Swiss Federal Institute of Technology in Lausanne
School of Computer and Communication Sciences

Master Thesis in Data Science

Structured Representations for Fine-Grained Text-to-Image Retrieval in Remote Sensing

Oussama Gabouj
oussama.gabouj@epfl.ch

Hosted by AXA Group Operations

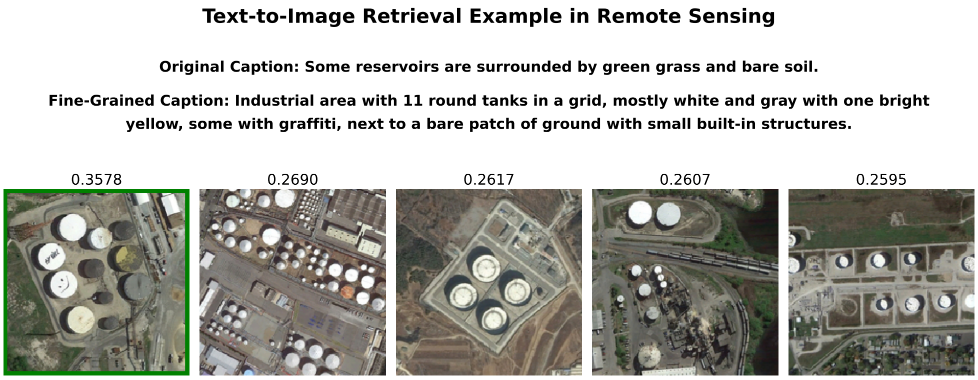
EPFL Supervisor
Prof. Devis Tuia
EPFL ENAC
devis.tuia@epfl.ch

AXA Supervisor
Ciprian Tomoiaga
AXA Group Operations
ciprian.tomoiaga@axa.com

August 2025

Abstract

Contrastive Language–Image Pre-training (CLIP) has shown strong performance across diverse vision–language tasks, and many studies have fine-tuned it for remote sensing (RS) applications, including text-to-image retrieval. However, CLIP's standard design, which represents both image and text as single global vectors for alignment, limits its ability to handle RS imagery, characterized by densely packed objects, fine-grained structures, and complex spatial and semantic relationships. To address these limitations, we introduce a novel RS retrieval framework that jointly leverages global and structured representations to capture both semantic context and detailed spatial relations. Our contributions are: (1) the creation of fine-grained RS caption datasets using vision–language models (VLMs), validated through human preference studies for improved relevance and grounding; (2) the introduction of multimodal structured embeddings using graph- and scene-based representations; (3) the development of graph based retrieval strategies: Graph-to-Graph (G2G), a hybrid VectorGraph-to-VectorGraph (VG2VG) using Graph Matching Networks, and (4) a lightweight, non trainable Scene-based matching for resource-limited settings. On a fine-grained benchmark, VG2VG outperforms all baselines by a +8.30% mean accuracy gain, while scene-based matching achieves +3.55% with minimal computation. By combining CLIP's global semantic strengths with fine-grained, structured understanding, our framework delivers state-of-the-art performance in RS cross-modal retrieval and offers a scalable, plug-and-play solution for text-to-image retrieval.


Text-to-Image Retrieval Example in Remote Sensing
Original Caption: Some reservoirs are surrounded by green grass and bare soil.
Fine-Grained Caption: Industrial area with 11 round tanks in a grid, mostly white and gray with one bright yellow, some with graffiti, next to a bare patch of ground with small built-in structures.
0.3578
0.2690
0.2617
0.2607
0.2595

Contents

1 Introduction

Foundation models, which learn to comprehend both images and text from extensive datasets, have demonstrated significant potential. While predominantly trained on natural images, significant advancements are now being made in remote sensing (RS), a domain that presents distinct challenges, including dense object distributions, low spatial resolution, and a scarcity of annotated data. This progress has shifted focus from traditional closed-set tasks to Open-Vocabulary (OV) approaches, allowing models to understand new and unseen concepts described in natural language. This has enabled various OV applications in RS, powered by different Vision-Language Models (VLMs) that can be categorized into contrastive-based, generation-based, and instruction-tuned models, each serving distinct purposes: generation-based models used for image captioning, instruction-tuned models facilitate guided tasks like object detection, and contrastive models, such as CLIP, are particularly effective for retrieval. This thesis focuses on text-to-image retrieval in RS, a task that involves retrieving the most relevant image(s) given a textual query. This is highly relevant for industries like insurance, where it can be used for assessing risk and verifying claims by retrieving specific visual evidence from satellite imagery based on descriptive reports.

Existing research commonly employs contrastive models based on CLIP for cross-modal retrieval, which encode an entire image and text query into single global vectors within a shared embedding space. However, representing an image as a single vector struggles to capture critical information such as object counts and spatial arrangements. Similarly, compressing detailed, descriptive captions into a single vector leads to loss of important contextual cues. This semantic oversimplification restricts fine-grained matching. Moreover, existing datasets tend to be too small and generic, limiting their effectiveness for fine-grained retrieval tasks in remote sensing.

In this research project, we introduce novel fine-grained datasets, developed as an extension of existing RS datasets, which has been validated through human evaluation. Additionally, we propose a comprehensive framework designed for cross-modal retrieval within the RS domain. Our approach addresses the limitations of single-vector embeddings by integrating both global and structured representations. It incorporates graph-based and scene-level features to improve semantic alignment between text and images. The main contributions are summarized as follows:

- **Datasets Benchmarking and Fine-Grained Caption Generation.** We conduct qualitative and quantitative evaluation of existing RS image-caption datasets to assess their suitability for retrieval tasks. We also introduce a new fine-grained datasets generated using vision language models (VLMs). These fine-grained datasets are validated through human-centric evaluations comparing VLM-generated captions to original ones.

- **Multimodal structured representations for text and image.** We introduce structured representations for both modalities, including graph and scene generation for images and captions. These representations capture object counts and spatial semantics relationships, which are critical for fine-grained retrieval in RS.

- **Hybrid retrieval with graph and global matching.** We propose a novel retrieval framework that combines Graph-to-Graph (G2G) and VectorGraph-to-VectorGraph (VG2VG) matching to support both fine-grained semantic alignment and global context retrieval. We also introduce a Graph Matching Network (GMN) that handles variable-length graph inputs, uses histogram-based similarity and attention mechanisms, to produce a similarity score between two graphs.

- **Scene-based matching as a lightweight alternative to graph construction.** We propose scene segmentation and alignment as a cheaper alternative to explicit graph construction. A matching algorithm is introduced that operates on these semantic scene regions, providing a middle-ground between full graph matching and global vector embeddings.

2 Related Work

2.1 Remote Sensing foundation models

Remote sensing (RS) analysis is evolving rapidly with the rise of vision-language models (VLMs) and multimodal large language models (MLLMs). Unlike traditional models, foundation models are pre-trained on large, diverse image-text datasets, giving them a broad understanding of the world. Extensive reviews by [43] and [38] describe this rapidly evolving field, outlining the major VLM architectures: contrastive-based, generation-based, and instruction-based models. These architectures support a wide range of tasks, including image captioning, scene classification, segmentation, detection, target recognition, visual question answering, and cross-modal retrieval. Recent models like EarthGPT [49], LHRS-Bot [25], VHM [29], SkyEyeGPT [47], RS-LLaVA [2], and GeoLLaVA-8K [40] highlight a growing trend toward building unified systems that can handle a wide range of remote sensing tasks. At the same time, models such as RemoteCLIP [19] and GeoRSCLIP [51] emphasize aligning visual and textual representations to boost performance on downstream tasks. These advancements provide valuable insights for our work, which focuses on improving text-to-image retrieval. By integrating image and text encoders, we aim to effectively align visual features with textual embeddings, thereby enhancing cross-modal performance.

2.2 Existing Remote Sensing Datasets

The development of robust RS models has relied on high-quality, large-scale, and diverse datasets that capture the complexities of RS imagery. Early work introduced foundational datasets for image captioning and image–text matching, including UCM-Captions [31], RSICD [23], RSITMD [16], Sydney-Captions [30], and NWPU-Captions [4], which provided crucial benchmarks for early VLMs. The field has since shifted toward more comprehensive benchmarks, such as RS5M [51] with 5M RS images, FIT-RS [24] for instruction tuning, and STAR [17] for scene graph generation in large RS imagery. However, many of these datasets still lack diversity and rely on generic captions, limiting their effectiveness for retrieval. We will conduct an in-depth analysis, create new fine-grained data by recaptioning the original datasets, and validate the results through human evaluation as the gold standard.

2.3 Remote Sensing Image-Text Retrieval

RS image-text retrieval involves efficiently extracting information by connecting textual queries with relevant images. Early approaches like GaLR [46] and AMFMN [45] focus on integrating global and local information, capturing various features to improve retrieval performance. RS-CapRet [37] combines RS-specific image encoders with large decoder language models, achieving state-of-the-art in image captioning and text-image retrieval. RS-M-CLIP [36] expands this by exploring multilingual vision language pre-training, improving cross-modal and multilingual retrieval. To tackle the challenges of handling long captions, LRSCLIP [3] presents a novel architecture designed to better align RS images with detailed textual descriptions. Additionally, methods like KAMCL [10], the EBAKER [11], and iEBAKER [50] focus on filtering weakly correlated pairs and reasoning about keywords, improving retrieval accuracy. Models like RemoteCLIP [19] and GeoRSCLIP [51] showcase robust visual features and text embeddings, essential for seamless retrieval. However, existing methods often rely on simple vector-based similarity, embedding entire images into a single vector, as in CLIP, which enables efficient retrieval but fails to capture the complexity of RSimagery,y, such as dense objects, spatial layouts, and intricate relationships. To address this, we propose using graph- and scene-based embeddings that capture object density, spatial structure, and inter-object relationships, enabling richer representation of RS imagery.

2.4 Open-Vocabulary RS Scene Understanding: Detection, Segmentation, and Scene Graph

With the emergence of VLMs and foundation models, RS analysis is shifting toward open-vocabulary (OV) approaches that overcome the limits of traditional closed-vocabulary (CV) systems restricted to predefined categories. OV models enable flexible interpretation of complex scenes by generalizing to novel concepts described in natural language, which is crucial for tasks like object detection, semantic segmentation, and scene graph generation. For OV object detection (OVOD), models like LAE [28] address the domain gap between natural and RS imagery, enabling broader detection capabilities. Similarly, Scale-MAE [33] and SatMAE [5] incorporate scale-awareness to support detailed land cover classification. Instruction-tuned models like EarthMarker [48] and InstructSAM [52] demonstrate how flexible, prompt-based recognition can be achieved without task-specific training, offering more adaptive tools for RS analysis. For semantic segmentation and scene understanding, OV models are advancing toward capturing not only object categories but also multi-level relationships among them. Notable examples include SkySenseGPT [24] and SkyEyeGPT [47], which support multi-granularity vision-language comprehension and begin to bridge the gap toward scene graph generation (SGG) in RS. These models lay the groundwork for extracting structured semantic representations. Since our project aims to generate high-quality scene representations in OV settings, we rely on these architectures to support tasks such as object detection, segmentation, and spatial/semantic relationship extraction, all foundational components of SGG. By integrating VLMs capable of generating OV-SGG, we aim to build a robust framework for open-ended cross-modal retrieval.

2.5 Open Vocabulary Scene Graph Generation for Natural Images

Open-Vocabulary SGG (OV-SGG) enhances visual understanding by modeling objects, attributes, and relationships. While scene graph generation remains challenging in RS imagery, significant progress has been made for natural images. Early efforts, like [8], introduced prompt-based finetuning for OV-SGG. More recent approaches, such as INOVA [14], model object interactions to improve relation recognition, while methods like those by [15] use VLM for image-to-graph conversion. Other advancements, like the RAHP framework [20], refine OV-SGG by integrating relation information based on subject-object and region-specific contexts. Despite these advancements, generating scene graphs for RS data remains challenging, and we will leverage powerful VLMs for SG generation.

2.6 Graph Matching Algorithms and Late Interaction Mechanism

Scene graphs are crucial for graph matching tasks like image-text and image-image retrieval, where understanding object interactions is key. The late integration mechanism, introduced in ColPali [7], computes interactions between the query and database at inference time. Unlike traditional cosine similarity-based methods, it uses multi-vector assignment algorithms followed by similarity scoring based on the matching output. In this context, graph matching aims to compute the similarity between all possible node pairs in two graphs and then solve a one-to-one or one-to-many matching problem. For one-to-one matching, the Hungarian algorithm [13] is a well-established method. For one-to-many matching, approaches like ColPali are more suitable. However, instead of solving these complex problems directly, neural networks can be trained to produce a graph similarity score. Early work by Johnson et al. [12] demonstrated the use of scene graphs as queries for image retrieval. Recent advancements, such as Graph Structured Network Matching (GSMN) [18] and Local and Global Scene-Graph Matching (LGSGM) [26], have improved retrieval by learning fine-grained correspondences and integrating graph convolution networks. Cross-modal scene graph matching [41] unifies image and text representations, while efficient graph similarity techniques like SimGNN [1] enhance the effectiveness of graph-based approaches. We aim to adapt these matching techniques to enhance the analysis and retrieval capabilities in RS applications.

3 Evaluating and Enhancing Caption Diversity in Remote Sensing Datasets

This section provides an overview and exploratory analysis of existing RS datasets used for retrieval tasks. We closely examine each dataset and investigate its diversity, which serves as a core metric for effective retrieval. In practice, the data must exhibit sufficient diversity for human evaluators. While AI models can distinguish between images and captions even when the differences are minimal, humans often perceive these instances as equivalent. As a result, overfitting to dataset-specific cues can mislead the interpretation of retrieval performance. To address this, we introduce several evaluation metrics designed to better capture the quality of the datasets for the retrieval task and help interpret retrieval results beyond raw scores. Furthermore, we construct new fine-grained caption datasets derived from the original datasets and conduct a comparison with its original version.

3.1 Existing Remote Sensing Datasets

Table 1 provides a brief overview of the most widely used datasets for retrieval tasks:

UCM-Captions [31]: Derived from the UC Merced land-use dataset, it contains 2,100 aerial images (21 classes, 100 images each), each annotated with five human-written captions.

RSICD [23] : Comprising 10,921 Google Earth images labeled with five human-written captions across diverse scenes (urban, rural, industrial), it serves as a standard benchmark for remote sensing captioning and retrieval.

RSITMD [30]: Includes 4,743 image-text pairs for fine-grained cross-modal retrieval, focusing on multiscale image–text alignment in remote sensing.

Sydney-Captions [30]: Comprises 2,100 high-resolution aerial images from Sydney, each annotated with five captions, emphasizing urban scenes.

NWPU-Captions [4]: Built on the NWPU-RESISC45 benchmark, includes 31,500 images (45 classes, 700 images each), each paired with five captions; it emphasizes diverse urban and rural scene descriptions.

FIT-RS [24]: Introduced in SkySenseGPT, FIT-RS includes 1.8 million instruction-image pairs across multiple fine-grained tasks, from captioning and VQA to relation reasoning and scene graph generation, supporting advanced retrieval and relational understanding.

