79.0	40.1	38.4	66.7	34.4	69.0	78.0	31.0	51.7	44.0	49.6	4.6k (↓72%)	1210
w/o Co-occurrence Merging	31.7	40.0	39.0	77.0	39.1	37.3	66.7	33.4	73.0	79.0	31.0	50.0	39.0	48.9	3.3k (↓80%)	1147

Token budget adjustment. We examine the effect of token budget adjustment by disabling both adjacent Ward merging and regional-mean replication. This variant directly aggregates each initial region into one visual token, yielding M_0 output tokens per image.

As shown in Table 15, removing token budget adjustment yields 49.6% average accuracy, close to the default 49.7%. Thus, directly aggregating the co-occurrence-guided regions preserves GeoCoCo's accuracy without subsequent budget adjustment. Together with the accuracy improvement from 49.0% to 49.7% introduced by co-occurrence-guided grouping in Table 8a, these results support the contribution of the learned co-occurrence prior to task accuracy. Budget adjustment primarily controls the output token count, reducing the average count from 4.6k to 3.3k and prefill time from 1210 to 1141 ms while maintaining comparable accuracy.

Appearance-only symbol merging. We replace co-occurrence-guided merging with appearance-only merging while retaining the same basic codebook and number of high-level visual symbols.

At each step, this variant merges the groups containing the closest pair of basic-symbol centers, measured by squared Euclidean distance, without using co-occurrence statistics or an additional distance threshold. The online compression pipeline remains unchanged. As shown in Table 15, appearance-only merging achieves 48.9% average accuracy, compared with 49.7% for GeoCoCo, at similar prefill times of 1147 and $1141\,\mathrm{ms}$, respectively. This comparison supports learning the shared compression prior from spatial co-occurrence rather than appearance similarity alone.

D LIMITATIONS

We primarily evaluate GeoCoCo on RGB remote sensing imagery. Adapting its low-dimensional image descriptors and shared compression prior to other modalities, such as multispectral and SAR imagery, remains a direction for future work.

The reported latencies are measured under the hardware and execution configurations used in this work. Evaluating and optimizing GeoCoCo across additional hardware platforms and concurrent-request workloads also warrants further investigation.