STAR [17]: Provides structured scene graphs with object-level annotations and spatial relations, enabling reasoning-based retrieval in satellite imagery.

Table 1: Comparison of Remote Sensing Image–Caption Datasets

Dataset	**Train**	**Val**	**Test**	**Captions per Image**	**Resolution**	**Primary Tasks**
FIT-RS [24]	4,743	1,185	1,069	1	512	Retrieval, SGG
NWPU-Captions [4]	25,199	3,150	3,150	4	256	Captioning, Retrieval
RSICD [23]	8,734	1,094	1,093	5	256	Captioning, Retrieval
RSITMD [30]	3,433	858	452	5	256	Matching, Retrieval
Sydney-Captions [30]	497	58	58	5	500	Captioning, Retrieval
UCM-Captions [31]	1,680	210	210	5	256	Captioning, Retrieval
STAR [17]	771	264	238	—	512 ~31,096	SGG, RR

Val: validation split; SGG: Scene Graph Generation; RR: Relation Reasoning

3.2 Image Graph Datasets Overview

Since our focus is on retrieval with detailed captions, and such detail can be represented through scene graphs, it is of interest to consider image–graph datasets. We highlight two examples: **STAR** [17], offering rich scene graph annotations without captions, and **FIT-RS** [24], derived from STAR with added natural language descriptions.

STAR Dataset [17]: This dataset enables detailed scene understanding with 48 object categories, 11 scene types, and 58 fine-grained relationships grouped into 6 relation categories, including spatial, functional, motion-related, and circuit-based interactions. Annotation rules were defined by 6 remote sensing experts, with 9 trained professionals annotating each image, followed by expert review. However, since the experts focused on specific scenarios, the generated scene graphs do not fully capture the complexity of the entire scene and objects. Fine-grained captions could be generated from these scene graphs to support retrieval, but this has already been done in the FIT-RS dataset.

FIT-RS Dataset [24]: The FIT-RS dataset is derived from STAR by generating captions from scene graphs to support retrieval tasks. Large-resolution remote sensing images are segmented into overlapping 512×512 patches using a sliding-window strategy. Region-level triplets are extracted and paired with background descriptions generated by TinyLLaVA-3.1B. These elements are then merged into coherent, natural language captions using GPT-4, resulting in an image–caption dataset suitable for fine-grained retrieval-based applications.

3.3 Image Caption Datasets Analysis

Although these datasets are widely adopted, we argue that they are not well-suited for fine-grained retrieval tasks. To better understand the root of this issue, we identify two major limitations common to most existing datasets:

- **Visual Diversity:** Many datasets contain only a few scene categories (e.g., airport, industrial area), and within each category, visual variation is often limited.
- **Textual Diversity:** Captions are often generic and repetitive, and in some cases, a single caption could describe multiple images, for example, "several boats in a port".

These weaknesses can lead to a scenario where a single caption could plausibly describe multiple distinct images, or an image could be described by various captions, as illustrated in Figure 1.

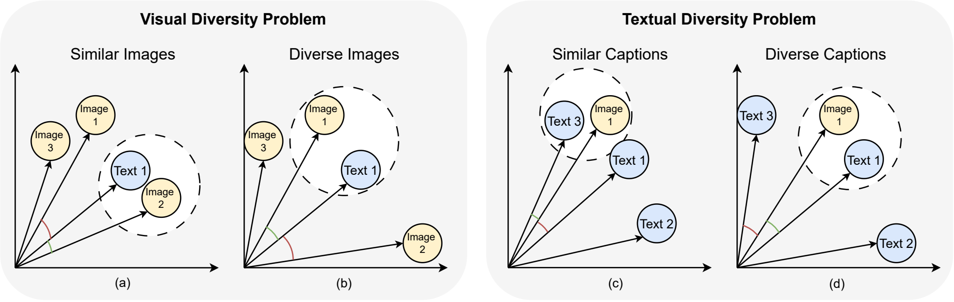


Figure 1: Illustration of visual and textual diversity issues.

Figure 1 highlights two key problems. (a) and (b) depict visual diversity: in (a), similar images cluster together, making one caption describe many images; in (b), diverse images remove the confusion, making the text 1 match the image 1 correctly. (c) and (d) depict textual diversity: in (c), similar captions could describe different images; in (d), diverse captions remove ambiguity. These issues reduce dataset specificity and descriptive precision.

3.3.1 Visual Diversity Evaluation for Retrieval Tasks

We run multiple analyses of visual content only, as follows:

Image Similarity Matrix (ISM): We encode all images using a CLIP-based image encoder and compute pairwise cosine similarities. A high concentration of high similarity scores within the matrix indicates redundant visual content, thus signaling poor visual diversity. Figure 2 illustrates the similarity matrix for the 6 datasets, using the RemoteCLIP_ViT-L-14 model as visual encoder.

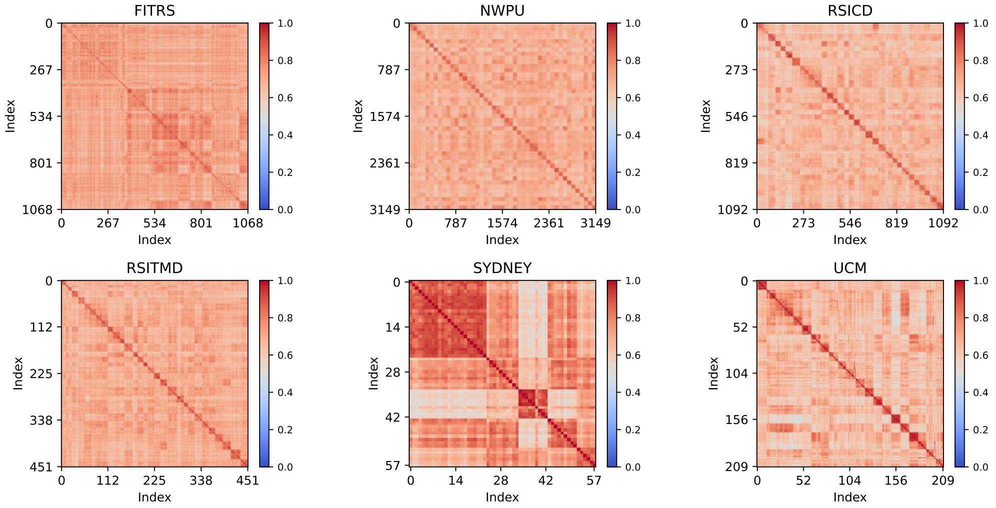


Figure 2: Image-to-image similarity matrix

It is important to interpret these matrices cautiously, particularly regarding dataset size. The goal is not to compare datasets, but to assess whether a dataset is visually diverse. This can be done by analyzing the contrast between diagonal and off-diagonal values. We notice significant off-diagonal high scores, causing cases where one caption describes multiple images (Figure 1 (a) and (b)).

Histogram of Similarity Scores: We extract all upper-triangular pairwise similarities from the similarity matrix and compute a histogram. The similarity scores are normalized and plotted per dataset. The x-axis represents the similarity score, and the y-axis shows the percentage of image pairs falling into each bin. Figure 3 visualizes these distributions across six datasets.

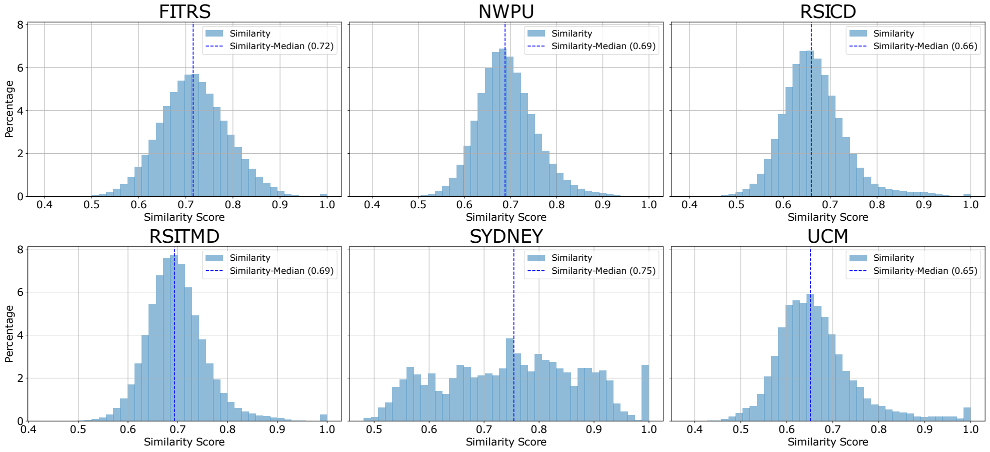


Figure 3: Pairwise Image-to-image similarity score distribution

SYDNEY shows the highest median similarity (0.75), indicating low visual diversity. RSICD and UCM suggest more diverse content, while FITRS and RSITMD exhibit moderate redundancy. The distribution appears roughly normal across most datasets, except for SYDNEY, due to its low image diversity (contains only images of Sydney city), and is much smaller compared to the others.

Scene Distribution Analysis: We analyze the distribution of annotated scene categories across datasets to understand class imbalance and diversity. Table 2 summarizes key statistics, including the number of scene types and image frequency ranges. We focus our analysis on datasets with sufficient image coverage per scene, such as RSICD, RSITMD, and NWPU. In contrast, datasets like SYDNEY and UCM contain fewer images overall, and their scene labels are either absent or not explicitly defined, making it harder to infer or classify scenes automatically. From Table 2, we can see that RSICD and RSITMD exhibit noticeable class imbalance. In contrast, NWPU shows a more balanced distribution, with high and consistent image counts per scene.

Table 2: Scene distribution statistics across datasets

Dataset	**# Scenes**	**Max Images / Scene**	**Min Images / Scene**	**Median Images / Scene**	**Mean Images / Scene**
RSICD	30	420	220	345	333.33
RSITMD	33	251	6	154	143.73
NWPU	44	700	700	700	700

3.3.2 Textual Diversity Evaluation Metrics for Retrieval Tasks

Captions in many RS datasets tend to be repetitive, generic, and structurally similar. Captions such as *"Many airplanes at the airport"* frequently appear across samples. In RS imagery, object presence and spatial relationships are often dictated by scene context (e.g., airports typically contain airplanes, taxiways, and terminals). As a result, captions must be highly specific and detailed, rather than offering vague descriptions. Generic captions like *"Many airplanes in an airport"* are not suitable for fine-grained retrieval tasks. We provide a brief analysis of the textual content of the datasets below.

Caption Similarity Matrix (CSM): We embed all captions using a pre-trained BERT model [6]. These embeddings are then used to compute pairwise cosine similarity scores between captions. The resulting matrix, Figure 4, visually represents the redundancy.

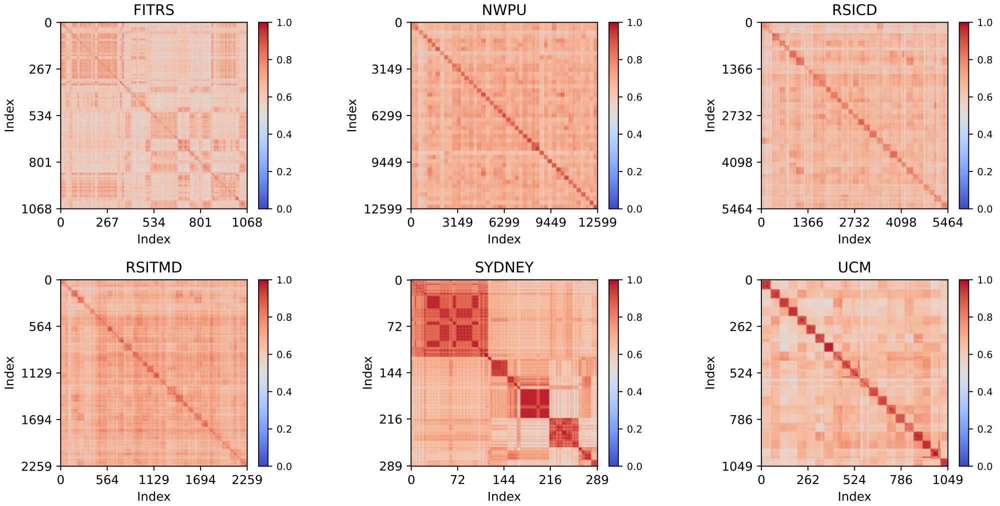


Figure 4: Text-to-text similarity matrix

As seen in the CSM, we observe the same issue here: many cells (except the diagonal) are heavily colored, indicating high similarity between captions. This suggests that, similar to the image representations, a significant portion of the dataset contains highly repetitive or generic descriptions. Many text pairs are highly similar, which can lead to cases where a single image could be described by multiple texts, as described in Figure 1 (c) and (d).

Histogram of Similarity Scores: We apply the same procedure as with visual similarity. Figure 5 shows the distributions across datasets. Sydney shows many high-similarity scores, indicating less diverse captions, likely due to having only images of Sydney city. This isn't a major issue, as we intend to keep the original image. In other datasets, high similarity scores are less prominent, and the distributions are more balanced.

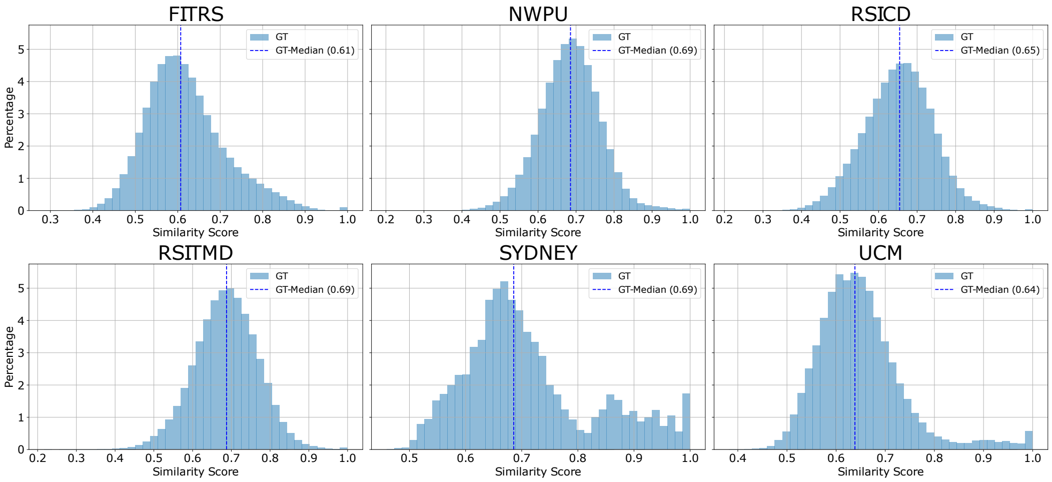


Figure 5: Pairwise Text-to-Text similarity score distribution

Caption Length Distribution: Analyzing the number of words per caption offers insights into verbosity or conciseness across datasets. Shorter captions often reduce semantic diversity while longer captions provide richer, more descriptive content. The caption length distribution statistics are provided in Table 3.

Table 3: Caption Length Statistics Across Datasets

Metric	**FITRS**	**NWPU**	**RSICD**	**RSITMD**	**SYDNEY**	**UCM**
Max	472	51	35	35	21	23
Mean	91.88	11.58	11.54	10.37	13.20	11.51
Median	82.0	11.0	11.0	10.0	13.0	11.0

Except for FITRS, which contains significantly longer and more informative captions, most datasets feature concise descriptions, typically under 30 words. This brevity may lead to higher semantic overlap, reduced diversity, and increased redundancy in caption content.

Together, these metrics enable a more rigorous assessment of whether a dataset offers the necessary variability and grounding to support robust retrieval. Datasets with limited visual or textual diversity can produce high retrieval scores that reflect memorization rather than genuine semantic alignment.

3.4 Fine-Grained Captions generation

To address the limitations identified in current RS datasets, particularly the lack of caption diversity, we propose creating new fine-grained datasets derived from the original datasets. Specifically, we will retain the same images but generate new, more detailed, and informative captions. The goal is to enrich the datasets with captions that are more descriptive, scene-specific, and less repetitive compared to the originals. We will detail the caption generation process and then compare the new datasets with the original ones.

3.4.1 Caption Generation via VLM

We use a state-of-the-art VLM (Gemini-2.5-flash) for caption generation. Generally, better VLMs produce higher-quality captions; however, Gemini-2.5-pro is very expensive and time-consuming to run on the entire datasets. Therefore, we opted for the faster flash version.

To generate new fine-grained datasets, we proceeded as follows: For each image, we generated a caption using the structured prompt detailed in Appendix A.1.1. This prompt was designed to encourage the model to produce fine-grained, contextually rich descriptions. We explicitly guided the model to focus on objects and spatial and semantic relationships to obtain detailed, diverse captions that are more informative than those in the original datasets.

3.4.2 Comparison with the Original Datasets

We evaluate and compare the original and augmented versions of the datasets using the metrics introduced in Section 3.3.2: similarity matrix and score distribution, and caption length statistics.

Caption Similarity Matrix (CSM) We evaluate the pairwise similarity scores between all possible caption pairs, as illustrated in Figure 6.

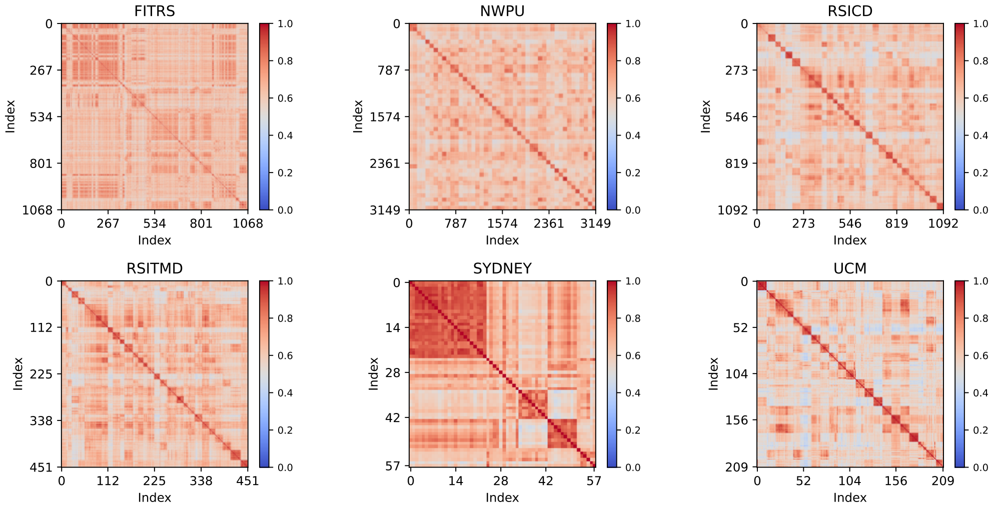


Figure 6: Text-to-text similarity matrix

In Figure 6, we still observe notable off-diagonal activations, but they are less pronounced compared to the original datasets. There is a stronger contrast between diagonal and off-diagonal values, with more white-to-blue regions, indicating that the generated captions are increasingly diverse. This suggests that the fine-grained captions reduce redundancy and enhance semantic variation between text pairs.

Similarity Scores distribution Comparison

Figure 7 shows the distribution of similarity scores for original captions (blue histograms) and fine-grained captions (orange histograms). The distributions for the fine-grained captions are generally shifted to the left and are wider, signifying a more varied range of descriptive content. This difference indicates that the fine-grained captions are more diverse and less repetitive than the original human-written captions. The only exception is FIT-RS, which does not follow this pattern, as its captions are fine-grained, generated from the ground-truth scene graphs of the images rather than written by humans.

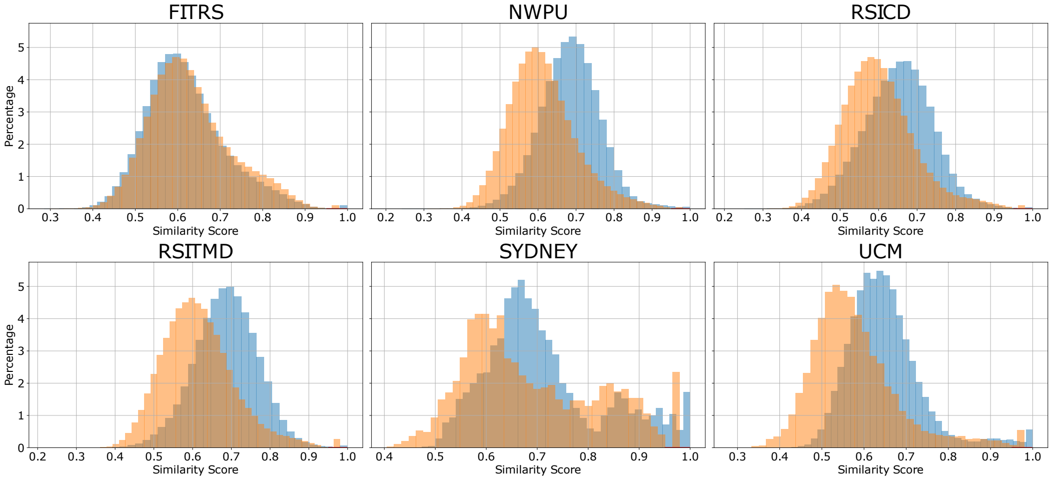


Figure 7: Text-to-Text similarity distribution comparison 🟦 Original data 🟧 Fine-grained data

Caption Length Distribution Comparison The caption length statistics across different datasets are presented in Table 4. By comparing the original datasets values with the fine-grained datasets values, we can observe how the fine-grained captions show a shift towards longer captions, as opposed to the skewed patterns of the original captions.

Table 4: Caption Length Statistics Across Datasets (Original vs. Fine-Grained)

Dataset	Original				Fine-Grained			
	Min	Max	Median	Mean	Min	Max	Median	Mean
FITRS	10	472	82.0	91.88	30	214	81.0	84.75
NWPU	5	51	11.0	11.58	32	198	89.0	91.57
RSICD	4	35	11.0	11.54	40	238	101.0	105.10
RSITMD	4	35	10.0	10.37	42	229	97.0	100.66
SYDNEY	6	21	13.0	13.20	38	230	84.0	87.40
UCM	4	23	11.0	11.51	37	193	84.0	86.63

Overall, these comparisons indicate that the new fine-grained datasets appears to be more diverse and therefore has potential for retrieval tasks. It is important to highlight that such diversity is necessary for retrieval; otherwise, models may achieve high scores by overfitting minor details, like punctuation or sentence length, factors that are not meaningful to humans and reduce the system's reliability in real-world scenarios. Even though the fine-grained data is generated by a VLM, it appears more compelling and better aligned with real-world implementation. This will be further validated by human evaluation in the following sections.

4 Multimodal Representations

4.1 Motivation

In real-world RS applications such as risk assessment, the objective goes beyond scene classification. It requires identifying fine-grained, localized information, such as object presence, arrangement, and spatial and semantic relationships that typical global embeddings fail to capture. While a CLIP-based embedding might successfully classify a scene, it often misses structural details.

RS Imagery Paradox: RS images may appear straightforward due to the domain-specific context of the scene. For instance, airports typically contain runways, terminals, and airplanes, rather than stadiums or boats. Objects can often be inferred from the scene context. However, in practice, these scenes are highly complex. They frequently exhibit high object density, fine spatial resolution, and scale variations, challenges that are less common in natural image datasets.

Visual Representation Limits: Satellite images often cover large areas with dense arrangements of small objects, such as vehicles or buildings. A single CLIP embedding lacks the capacity to represent fine-grained details such as the number of airplanes, their spatial arrangement, or proximity to structures such as hangars. These relational and spatial attributes are essential for downstream tasks like insurance risk evaluation. Another limitation arises from the preprocessing step of resizing images to fit the fixed input size expected by the CLIP ViT backbone. RS images lose valuable detail when resized because objects of interest can become too small, and spatial relationships are distorted.

Text Representation Limits: Textual descriptions of RS scenes are often long, dense, and multitopic, describing various regions and features within a single caption. Collapsing such descriptions into a single embedding flattens spatial cues, object references, and contextual groupings. This weakens alignment with image regions. Structured or modular text representations, aware of spatial and object-level context, are better suited for multimodal alignment in RS.

4.2 Visual Representations

4.2.1 Image to Graph Representations

We propose representing each image as a **scene graph (SG)**. This structure provides an abstract representation that is human-understandable. In the context of risk assessment, such a representation enables detailed reasoning, object relationships, and fine-grained information retrieval.

Initial SG: We start with a basic SG design, where nodes represent objects and edges represent the relationships between them. Each object instance is listed separately (e.g., car 0, car 1, etc.). This simple structure provides a foundation but proves insufficient for capturing the full image content.

Challenges with the Initial SG: The basic SG cannot capture key visual details, and several issues arise when applying it to real-world images. These challenges are outlined below:

- **Graph Collision:** The initial SG's simplicity causes distinct images (e.g., different airports) to yield identical scene graphs, losing uniqueness. For example, two airport scenes with different features may be represented with the same SG.
- **Subgraph Matching and Object Count Issues:** A major issue arises when comparing object counts. If an image contains 10 airplanes, but the query mentions "2 airplanes" or "5 airplanes," the SG returns similar scores, despite the object count differences.
- **Uncountable Objects and Scene Segmentation:** Certain elements, like rivers, water bodies, and vegetation, are uncountable or not considered objects. The initial SG struggles to handle such cases, losing important semantic distinctions.

Augmented SG: The initial SG was too simplistic to handle the complexity inherent in real-world scenes. As a result, we recognized the need for a more robust structure to address these challenges. Consequently, we propose different augmentation techniques as follows:

- **Node Augmentation:** Nodes can represent multiple aspects of the scene, including:
    1. **Visible objects:** Examples include buildings, playgrounds, and cars.
    2. **Semantic regions:** Such as rivers, forests, and other land cover types.
    3. **Attribute values:** Properties of objects, such as color, shape, or size, and counts.
- **Edges:** Edges encode three main types of relationships:
    1. **Spatial relations:** These describe the physical placement of objects in relation to each other, such as "adjacent to," "surrounded by," "on," "inside," and "connected to."
    2. **Attribute types:** Specify the category of the attribute that links an object node to its corresponding attribute value node, such as "color," "size," or "count".
    3. **Functional relations:** These express the purpose or interaction between nodes, such as "used for," "transports," or "supports,". For example, a crane "lifts" a container, or a road "connects" two buildings.
- **Object Count:** If the objects are countable and clearly visible, we create a single node with the count as an attribute connected to it.
- **Global Scene Context:** To summarize the overarching context of the scene, we introduce a **hyper-node**, which captures the global theme of the scene. For instance, the hyper-node could represent categories like "Urban Area," "Airport," or "Port."
- **Constraints:** To ensure consistency and accuracy in the representation, the following constraints are applied:
    - Every object node must appear in at least one spatial and one attribute relation.
    - Vague relations, such as "near," are avoided. Instead, we use accurate and specific terms to describe the relationships between objects.

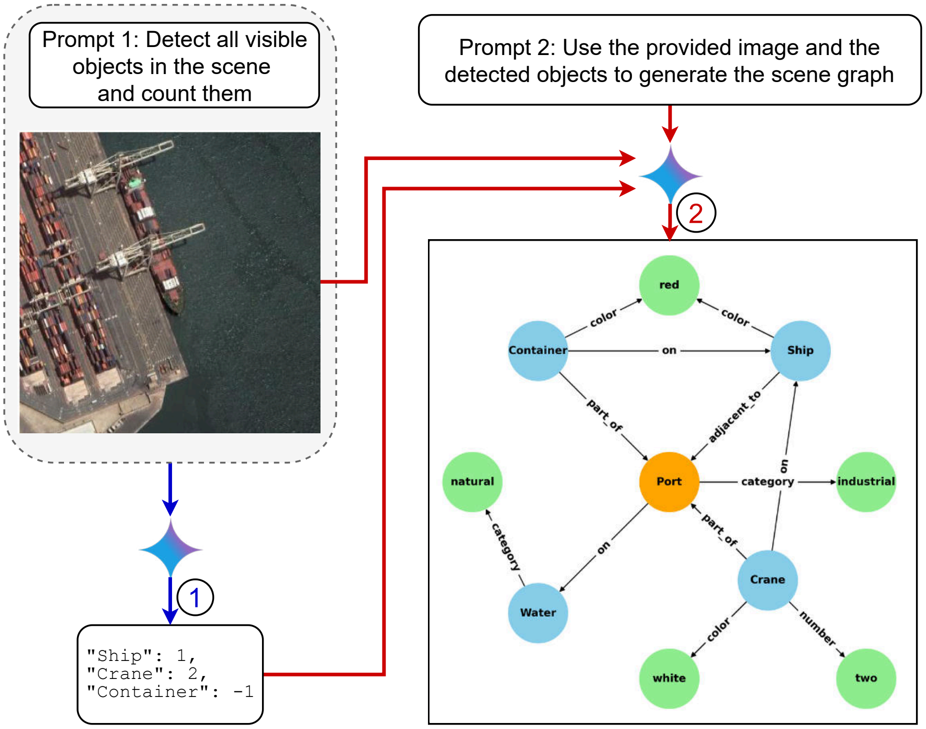
SG generation pipeline Unlike traditional object detection pipelines, this approach captures both tangible objects and abstract semantic regions, allowing for richer scene understanding. Currently, no dedicated SG generation models exist for remote sensing imagery. However, we can leverage existing models for object detection and semantic segmentation, and use a VLM to infer relationships between detected entities. This task is inherently complex, but we simplify it using a VLM such as **Gemini-2.5-flash**, which can generate a scene graph based on visual and contextual information.

Prompt strategies are essential for extracting detailed and accurate scene graphs from VLMs. To enhance coherence and structure, we apply **chain-of-thought (CoT)** [42] prompting. The process consists of the following steps:

1. **Segmentation and Object Detection and Counting** : Detect all relevant objects and segment semantic regions in the image. For countable objects, record the count; for uncountable entities (e.g., rivers, forests), assign a value of -1.
2. **Scene Graph Construction**: Use the detected objects, their counts, and the image context to construct a scene graph. Extract attributes for each object (e.g., type, color, spatial extent), and define relationships (e.g., “next to,” “surrounded by”). We emphasize that relationships should be specific and meaningful, avoiding vague labels like “related to.”

This structured approach is expected to enhance representation quality, offering richer and more complete scene graphs with a greater number of nodes and attributes that better reflect the complexity of remote sensing scenes.

Figure 8 illustrates the complete pipeline. In the resulting scene graph, nodes represent objects or semantic regions. The orange node denotes the central scene (e.g., the port), while blue nodes represent detected objects such as containers, ships, and cranes, along with other contextually relevant entities not directly detected in step 1. Green nodes indicate object attributes, such as color, material, or number. For further details on the prompts used, please refer to the Appendix A.1.2.



Prompt 1: Detect all visible objects in the scene and count them

Prompt 2: Use the provided image and the detected objects to generate the scene graph

```
"Ship": 1,
"Crane": 2,
"Container": -1
```

Figure 8: Pipeline for Scene Graph Generation from RS Imagery. **Step 1 (Blue arrows)**: We prompt Gemini to detect and count all visible objects in the image. **Step 2 (Red arrows)**: Using the detected objects, their counts, and the original image, we prompt Gemini again to generate a structured scene graph that includes both objects and their relationships.

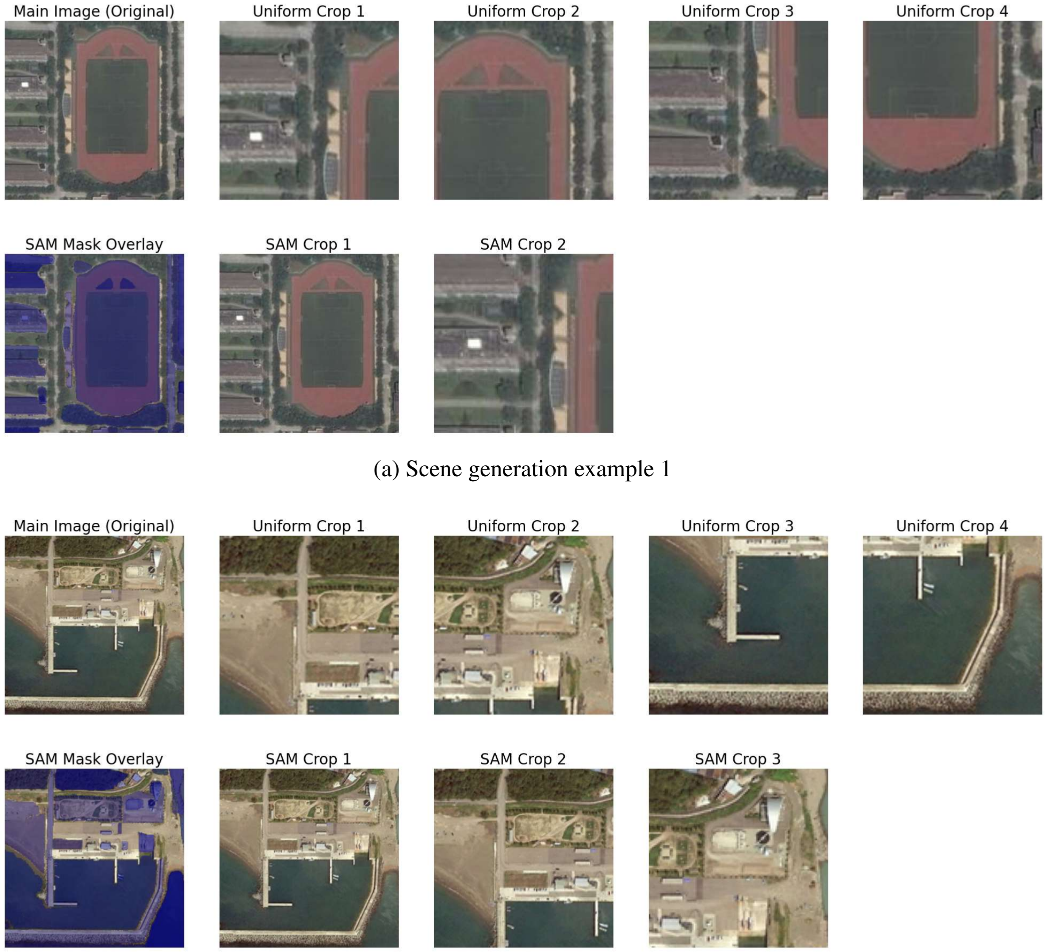
4.2.2 Image to Scene Representations

While SG construction captures rich visual relationships, it can be computationally intensive and may miss subtle but important details. As a cost-effective alternative, we propose a **region-based decomposition approach** that is significantly easier to implement. Rather than constructing a full scene graph, this method focuses on cropping the image into multiple patches or regions of interest. The image is then represented as several overlapping scenes derived from the main image, rather than treating the image as a single whole. This approach provides a scalable and flexible middle ground between simple vector representations and dense scene graphs.

Uniform Cropping with Overlap (UC) In this approach, the image is divided into a uniform grid of fixed-size patches with overlapping margins. The overlap ensures continuity between adjacent patches, preserving spatial context. Each patch is then resized to match the input resolution required by the CLIP encoder and passed through the model independently. This technique is suitable for high-resolution RS imagery, where maintaining local detail is crucial for accurate interpretation.

SAM-Based Cropping with Bounding Box Aggregation (SC) This method leverages the Segment Anything Model SAM 2 [32] to perform content-aware segmentation of images. Rather than using fixed-size patches, the model identifies and segments visually meaningful regions, such as objects, land cover types, or structural components. For each segment, a bounding box is computed to spatially localize the region. These bounding boxes are then grouped and clustered based on spatial proximity to form larger, aggregated regions that capture semantically rich content. The resulting composite bounding boxes are used to extract representative crops from the image, reflecting the core visual elements of the scene. This approach offers greater semantic precision than uniform cropping, as it focuses on actual content rather than arbitrary spatial divisions.

Scene representation examples Figure 9 compares uniform cropping with SAM-based segmentation. In the first row, we show the original image and the uniform crops, which divide the image into equal patches without considering content. In contrast, the second row presents SAM-generated crops, which adaptively focus on semantically meaningful regions. This is especially evident in Example 1, where uniform cropping splits a large object into arbitrary segments, while SAM preserves it as a whole. Similarly, in Example 2, SAM captures distinct, related features more effectively. Overall, SAM's content-aware approach enables more coherent scene representations than uniform cropping.



(a) Scene generation example 1

(b) Scene generation example 2

Figure 9: Comparison of SAM-Based Segmentation and Uniform Cropping

4.3 Textual Representations

4.3.1 Text to Graph representation

Transforming descriptive text into scene graphs has become more flexible with the use of large language models (LLMs). Earlier methods relied on rule-based systems or triplet extraction pipelines. Recent research, such as [44], [9], and [21], now leverages LLMs to extract nuanced relationships and handle longer contexts. Although this approach requires slightly more computational resources, it produces richer graph structures by identifying entities, attributes, and complex logical or spatial relations. The system prompt is provided in Appendix A.1.3.

- **Nodes** represent entities like physical objects (e.g building, airplane), semantic regions (e.g forest), abstract concepts (e.g. textit residential area), or attributes (type, size, or color).
- **Edges** define relationships such as spatial proximity (*next to*, *surrounded by*), structural links (*contains*, *part of*), or interactions (*connected to*, *flows through*).

Figure 10 illustrates the graph generated from the human caption. This graph-based structure enables a human-interpretable representation of textual scene descriptions.

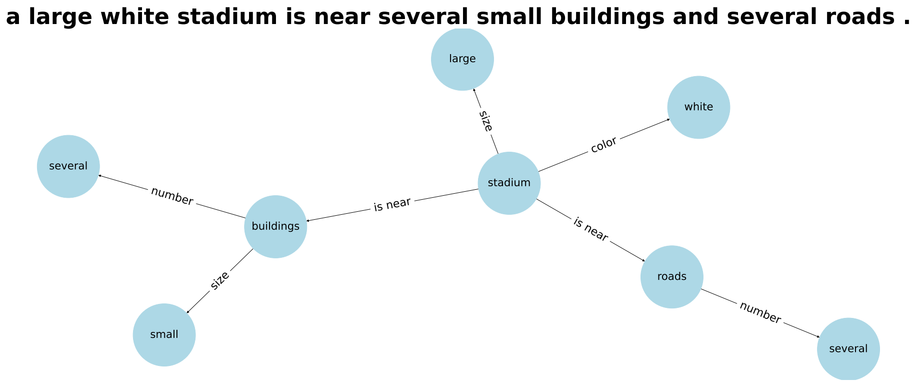
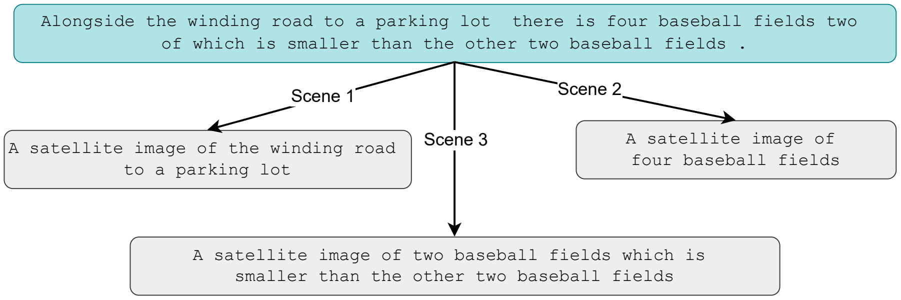

a large white stadium is near several small buildings and several roads .

Figure 10: Graph-based representation of an image caption.

4.3.2 Text to Scene

We propose a lightweight alternative: decomposing long descriptive captions into coherent scenes. LLM is used to segment the original text into short, logically grouped sentences, each corresponding to a distinct spatial or thematic region in the image as illustrated in Figure 11. This allows for more localized and interpretable scene understanding while maintaining alignment with the original text. The system prompt is provided in Appendix A.1.4.


Alongside the winding road to a parking lot there is four baseball fields two of which is smaller than the other two baseball fields .

Scene 1

A satellite image of the winding road to a parking lot

Scene 2

A satellite image of four baseball fields

Scene 3

A satellite image of two baseball fields which is smaller than the other two baseball fields

Figure 11: Decomposition of a textual description into multiple coherent sub-scenes using LLM.

5 Retrieval Framework

In our framework, we present the different matching modes across multiple representation modalities for both images and text, as detailed in Section 4. We describe various cross-modal and hybrid matching techniques and explain the embedding process for each modality. We also detail the similarity modules used to compare embeddings and the overall training pipeline.

5.1 Cross-Modality Retrieval Modes

To support a broad range of alignment strategies between images and text, we explore various cross-modal matching modes. These modes vary in the level of granularity and the structural representation used for each modality, as illustrated in Figure 12.

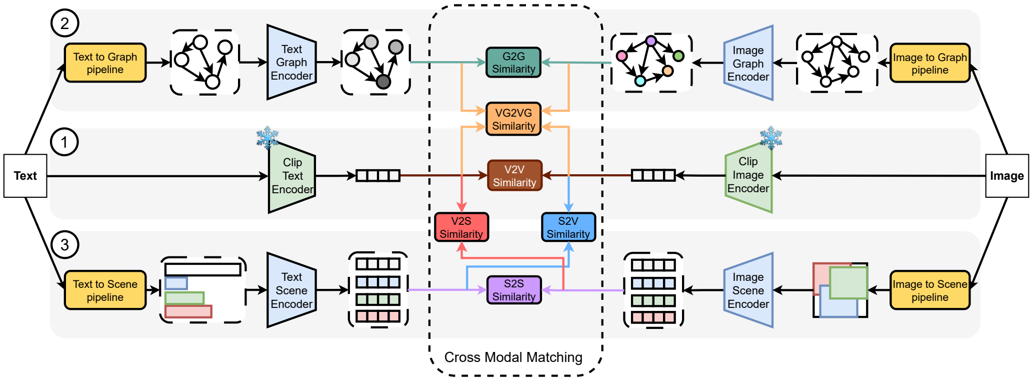


Figure 12: Overview of Text-to-Image Retrieval Framework

① **Vector-to-Vector (V2V) Matching** In this baseline, image and text inputs are encoded into global vectors using CLIP encoders. Similarity is computed via cosine similarity between these embeddings.

② **Graph-Based Matching** These methods involve SG construction (Sections 4.2.1, 4.3.1):

- **Graph-to-Graph (G2G) Matching:** Images and texts are represented as graphs: scene graphs for images and semantic graphs for texts. Each graph is encoded into a graph embedding. Then, a similarity function compares the embeddings to produce the final matching score.
- **VectorGraph-to-VectorGraph (VG2VG) Matching:** For each image and text input, both a global vector embedding and a graph embedding are computed. The final similarity score is derived by fusing the similarities from the vector and graph components.

③ **Scene-Based Matching** These methods involve the construction of scenes (Sections 4.2.2, 4.3.2):

- **Scene-to-Scene (S2S) Matching:** Both image and text inputs are segmented into multiple scenes. Each scene is independently encoded using CLIP-style encoders, resulting in a set of scene embeddings for each modality. Pairwise similarities are computed between all scene pairs, and a specialized module aggregates these similarities into a global score.
- **Vector-to-Scene (V2S) Matching:** The text input is encoded as a single global vector, while the image is decomposed into multiple scene embeddings. The global text vector is compared against each image scene embedding individually, and a specific module is used to aggregate these comparisons into the final score.
- **Scene-to-Vector (S2V) Matching:** The image is encoded as a single global vector, while the text is segmented into multiple scenes, each encoded separately. Each textual scene embedding is compared to the image vector, and a module is applied to derive the final score.

Table 5 summarizes the strengths and limitations of each cross-modal matching mode. It offers a detailed comparison to highlight the trade-offs and suitability for different retrieval scenarios. This overview helps identify promising directions based on task requirements.

Table 5: Advantages and Disadvantages of Cross-Modal Retrieval Matching Modes

Matching Mode	**Advantages**	**Disadvantages**
V2V	Simple and easy to implement Computationally efficient.	Misses fine-grained information. Lacks structural and relational context.
G2G	Fine-grained, relational alignment. Captures explicit object attributes Captures explicit relationship. It can be grounded with many descriptions.	Sensitive to the quality of the graphs. RS graph generation is hard. Ignores direct use of visual embeddings.
VG2VG	Combines global context and fine-grained details. Robust to noisy modality components.	RS graph generation is hard. Balancing between modalities and similarity functions is challenging.
S2S	Region-level matching: Mid-point between global and graph representation. No graph construction needed.	Sensitive to image segmentation. Text scene generation can explode: many scene combinations for longer captions.
V2S	Enables localized comparison across various image regions. Ideal for captions that focus on specific visual elements.	Text is compressed into a single vector. Poor handling of complex or compound sentences.
S2V	Handles rich and long captions. Captures detailed semantics from text.	Ignores image spatial structure. Image is compressed into a single vector. Sensitive to the text segmentation.

From Table 5, we observe that graph-based approaches (e.g., G2G) are particularly promising due to their ability to capture fine-grained relational alignment between modalities. However, their performance is highly sensitive to graph generation quality. While generating graphs for text is easy using LLMs that intelligently process text, generating scene graphs from images remains challenging, especially for RS applications. The hybrid VG2VG approach stands out as a strong candidate, combining the strengths of both global and structured representations. We expect this method to perform well, especially in scenarios where robustness to missing components is crucial.

Another promising direction is scene-based matching (S2S, V2V, and S2V), which avoids explicit graph generation while still enabling region-level alignment. While S2V may underperform, as it compresses the image into a single vector like CLIP models, V2S and S2S appear to be interesting directions for further exploration. They allow for localized, region-based interactions and offer a good trade-off between complexity and expressiveness, especially when graph construction is undesirable or infeasible.

5.2 Embedding models

In this section, we describe how embeddings are produced for each modality (image and text) within our framework. These embeddings serve as the basis for computing similarity scores across different matching strategies.

5.2.1 Vector Embedding models

In our retrieval framework, we do not fine-tune the CLIP model for full image and text embeddings, as existing methods (e.g., RS-M-CLIP, Remote-CLIP, GeoRSCLIP) have already addressed this. Instead, we focus on leveraging these models as a foundational baseline.

5.2.2 Graph-Based Embedding Models

We consider two variants of graph-based embedding models: a vanilla version and a trainable version.

Vanilla Embedding Model After constructing SG, where nodes and edges are described textually, we pass each node and edge description through a BERT model to obtain embeddings. These embeddings form the graph representation without training, as illustrated in Figure 13.

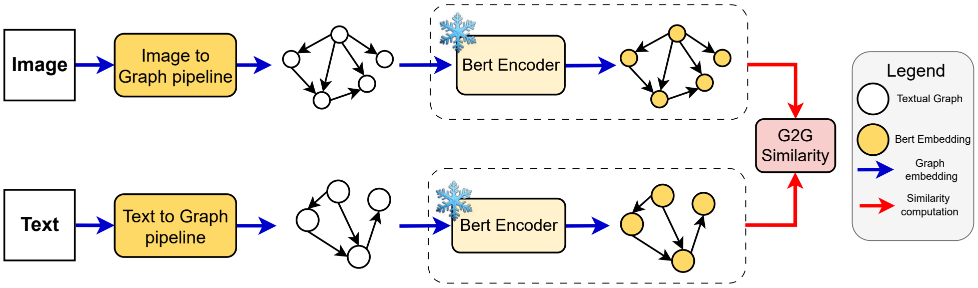


Figure 13: Vanilla Graph-based Retriever Architecture

Trainable Embedding Model Building on the vanilla model, we introduce trainable graph neural network (GMN) architectures that can learn more expressive graph embeddings. Initially, a vanilla embedding is computed, then passed through a trainable GNN module to produce a refined, task-adaptive graph representation. Figure 14 describes the full pipeline.

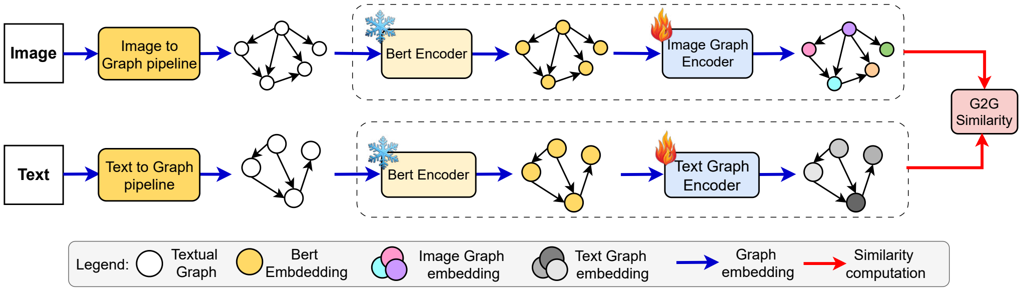


Figure 14: Trainable Graph-based Retriever Architecture

Given that our graphs include rich edge attributes, we emphasize GNN models that are capable of leveraging both node features and edge information effectively:

- **Graph Attention Network (GAT)** [39]: Leverages multi-head masked self-attention over each node's neighborhood to assign learnable weights to neighbors. It performs efficient, inductive message passing without costly matrix operations.
- **Graph Transformer Architectures** [35]: extend Transformer layers to graph structures using multi-head dot-product masked attention mechanisms. In this context, the mask is derived from the adjacency matrix, which helps incorporate the structural information of the graph, including edge attributes.

Vector-Graph Embedding Model We propose a hybrid model that combines vector-based and graph-based embeddings. This approach leverages complementary strengths: semantic representations from pre-trained models and structural patterns from graph-based reasoning. As illustrated in Figure 17, the architecture includes a trainable graph embedding module and integrates a parallel vector embedding module. Each image/text input is transformed into both a vector and a graph embedding that are fed into a dedicated similarity module, which produces a single matching score.

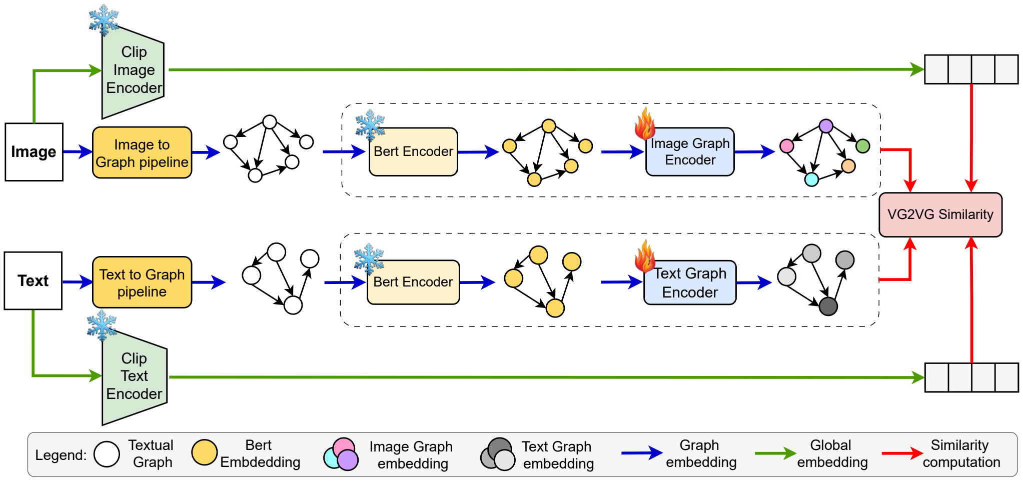


Figure 15: Hybrid Trainable Vector-Graph Retriever Architecture

5.2.3 Scene Embedding model

For scene-based matching, we believe that the CLIP-based models (e.g., RS-M-CLIP, Remote-CLIP, GeoRSCLIP) is sufficiently powerful. Our work focuses on enhancing the way these models are utilized. We divide both the image and the text into distinct scenes and embed each scene, as illustrated in Figure 16.

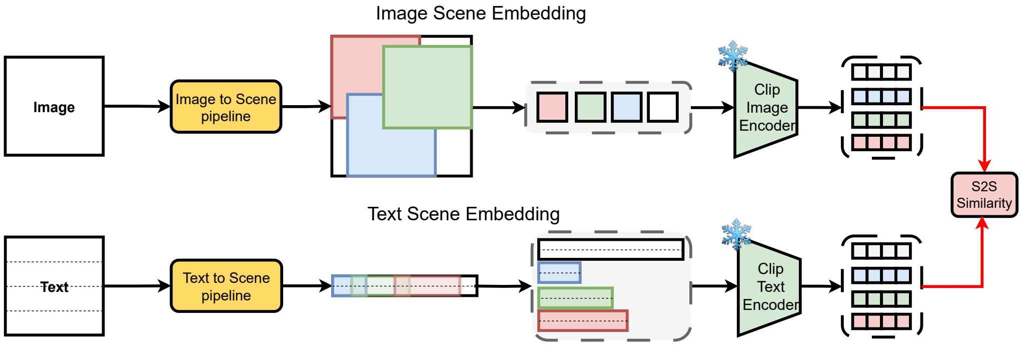


Figure 16: Scene embedding pipeline

For image embeddings, we proceed as follows: we use the scene generation pipeline described in Sections 4.2.2 and 4.3.2 , and then apply the CLIP-based model to each scene segment. As a result, each image is represented by a variable-length set of visual embeddings. Note that the full image is also treated as a *meta-scene*, and its embedding is included in the final set of embeddings. We apply the same process to text by segmenting it into scenes and generate embeddings for each segment.

5.3 Similarity Modules

To assess the alignment between textual queries and image candidates, we use several similarity modules depending on the cross-modal matching mode. Let E_q be the query embedding for query, and E_d be the image document embedding.

5.3.1 Vector Matching

For full image/text embeddings, we compute the cosine similarity directly between the vector representations of the image and the caption as described in Equation (1). This method is simple and computationally efficient, making it suitable for basic text-to-image matching tasks.

$$\text{sim}_{\text{vec}}(E_q, E_d) = \frac{E_q \cdot E_d}{\|E_q\| \|E_d\|} \tag{1}$$

5.3.2 Graph Matching

For structured graph representations, we pose the similarity task as an assignment problem: finding the optimal one-to-one alignment between nodes in the query (text) and document (image) graphs.

Problem Setup: Let G_q and G_d represent the query and document graphs, respectively. Node is represented by a vector ($E_q(i)$ for query node i and $E_d(j)$ for document node j). The query and document graphs may differ in the number of nodes and edges. We assume a hard one-to-one matching. Since graph sizes differ, we match only $n = \min(|V_q|, |V_d|)$ nodes for a partial bijection.

Matching Algorithm and Limitations:

1. **Pairwise Similarity:** We compute a cosine similarity matrix $S \in \mathbb{R}^{N_q \times N_d}$ between the nodes.
2. **Late Interaction via Optimal Node Assignment:** The Hungarian algorithm is applied to S to find the node permutation that maximizes the overall score under the one-to-one constraint.

This approach has limitations: it considers only node-level similarity and ignores edge structure. It also requires all nodes to be matched, which complicates object counting (Initial SG issues).Being aware of these limitations is important for SG construction.

5.3.3 Scene Matching

Scene matching similarity is used for scenarios where both queries and documents are represented as sets of unstructured embeddings, such as global scene descriptors or multi-scene image patches.

Problem Setup: Let the query and document be represented by sets of embeddings $E_q \in \mathbb{R}^{N_q \times D}$ and $E_d \in \mathbb{R}^{N_d \times D}$, respectively. Here, D is the shared embedding dimension, which is 768 for BERT. this approach does not impose hard one-to-one alignment, but flexible many-to-one alignments.

Matching Algorithm:

1. **Pairwise Similarity:** For query embeddings $E_q \in \mathbb{R}^{N_q \times D}$ and document embeddings $E_d \in \mathbb{R}^{N_d \times D}$, we compute the dot product between each $E_q(i)$ and all $E_d(j)$ to form a similarity matrix $S \in \mathbb{R}^{N_q \times N_d}$. Given a database with N documents, this results in N such matrices, one for each document.
2. **Late Interaction via Query-Wise Max-Aggregated Matching (Q-MAM):** As proposed by [7], for each query vector, retain the maximum score over all document vectors, then average these maxima across all query vectors, as explained in Equation (2).

$$\text{LI}(q, d) = \frac{1}{N_q} \sum_{i=1}^{N_q} \max_{j \in [1, N_d]} \langle E_q(i), E_d(j) \rangle \tag{2}$$

According to [7], this method significantly improves retrieval performance by focusing on the semantic coverage of the overall visual content. It also simplifies and accelerates the process.

5.3.4 Late Interaction via Graph Matching Network (GMN)

Since traditional graph matching methods rely on hard alignment algorithms like the Hungarian algorithm, which are inherently non-differentiable, they limit end-to-end learning and adaptation. To address this, we propose employing a graph matching network (GMN) that learns the graph matching process directly, enabling trainable similarity measures that can capture complex relationships beyond fixed heuristics. The model operates in **two main stages**: (A) **feature extraction** and (B) **scoring module**

(A) Feature extraction as illustrated in Figure 17, this module takes as input two graphs of arbitrary size and produces a **single fixed-dimensional vector v** summarizing their similarity as follows:

1. **Matrix Computation:** As before, we compute a pairwise cosine similarity matrix $S \in \mathbb{R}^{N_q \times N_d}$ between the query and document graph node embeddings.
2. **Histogram Encoding:** For each query node, we generate a histogram over its similarities to all document nodes. These histograms summarize the similarity distribution per node. Histograms have a fixed number of bins but are dynamically constructed based on the similarity values.
3. **Histogram Feature Extraction:** The resulting histograms are passed through an MLP (Multi-Layer Perceptron) to obtain a fixed-dimensional embedding h_i for each query node (3):
$$h_i = \text{MLP}(\text{Hist}(S_{i,1:N_d})) \tag{3}$$
4. **Attention and Projection:** A self-attention mechanism is applied using a learnable query vector $\mathbf{q}_{\text{att}}$ to compute attention scores α_i. The hidden representations h_i are aggregated and weighted by α_i, then projected through a multi-layer perceptron (MLP) to obtain a feature vector $\mathbf{v}$ according to (4).

$$\mathbf{v} = \text{MLP}(\sum_{i=1}^{N_q} \alpha_i h_i), \quad \alpha_i = \text{softmax}(q_{\text{att}}^\top h_i) \tag{4}$$

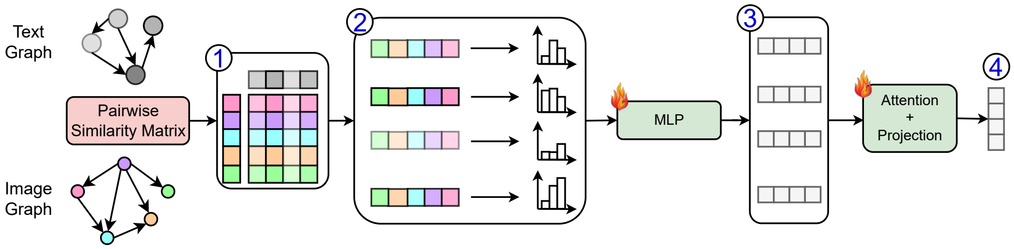


Figure 17: Feature extraction in Graph Matching Network

(B) Scoring Module: Depending on the matching type, the final score is given by (5):

$$\text{score} = \begin{cases} \text{MLP}\big([\mathbf{v}; sim_{\text{vec}}]\big), & \text{if VG2VG matching} \\ \text{MLP}(\mathbf{v}), & \text{otherwise} \end{cases} \tag{5}$$

Here, $[\mathbf{v}; sim_{\text{vec}}]$ denotes the concatenation of the feature vector $\mathbf{v}$ with the global score sim_{vec}.

This architecture integrates fine-grained node interactions and similarity score computation based on the varied-size pairwise similarity matrix as input.

5.4 Training pipeline

5.4.1 Training Objectives and Loss Functions

We focus on two widely used loss functions for training retrieval models: **InfoNCE loss** and **triplet loss**. Both are designed to promote high similarity between correctly matched image-text pairs while pushing apart unrelated pairs. These similarities are typically computed using *cosine similarity*, which produces scores in the range $[-1, 1]$, where 1 indicates perfect alignment and -1 implies complete opposition in direction.

The **InfoNCE loss [27]** is a form of contrastive learning that encourages the model to correctly identify positive (i.e., matching) pairs among a set of negatives within the same batch. It effectively maximizes the similarity between a query and its positive while minimizing its similarity with all other candidates. It is defined in Equation (6):

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(x_q, x_p)/\tau)}{\sum_{i=1}^{N} \exp(\text{sim}(x_q, x_i)/\tau)} \tag{6}$$

where $\text{sim}(\cdot, \cdot)$: cosine similarity; τ: temperature parameter controlling the sharpness of the distribution; N: total number of samples.

The **triplet loss** [34] explicitly models relative similarity by comparing a query to both a matching (positive) and a non-matching (negative) sample in the embedding space. Its main objective is to ensure the positive is more similar to the query than the negative by a margin δ. The loss function is given by Equation (7):

$$\mathcal{L}_{\text{triplet}} = \max(0, \text{sim}(x_q, x_n) - \text{sim}(x_q, x_p) + \delta), \quad (\delta > 0) \tag{7}$$

This ranking-based approach helps the model learn meaningful ordering in the similarity space, which is crucial for retrieval performance.

When selecting a loss function for contrastive learning, InfoNCE is often preferred in large-scale setups for its efficiency, scalability, and robustness. InfoNCE uses all other samples in the batch as negatives, rather than relying on explicit negative sampling strategies. This approach encourages the learning of discriminative representations and is computationally efficient, especially in large-batch scenarios. In contrast, triplet loss requires explicit negative sampling, which may cause unstable training if not handled carefully. Due to these challenges, we use InfoNCE loss in our setup.

5.4.2 Handling Batch Contamination in Multi-Caption Datasets

During training, for each batch, we have multiple image-caption pairs and construct the similarity matrix between all image-caption pairs. The objective is that the **diagonal elements**, which correspond to the positive pairs, should have the **maximum similarity**, while all off-diagonal elements correspond to the negative pairs, ensuring proper alignment.

However, in some datasets (e.g., RSICD, RSITMD, . . .), a single image is associated with **multiple captions** describing the same visual content. This occurs only in the original datasets and not in the fine-grained version, since for the latter we generate only one caption per image. This issue can lead to *batch contamination* during contrastive learning, particularly if two captions referring to the same image end up in the same training batch. In such cases, these captions will appear as separate instances in the similarity matrix. One of them will not appear on the diagonal of the similarity matrix. This undermines the assumption that the **diagonal elements** in the similarity matrix (i.e., the correct image-caption pairs) should have the **maximum similarity**, with **high contrast** to all off-diagonal (incorrect) pairs.

To address this issue, we implement two main solutions:

1. **Random Caption Selection per Epoch**: For each image, we randomly select *one caption per training epoch* from the set of available captions. This introduces *randomness into the training process*, ensuring that across epochs the model sees diverse text descriptions while avoiding multiple captions for the same image in a single batch.
2. **Biased Shuffling**: We perform a *controlled random shuffling* where, for each batch, we randomly select image-caption pairs but enforce a constraint: *no two captions from the same image* are allowed within the same batch.

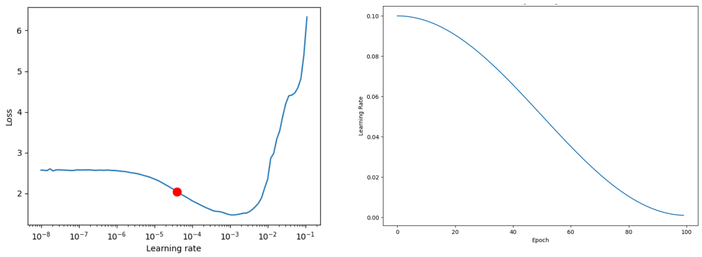
Each strategies help maintain the integrity of the contrastive learning signal and improve the reliability of similarity comparisons during training.

5.4.3 Learning rate and batch size selection

For each dataset, we have training, validation, and test splits. We use only the training and validation splits for model validation. Here, we present the learning rate, scheduler, and batch size used.

Optimal Learning rate: For learning rate selection, we use the *FindLR* library to identify the optimal learning rate by performing multiple forward passes with different learning rates. The idea is to find and track the gradients for each learning rate, and then select the value with the steepest gradient, which corresponds to the point where the loss decreases most rapidly. Figure 18 (a) shows the loss evolution with different learning rates, where the best learning rate is marked at the initial point of the steepest gradient.

Cosine Scheduler: We also use a **cosine annealing scheduler** [22], which adjusts the learning rate over time. The cosine scheduler starts with a high learning rate and gradually decreases it according to a cosine function, promoting stable convergence while allowing the model to escape local minima. Figure 18 (b), shows the evolution of the learning rate with the cosine scheduler over the epochs.



(a) Optimal Learning Rate Selection

(b) Learning Rate Decay over Epochs

Figure 18: Optimal Learning Rate Selection and Cosine Annealing decay

Batch size: Due to GPU memory limitations, we are unable to use a large training batch size directly. To address this, we employ gradient accumulation, which simulates a larger effective batch size by accumulating gradients over several mini-batches before updating the model weights. We set a smaller batch size per step that fits in memory and use the batch accumulation parameter to control how many steps to accumulate. The effective batch size is computed in Equation (8). This approach enables us to train with a larger batch size without exceeding memory constraints.

$$\text{Effective Batch Size} = \text{Gradient\_accumulation} \times \text{batch\_accumulation} \times \text{Batch Size} \tag{8}$$

6 Results and Discussion

We focus exclusively on Text-to-Image Retrieval (T2I), where the goal is to retrieve the most semantically similar image given a textual query. In this section, we briefly mention the models and datasets used, as well as the evaluation metric employed for the retrieval task. We then explain each experiment, present the results, and provide a discussion of the findings.

6.1 Experimental Setup and Evaluation Metrics

Datasets: We evaluate the models on six benchmark remote sensing datasets: RSICD, RSITMD, FIT-RS, SYDNEY, UCM, and NWPU. Each dataset is evaluated in two settings: **Original Datasets**, which include the raw image-caption pairs, and **Fine-Grained** versions, which are enhanced with more detailed and expressive captions, as detailed in Section 3.4.

Models: A wide range of CLIP-based models have been developed for T2I tasks in RS. Due to the large number of available models and datasets, we selected a subset of recent models that have demonstrated state-of-the-art performance on the RSCID and RSITMD benchmarks. Specifically, we focus on the following models:

- **RS-M-CLIP** [36]: uses ViT-B/32 as the image encoder.
- **GeoRSCLIP** [51]: available with two encoders—ViT-B/32 and ViT-L/14.
- **RemoteCLIP** [19]: available with ViT-B/32 and ViT-L/14 as image encoders.

These models represent some of the most recent advancements in the field and serve as strong baselines for evaluating T2I performance in our study.

Hardware and Experimental Environment: All experiments were conducted primarily on an NVIDIA RTX A6000 GPU with 48GB of VRAM using CUDA for efficient model training and evaluation. For development and debugging purposes, we also run models locally on a macOS system equipped with an M4 chip and 32GB of RAM.

Evaluation Metrics for Retrieval To evaluate the performance of the model, we employ widely used retrieval metrics. These metrics help quantify how effectively the model retrieves relevant images based on textual queries:

- **Recall@K (R@1, R@5, R@10)**: Measures the percentage of times the correct image appears in the top-K retrieved results.
- **Mean Recall (mR)**: Computes the average of the three recall values: R@1, R@5, and R@10. It provides an overall measure of retrieval performance across different cutoff points.

In datasets where multiple captions are available for a single image, we treat each caption as an independent query. This means each caption is evaluated as if it corresponds to a distinct image.

6.2 Experiment 1: Human Evaluation for Original vs Fine-Grained datasets

Since retrieval systems are designed for human users, we incorporate human-in-the-loop evaluation to better assess their real-world performance. Human evaluation is crucial because automated models can overfit to minor details like punctuation and caption length, which are not key for retrieval tasks. In contrast, humans are better at capturing nuanced semantic meaning, such as object count and spatial relationships, which are more relevant for retrieval. The captions in many original datasets are generic, and even if models show good results, these results are meaningless to humans. That's why we run human experiments to assess the caption quality for the original and the fine-grained datasets used in retrieval tasks.

6.2.1 Experimental Design

This experiment evaluates how effectively humans can associate textual descriptions with corresponding images in RS datasets. By analyzing user selections, we gain insights into how suitable the datasets are for real-world text-to-image retrieval.

Setup: For each original dataset (RSICD, RSITMD, NWPU) and its fine-grained version, we randomly select 7 different scene categories. For each category, one caption is chosen at random, and 9 images from the same category are gathered (1 image matching the caption and 8 distractors). This results in 7 query sets, each containing a caption and 9 candidate images. Participants are instructed to select the image that best matches the given caption based solely on its semantic content.

Example: We provide two examples of questions that use the same set of images but different captions: one from the original dataset and another from the fine-grained dataset.

- **Original caption:** There are some buildings beside the tennis courts.
- **Fine-Grained caption:** The remote sensing image shows a sports complex and residential area. The sports complex includes a large green field, likely used for soccer or other field sports, and several tennis courts with blue surfaces and white lines. Adjacent to the field is a building with a brown roof. The residential area features houses with varying roof colors, including gray and brown, surrounded by green lawns and trees. A curved road is visible near the houses, suggesting a suburban neighborhood setting. The image highlights the integration of recreational facilities within a residential environment.

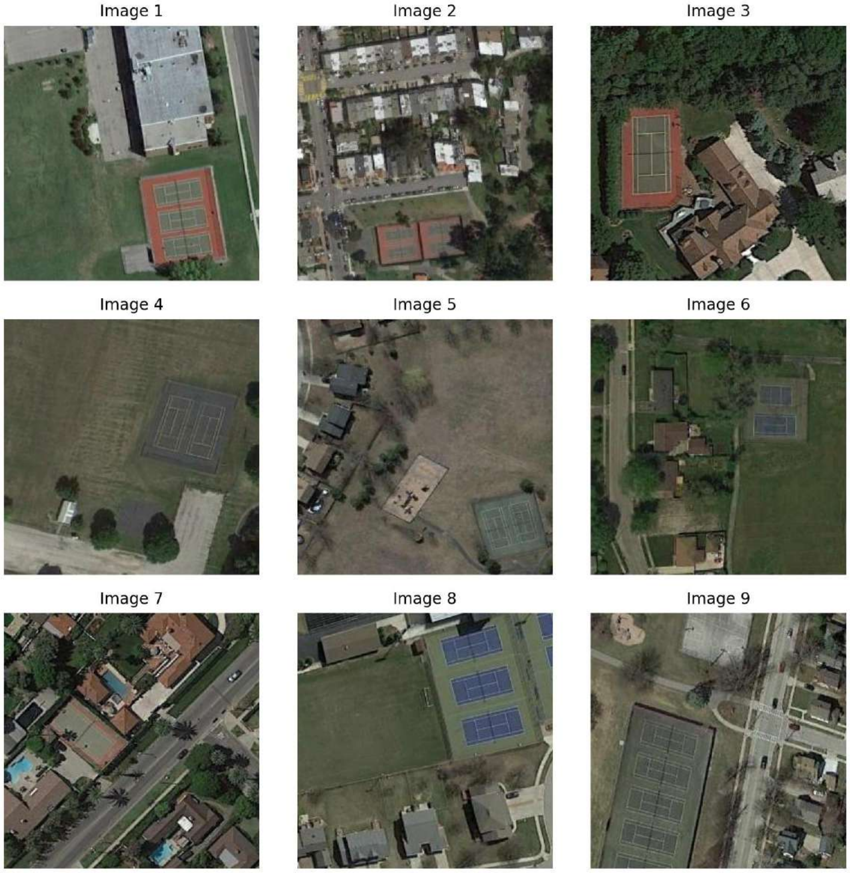


Figure 19: Example Question - Participants select the image that best matches the given caption.

The images belong to the same category and appear visually similar. The original caption is too generic to guide selection accurately, while the fine-grained caption includes distinctive details such as blue tennis courts and a curved road, which help identify the correct image.

6.2.2 Human Retrieval evaluation

We conducted the experiment with 38 EPFL students (bachelor's, master's, and PhD) from various sections. The form had 7 questions for 3 original datasets and their fine-grained versions, resulting in a total of 42 questions. Participants typically completed the forms within 20 minutes. We compared human performance using the following metric:

- **Mean Correct (%):** The average percentage of times participants correctly identified the image matching the given caption. This reflects the overall accuracy of human retrieval.
- **Median Correct (%):** The median percentage of correct matches.
- **Mean Unique Choices:** The average number of distinct images selected by participants for a given caption. Lower values indicate more agreement and less ambiguity in human judgment.
- **Median Unique Choices:** The median number of unique selections.

Table 6 shows a clear trend: fine-grained data outperforms original data in terms of retrieval accuracy. For instance, the Mean Correct % for fine-grained data is significantly higher than for original data. The Median Correct % also confirms this trend. This improved performance is underscored by the Mean Unique Choices and Median Unique Choices metrics. fine-grained data leads to lower values for these metrics, indicating that participants considered fewer images before selection. This suggests that fine-grained captions are more discriminative. In summary, the human evaluation demonstrates that fine-grained datasets result in more accurate and less ambiguous image-caption matching.

Table 6: Human Retrieval Evaluation Results

	NWPU		RSICD		RSITMD	
Metric	**Original**	**Fine-grained**	**Original**	**Fine-Grained**	**Original**	**Fine-Grained**
Mean Correct (%)	16.07	**55.36**	37.50	**66.07**	57.14	**75.00**
Median Correct (%)	0.00	**50.00**	25.00	**87.50**	62.50	**87.50**
Mean Unique Choices	4.00	**3.43**	2.86	**2.29**	3.14	**2.29**
Median Unique Choices	4.00	**3.00**	3.00	**2.00**	3.00	**2.00**

6.3 Experiment 2: Vector Matching

We present baseline results using the vector matching mode. Table 7 shows the comparison between the original datasets and the fine-grained datasets (full results are provided in the Appendix A.2.1).

Dataset Comparison: Original vs. Fine-Grained Across all datasets, we observe a consistent improvement in model performance when transitioning from the original to the fine-grained versions. This outcome is expected for two main reasons. First, as shown in Section 3.4.2, the fine-grained datasets contain more diverse and descriptive captions. Second, human evaluation results (Section 6.2) confirm that the fine-grained captions are more helpful for retrieval tasks.

Model Performance Comparisons GeoRSCLIP models (both ViT-B-32 and ViT-L-14) demonstrate strong generalization, clearly dominating performance on the fine-grained datasets. They consistently outperform RemoteCLIP and RS-M-CLIP across nearly all fine-grained benchmarks. In contrast, RS-M-CLIP, although dominant in the original datasets and often outperforming all other models, suffers a substantial performance drop in the fine-grained setting. RemoteCLIP remains moderately stable but is consistently outpaced by GeoRSCLIP in both settings.

RS-M-CLIP Underperformance and Overfitting RS-M-CLIP performs very well and often achieves the best results on the original datasets, with the exception of FIT-RS. This is expected, as it was trained on four of these datasets, which boosts its scores in the original setting. However, its performance drops sharply on the fine-grained datasets, revealing poor generalization and a clear case of overfitting. This lack of robustness motivated us to include a human evaluation, since high benchmark scores can be misleading and may not reflect real-world utility.

Table 7: Retrieval performance on original vs. fine-grained datasets for various models.

Dataset	Model	Backbone	Original Dataset				Fine-Grained Dataset			
			R@1	R@5	R@10	mR	R@1	R@5	R@10	mR
FIT-RS	RS-M-CLIP	ViT-B-32	1.50	5.43	9.07	5.33	1.87	8.14	13.28	**7.76**
	RemoteCLIP	ViT-L-14	5.14	14.31	22.83	14.09	9.35	22.83	32.09	**21.42**
	GeoRSCLIP	ViT-L-14	7.48	21.33	30.87	19.89	23.76	49.77	61.55	**45.03**
RSICD	RS-M-CLIP	ViT-B-32	44.83	60.27	69.09	**58.06**	6.04	20.31	32.11	19.49
	RemoteCLIP	ViT-L-14	12.90	38.02	53.27	34.73	17.47	44.56	63.77	**41.93**
	GeoRSCLIP	ViT-L-14	9.97	28.18	42.10	26.75	27.36	58.55	74.29	**53.40**
RSITMD	RS-M-CLIP	ViT-B-32	60.22	76.68	83.23	**73.38**	12.17	32.30	45.13	29.87
	RemoteCLIP	ViT-L-14	21.42	54.12	71.37	48.97	32.30	69.03	86.28	**62.54**
	GeoRSCLIP	ViT-L-14	17.21	41.37	56.33	38.30	42.04	80.53	91.37	**71.31**
SYDNEY	RS-M-CLIP	ViT-B-32	22.76	54.48	81.03	**52.76**	8.62	41.38	60.34	36.78
	RemoteCLIP	ViT-L-14	13.10	47.93	65.52	42.18	10.34	51.72	75.86	**45.97**
	GeoRSCLIP	ViT-L-14	15.86	55.17	77.59	49.54	34.48	68.97	86.21	**63.22**
UCM	RS-M-CLIP	ViT-B-32	15.90	54.67	82.67	**51.08**	11.43	28.10	48.10	29.21
	RemoteCLIP	ViT-L-14	17.43	62.38	93.62	57.81	26.19	63.81	89.52	**59.84**
	GeoRSCLIP	ViT-L-14	16.10	54.48	86.19	52.26	33.81	79.05	97.62	**70.16**

6.4 Experiment 3: Graph-Based Matching

Since our focus is on capturing fine-grained details, we created a fine-grained version of the dataset and observed that the baseline model V2V already achieves better results in this setting. In this section, we investigate whether these results can be further improved by using architectures that better exploit fine-grained information. Specifically, we explore graph-based representations, and evaluate two such methods: **G2G** and the hybrid **VG2VG**, against the standard **V2V** baseline.

- **V2V Matching:** This serves as the baseline retrieval method, based on cosine similarity between global embeddings. All other approaches are compared relative to this.
- **G2G Matching:** A graph-only approach where both the text and image inputs are converted into scene graphs. Matching is then performed entirely within the graph space.
- **VG2VG Matching:** A hybrid method that combines information from both the graph-based G2G representation and the original V2V embeddings.

Both the G2G and VG2VG methods are evaluated using two similarity functions: **Hungarian matching** and **GMN**. For each function, we explore two variants: **Vanilla**, which uses only Bert embeddings; and **Trainable**, where the graph embeddings are learned end-to-end. Note that the GMN is inherently a learnable matching function.

Given the large number of datasets, models, and methods, we report only the **mR** defined as the mean of R@1, R@5, and R@10, as the evaluation metric. The following section describes the two matching methods and presents the results analysis.

6.4.1 G2G Matching Evaluation

Setup We evaluate two graph similarity functions: **Hungarian matching** and the **GMN** using both *vanilla* and *trained* graph embeddings. All methods are compared against the baseline **V2V**, using three models: RS-M-CLIP, RemoteCLIP, and GeoRSCLIP, and ViT-B/32 as the vision encoder.

Table 8 presents retrieval results on the original datasets.

Table 8: Retrieval performance (mR) on **Original Data** using V2V and G2G methods.

Dataset	V2V-Cosine			G2G - Hungarian		G2G - GMN	
	RS-M-CLIP	RemoteCLIP	GeoRSCLIP	Vanilla	Trained	Vanilla	Trained
FIT-RS	5.33	11.01	17.77	2.81	4.71	1.72	7.86
RSICD	58.06	30.82	23.94	2.05	7.84	1.65	11.49
RSITMD	73.38	47.92	35.41	1.39	11.45	0.99	21.66
SYDNEY	52.76	48.28	46.67	14.37	32.87	11.49	38.81
UCM	51.08	57.59	49.21	5.93	24.63	2.57	40.33

Table 9 shows retrieval results on the fine-grained datasets.

Table 9: Retrieval performance (mR) on **Fine-Grained Data** using V2V and G2G methods.

Dataset	V2V-Cosine			G2G - Hungarian		G2G - GMN	
	RS-M-CLIP	RemoteCLIP	GeoRSCLIP	Vanilla	Trained	Vanilla	Trained
FIT-RS	7.76	17.43	41.91	3.81	7.67	3.21	11.78
RSICD	19.49	35.16	49.46	4.05	11.85	2.7	22.53
RSITMD	29.87	55.68	69.84	5.68	17.77	1.18	28.80
SYDNEY	36.78	42.53	57.47	18.97	40.80	12.64	49.85
UCM	29.21	52.54	66.19	11.59	29.21	3.49	44.97

Results Analysis
From Table 8 and Table 9 results, we draw three key observations:

- **Training enhances G2G performance.** Across all datasets and both similarity functions, trained embeddings outperform their vanilla models. This shows the importance of updating node embeddings based on local context. While many graphs contain similar nodes, their spatial layouts and relational structures differ. Training allows encoding these local differences by incorporating neighborhood and edge-level information into the embedding process.
- **GMN outperforms Hungarian matching:** GMN outperforms Hungarian matching. The Hungarian algorithm enforces rigid one-to-one node alignments and assumes a fixed node correspondence, which can be too restrictive in noisy or structurally diverse graphs. In contrast, GMN provides a flexible, learnable mechanism for graph similarity. It learns to exploit the pairwise similarity matrix between query and image nodes to extract a similarity score.
- **G2G benefits from fine-grained text data.** As intended in our method design, G2G can effectively exploit the additional information present in fine-grained captions, making graph-based representations well-suited for fine-grained retrieval tasks. The observed performance gains (compared to the original data) in the fine-grained datasets are due to richer captions that led to higher-quality graphs. However, fine-grained captions alone are not sufficient, since V2V can still outperform G2G due to the low quality of the image scene graph.

In summary, while G2G retrieval still trails V2V in performance, it offers unique advantages in interpretability, structure-awareness, and compositional reasoning. These strengths make it a promising direction for future research, particularly when paired with improved image SG generation pipelines.

6.4.2 G2G Results Discussion

We first observe that G2G methods perform significantly worse than V2V, which motivates a deeper investigation into the reasons behind G2G's underperformance and the potential improvements.

Why Graph-Only Matching Underperforms ?

To better understand the reasons behind the underperformance of the G2G, we conducted a manual analysis of the generated graphs, and the retrieval. This analysis revealed several key challenges:

- **Limitations in Image Graph Generation:** VLMs used for image scene graph extraction still struggle to produce fine-grained and grounded graphs. They tend to detect only the most common objects while missing small details. In many cases, object attributes are inferred from general priors rather than actual visual evidence.
- **Lack of Graph Diversity in Datasets:** Many RS scenes have similar layouts, and one graph is often a subgraph of another.For example, a soccer field with parking may appear within a larger scene that also includes tennis courts.
- **Absence of Visual Cues in Graphs:** All graphs, whether derived from captions or images, are built purely from textual descriptions. No visual information from image pixels is used. We tried including visual features (e.g., using embeddings of image crops), but this was ineffective for RS imagery where objects are tiny (often around 5×5 pixels) and distorted by the top-down view. Resizing such small objects for vision encoders results in very poor representations.
- **Lack global context:** Graphs capture object relations but miss broader scene information, such as whether it's an airport, industrial site, or residential area.

Potential Improvements for G2G Matching

There is considerable potential for improving G2G matching. Below, we outline several directions:

- **Grounding with External Metadata:** Incorporating structured metadata such as place names or building functions can enrich the graph with context-specific nodes. For example, identifying a generic "university" node as "EPFL" adds specificity and improves matching relevance, especially for named entity queries.
- **Robustness to Unseen or Emerging Concepts:** Traditional VLM embeds the entire query into a single vector, making them less effective at handling new or rare concepts introduced after training. When a concept is unfamiliar or lacks strong representation in the model's vocabulary, it influences the overall embedding. In contrast, graph-based matching encodes each concept as a separate node, preserving its identity even if the node's embedding is meaningless. This structure allows matching specific entities directly at the node level, enabling reliable zero-shot retrieval even when the concept was never seen during training.
- **Modular Graph Construction:** Instead of constructing a single graph, task-specific graphs (e.g., graphs for damage assessment, building status, or Functional Description) can be generated and dynamically selected based on the retrieval query. This modular approach enables more focused and flexible retrieval, useful in multi-purpose scenarios.
- **Graph Denoising:** Post-processing the generated graphs using LLMs could clean up redundant, irrelevant, or misclassified nodes and standardize node labels. This may improve matching robustness and reduce noise-induced mismatches.
- **Learning Graph Priors from Data:** Rather than using frozen graph construction rules, a learned graph generation module trained on retrieval performance could adaptively decide what objects and relations are most relevant, potentially using reinforcement learning or gradient-based policy optimization.

6.4.3 VG2VG Matching Evaluation

Motivation While V2V offers visual encoding and global context, it often misses finer relational details. Conversely, G2G lacks visual cues and global context altogether. To bridge these gaps, we introduce a hybrid VG2VG method that enables matching from coarse to fine granularity.

Setup The same models are used to generate the global embedding vector, with ViT-B/32 as the vision encoder backbone. Results using alternative backbones are provided in the Appendix A.2.2. We focus only on the fine-grained datasets, as it is better suited for real-world scenarios and validated by humans. Experiments on the original data are included in the Appendix A.2.2.

For clarity, note that our baselines were not trained on the fine-grained datasets. This choice does not undermine the comparison, as VG2VG is not intended to replace these models with a new end-to-end architecture. Rather, it serves as a plug-in layer that can be integrated into any compatible model without full fine-tuning. This approach offers three benefits: (1) efficient training in resource-constrained settings, (2) applicability even without access to model weights, and (3) usability when training code is unavailable, for example, in the case of RS-M-CLIP, whose training code is not publicly released.

Fine-Grained Datasets Results We evaluate the hybrid VG2VG approach on the fine-grained datasets, with results shown in Table 10. Each row corresponds to a dataset and the vision backbone used to generate the global embedding. The goal is to compare VG2VG performance against the V2V baseline. In each row, cells are colored green if the score exceeds the V2V baseline and red if it falls below. % Above Baseline indicates the percentage of cases where our approach outperforms the baseline, while Mean % Gain represents the average improvement over all settings.

Table 10: Retrieval performance on **Fine-Grained Data** using V2V and VG2VG methods

Dataset	**Global embedding backbone**	**V2V**	**VG2VG-Hungarian**		**VG2VG-GMN**	
			Vanilla	**Trained**	**Vanilla**	**Trained**
FIT-RS	RS-M-CLIP	7.76	7.95	10.85	7.58	25.95
	RemoteCLIP_ViT-B-32	17.43	15.25	22.42	17.18	25.67
	GeoRSCLIP_ViT-B-32	41.91	29.87	43.84	41.94	45.34
RSICD	RS-M-CLIP	19.49	17.96	21.32	19.55	26.25
	RemoteCLIP_ViT-B-32	35.16	29.34	37.82	34.80	41.47
	GeoRSCLIP_ViT-B-32	49.46	40.84	48.68	49.53	53.73
RSITMD	RS-M-CLIP	29.87	27.29	34.59	29.79	47.68
	RemoteCLIP_ViT-B-32	55.68	46.98	59.51	55.45	60.73
	GeoRSCLIP_ViT-B-32	69.84	56.86	69.54	69.21	71.21
SYDNEY	RS-M-CLIP	36.78	32.18	41.95	37.93	45.83
	RemoteCLIP_ViT-B-32	42.53	37.35	53.45	42.53	52.15
	GeoRSCLIP_ViT-B-32	57.47	45.98	63.79	59.20	60.20
UCM	RS-M-CLIP	29.21	26.83	37.46	30.16	51.95
	RemoteCLIP_ViT-B-32	52.54	44.60	56.83	52.38	57.67
	GeoRSCLIP_ViT-B-32	66.19	52.38	65.24	66.03	70.05
% Above Baseline		-	**6.67%**	**80.00%**	**40.00%**	**100.00%**
Mean % gain		-	**-6.64%**	**+3.77%**	**+0.13%**	**+8.30%**

Results Analysis

The results obtained from Table 10 are analyzed as follows:

- **VG2VG Methods Offer Significant Performance Gains:** Across all datasets and models, VG2VG-GMN using trained graph embeddings consistently outperform the V2V baseline (100% above baseline). They also achieve the highest mean % gain (8.30 %).
- **GMN outperforms Hungarian matching:** As in the G2G setup, GMN-based variants achieve higher retrieval performance (100% above baseline) compared to their Hungarian counterparts (80% above baseline), with a higher mean percentage gain of 8.3% versus 3.77% for the Hungarian version.
- **Trained Graph Embeddings Lead to Stronger Results:** Training significantly boosts retrieval performance, outperforming the baseline in 80% of cases with a +3.77% mean gain for Hungarian (compared to -5.91% for vanilla), and 100% of cases with a 8.30% mean gain for VG2VG (vs. 0.13% for vanilla).

6.4.4 VG2VG Results Discussion

These results clearly indicate that the VG2VG approach consistently outperforms the standard V2V method, particularly when using trained graph embeddings and GMN-based similarity. This hybrid approach benefits from:

- **Strong Base from V2V Matching:** The CLIP-based model's global embeddings provide a solid foundation for retrieval. Even without fine-grained graph-based information, these embeddings enable effective initial matching. They also offer a comprehensive overview of the scene and its composition, encoding global semantic cues that may be missed by the graph model due to limitations in scene graph generation pipelines.
- **Complementary Graph Semantics:** The graph component significantly enhances retrieval by incorporating semantic and relational details. It captures object-object and object-scene relationships that are often absent in vector embeddings alone.

Together, these factors ensure that the VG2VG approach starts from a strong baseline while benefiting from the finer relational information provided by the graph structure.

Furthermore, we highlight the following two key observations:

- **GMN's Expressiveness:** GMN's flexibility, compared to rigid Hungarian matching, allows for more dynamic and adaptive graph matching. By learning from the pairwise similarity matrix between the query and image nodes, GMN optimizes the matching process, leading to improved retrieval accuracy. This adaptive mechanism makes GMN particularly well-suited for graph-based matching.
- **End-to-End Training for Graph Embeddings:** The importance of training graph embeddings becomes evident. While vanilla graph embeddings are useful, they often fail to outperform V2V matching. Many graphs share similar substructures, for instance, most airport scene graphs include common nodes such as airplanes or taxiways, but the layout and relationships between these nodes differ. End-to-end training allows the graph embeddings to update based on their neighborhood and relationships, making each graph representation more distinctive and discriminative compared to vanilla embeddings.

In summary, while graph-only methods are hindered by limitations in graph generation, hybrid approaches such as VG2VG demonstrate that these challenges can be alleviated by combining global embeddings with structured graph information.

6.5 Experiment 4: Scene-based Matching as a cost-effective alternative

We have shown that the hybrid approach outperforms the baseline for coarse-to-fine solutions. However, it requires high-quality graph construction for both text and images. While building a graph for text is relatively straightforward, generating a scene graph for images is considerably more challenging. Moreover, to unlock effective gains, the approach requires training the graph structure. Although the model is lightweight (with fewer than 10 million parameters), implementing the training pipeline, preparing the data, and monitoring the training process are time-consuming. Motivated by these challenges, we aim to explore a simpler direction: scene matching as a more cost-effective alternative. This method serves as a middle ground between fine-grained and coarse matching. It focuses solely on solving the cropping of the image and splitting of the text, enabling matching without the need for graph construction or additional training.

6.5.1 Scene based Matching Evaluation

Setup We evaluate our approach using fine-grained datasets only, as previously demonstrated to be more effective for retrieval tasks and preferred by human evaluators. The results on the original datasets are provided in Appendix A.2.4 A.2.4.

In this experiment, we compare the performance of vector-to-scene (**V2S**), scene-to-scene (**S2S**), scene-to-vector (**S2V**), and the **V2V** baseline. For **V2S** and **S2S**, we implement two cropping strategies as described in Section 4.2.2: (1) uniform cropping with overlap (UC) and (2) SAM-based segmentation (SAM). All models use ViT-B/32 as the vision encoder, and results for other backbone variants are provided in Appendix A.2.4.

Results: The results are presented in Table 11.

Table 11: Retrieval performance on **fine-grained datasets** using V2V and Scene based matching.

Dataset	**Model**	**V2V**	**V2S**		**S2V**	**S2S**	
			UC	**SAM**		**UC**	**SAM**
FITRS	RS-M-CLIP	7.76	10.77	10.95	16.23	18.03	17.31
	RemoteCLIP_ViT-B-32	17.43	17.87	19.03	20.65	20.40	19.65
	GeoRSCLIP_ViT-B-32	41.91	41.17	43.29	34.56	38.83	36.86
RSICD	RS-M-CLIP	19.49	23.79	23.49	31.78	33.12	31.81
	RemoteCLIP_ViT-B-32	35.16	33.95	34.92	37.06	37.27	37.03
	GeoRSCLIP_ViT-B-32	49.46	52.34	53.04	43.07	48.46	46.51
RSITMD	RS-M-CLIP	29.87	32.98	33.20	50.24	51.27	49.50
	RemoteCLIP_ViT-B-32	55.68	52.30	55.40	58.57	56.43	56.73
	GeoRSCLIP_ViT-B-32	69.84	70.82	72.29	62.70	65.06	63.88
SYDNEY	RS-M-CLIP	36.78	39.06	40.21	46.53	46.53	45.95
	RemoteCLIP_ViT-B-32	42.53	47.68	46.53	43.65	51.70	48.25
	GeoRSCLIP_ViT-B-32	57.47	61.47	62.05	56.87	62.62	61.47
UCM	RS-M-CLIP	29.21	34.63	35.59	48.44	48.44	49.08
	RemoteCLIP_ViT-B-32	52.54	51.77	52.57	59.40	54.32	56.86
	GeoRSCLIP_ViT-B-32	66.19	66.70	67.97	65.74	67.49	69.40
% Above Baseline		-	**73.33%**	**86.66%**	**66.66%**	**80.00%**	**80.00%**
Mean % gain		-	**+1.04%**	**+1.57%**	**+2.57%**	**+3.55%**	**+3.11%**

Results Analysis

The results obtained from Table 11 are analyzed as follows:

- **Scene-based representations outperform the V2V baseline:** In every dataset-model combination, at least one of the scene-based variants (V2S, S2V, or S2S) outperforms V2V. This demonstrates the effectiveness of splitting the text or image into multiple scenes.

- **V2S vs V2V - Reliable improvements with segmentation-based cropping:** V2S outperforms V2V across most models and datasets. In cases where performance decreases (e.g., RemoteCLIP), the drop is marginal (less than 0.5%). SAM-based segmentation outperforms uniform cropping (UC) in 86.66% of cases (compared to 73.33% for UC), with a mean gain of 1.57% (versus 1.04% for UC), highlighting the benefits of content-aware segmentation.

- **S2V vs V2V/V2S - Higher potential, but model dependent:** S2V provides significant improvements over both V2V and V2S for RS-M-CLIP and RemoteCLIP, indicating that leveraging multiple text embeddings enhances semantic alignment. However, it consistently underperforms on GeoRSCLIP across all datasets. This explains the lower % above baseline (66.66%) compared to V2S-SAM, despite S2V achieving a higher mean gain of 2.57%.

- **Balanced and robust performance of S2S:** S2S combines the benefits of both V2S and S2V, delivering the best performance on RS-M-CLIP and RemoteCLIP while mitigating S2V's weaknesses on GeoRSCLIP. It achieves a strong % above threshold (not the highest, but close) and the highest mean gain of 3.55%.

- **SAM vs Uniform Cropping:** In V2S, SAM-based cropping outperforms UC (mean gain of 1.57% vs. 1.04%). In contrast, in the S2S setup, SAM underperforms compared to UC (3.11% vs. 3.55%). This highlights that the effectiveness of scene-based matching is highly sensitive to the cropping strategy.

- **Text as Scene improves performance:** Both S2V and S2S outperform V2S, achieving higher mean gains of 3.55% and 2.57%, respectively, compared to 1.57% for the V2S setting.

6.5.2 Results Discussion

Among all approaches, S2S shows the most consistent performance across backbones and datasets, making it our choice for comparison. We also highlight two key aspects from the obtained results:

1. **Text as Scene is important:** Splitting captions into multiple segments and treating each segment as a separate scene can be crucial for effective retrieval. Encoding the entire caption into a single CLIP vector risks losing important information, especially given CLIP's token limit, which can cause longer captions to be truncated. Dividing text into smaller, meaningful segments helps preserve detail and better aligns with scene-based representations.

2. **Image cropping strategy depends on the matching method.** The choice of image cropping strategy can significantly impact performance, and its effectiveness varies depending on the matching setup. For example, SAM-based segmentation tends to work well in V2S setups, while uniform cropping often performs better in S2S. One reason is that uniform cropping with overlap produces a fixed number of scenes, whereas SAM generates a variable number of segments depending on the image content. This difference affects how the image information is distributed across scenes and can influence the alignment with textual scene representations.

6.6 Retrieval Modes Comparison and Use Case

Given the variety of results across different methods and implementations, we selected the best-performing model from each matching mode (V2V, graph-based, and scene-based) for comparison. For graph based methods, we identify VG2VG using GMN as the most effective approach. In the scene-based category, S2S with the UC strategy achieves the best performance.

In this section, we compare the three selected models in terms of retrieval performance, relative inference cost, and discuss their potential use cases.

6.6.1 Retrieval Performance

Table 12 presents a comprehensive comparison of retrieval performance across different datasets.

Table 12: Comparison of retrieval performance: V2V Baseline vs. S2S-UC and VG2VG-GMN

Dataset	**Model**	**Original Dataset**			**Fine-Grained Dataset**		
		V2V	**S2S UC**	**VG2VG GMN**	**V2V**	**S2S UC**	**VG2VG GMN**
FIT-RS	RS-M-CLIP	5.33	10.83	8.64	7.76	18.03	25.94
	RemoteCLIP	11.01	14.26	13.34	17.43	20.40	25.67
	GeoRSCLIP	17.77	19.47	19.92	41.91	38.83	45.34
RSICD	RS-M-CLIP	58.06	60.67	57.75	19.49	33,13	26.25
	RemoteCLIP	30.82	33.78	30.82	35.16	37.27	41.47
	GeoRSCLIP	23.94	28.23	24.23	49.46	48.46	53.73
RSITMD	RS-M-CLIP	73.38	78.29	73.45	29.87	51.27	47.68
	RemoteCLIP	47.92	48.06	49.08	55.68	56.43	60.73
	GeoRSCLIP	35.41	38.13	26.46	69.84	65.06	71.21
SYDNEY	RS-M-CLIP	52.76	54.23	52.30	36.78	46.53	45.83
	RemoteCLIP	48.28	50.55	49.20	42.53	51.70	52.15
	GeoRSCLIP	46.67	44.00	48.39	57.47	62.62	60.20
UCM	RS-M-CLIP	51.08	57.08	50.51	29.21	48.84	51.95
	RemoteCLIP	57.59	59.17	47.94	52.54	54.32	57.67
	GeoRSCLIP	49.21	50.67	51.14	66.19	67.49	70.05
% Above Baseline		-	**93.33%**	**53.33%**	-	**80.00%**	**100%**
Mean % gain		-	**+4.11%**	**+1.86%**	-	**+3.55%**	**+8.30%**

From the results, we observe that the VG2VG approach consistently achieves the best performance on fine-grained datasets across all cases (100% above baseline) with the highest mean gain of 8.30%, highlighting its strength in structured retrieval tasks where graph-based relations enhance semantic alignment. In contrast, for the original datasets, S2S (using the uniform cropping variant) outperforms other approaches in 93.33% of cases and achieves the highest mean gain of 4.11%.

Overall, while S2S performs best on original datasets, despite weaknesses in scene graph generation, VG2VG proves most effective for fine-grained datasets where object relations are clearer and more structured, making it particularly suitable for real-world retrieval tasks.

6.6.2 Quantitative Cost and Process Analysis of Matching Methods

We provide a detailed analysis of the data processing and computational costs associated with three text-to-image retrieval methods: **V2V**, **VG2VG**, and **S2S**. Each method involves different pipelines. The tables below explain the steps, resources, and cost implications. Table 13 outlines the data preparation and inference steps required for each method.

Table 13: Pipeline Steps per Cross-Modal Matching Method

Method	**Data Preparation Steps**	**Inference Steps**
V2V	1. Compute CLIP embedding for each image. 2. Store embedding in database.	1. Compute CLIP embedding from text. 2. Perform cosine similarity with all embeddings. 3. Rank based on similarity scores.
VG2VG	1. Compute CLIP embedding for each image. 2. Generate image graph using Gemini. 3. Encode graphs using a graph embedding model. 4. Store both graphs and global embeddings.	1. Generate text query graph using Gemini. 2. Compute query graph embedding. 3. Perform late interaction using GMN. 4. Rank based on similarity scores.
S2S	1. Segment and crop each image. 2. Compute CLIP embeddings for each crop. 3. Store embeddings in database.	1. Generate scene text using Gemini. 2. Compute CLIP embeddings for each scene. 3. Perform late interaction for scoring. 4. Rank based on similarity scores.

In Table 14, we report inference performance on the RSICD dataset (1,069 images and 5,345 queries), focusing on runtime since data preparation is performed offline. A batch size of 1,024 is used, limited by GPU memory (NVIDIA RTX A6000, 48 GB). All image embeddings and databases are pre-initialized. CLIP ViT-B/32 serves as the backbone model for all variants. The implementation is optimized to support batched processing, inference, and multiprocessing for both graph and scene generation. Key performance indicators (KPI), are reported in the last columns of Table 14.

Table 14: Inference-Time Cost and Resource Analysis for Text to Image Retrieval

Method	**Inference Steps**	**Time per Step (s)**	**KPI**
V2V	1. No transformation needed 2. Compute CLIP embedding from text 3. Perform cosine similarity with all embeddings 4. Rank based on similarity scores	Step 1: 0 Step 2: 6.53 ± 1.5 Step 3: 0.03 ± 0.01 Step 4: 0.07 ± 0.02	**Total Time: 13.13 ± 1.5** **Total Tokens:** *0* **Total Cost:** *0 $*
VG2VG	1. Generate text query graph using Gemini 2. Compute query graph embedding 3. Perform late interaction using GMN 4. Rank based on similarity scores	Step 1: ∼48.00 Step 2: 2.40 ± 0.3 Step 3: 2.11 ± 0.1 Step 4: 0.07 ± 0.02	**Total Time: ∼52.51 ± 0.4** **Total Tokens:** *∼2M* **Total Cost:** < *1 $*
V2S	1. Generate scene text using Gemini 2. Compute CLIP embeddings for each scene 3. Perform late interaction for scoring 4. Rank based on similarity scores	Step 1: ∼13.00 Step 2: 13.33 ± 3.0 Step 3: 0.33 ± 0.05 Step 4: 0.07 ± 0.02	**Total Time: ∼26.73 ± 3.1** **Total Tokens:** *∼2M* **Total Cost:** < *1 $*

From Table 15, V2V is the fastest method, requiring no transformation. VG2VG and V2S are slower due to Gemini-based graph or scene generation. While late interaction is efficient, most cost arises from the generation stage. Additionally, VG2VG embedding is fast, as it encodes short node labels (1–2 words), unlike S2S and V2V, which embed scene or full captions. To summarize the computational complexity and data preprocessing requirements, we present a comparison in Table 15.

Table 15: Qualitative Comparison - Complexity and Preprocessing

Method	Computational Complexity	Data Preprocessing
V2V	Low. Relies on simple cosine similarity.	Low. Requires embedding the entire content into a single vector.
S2S	Medium. Involves scene segmentation and late interaction scoring.	Medium. Requires scene segmentation for both modalities.
VG2VG	High. Involves complex graph matching and network computations.	High. Requires generating full scene graphs for both modalities.

6.6.3 Use Case Scenarios

Considering the advantages, limitations, computational cost, and performance (Table 12), we identify the potential use cases for each method. This combined analysis highlights their suitability across different deployment scenarios. We summarize these insights below.

V2V: This method is ideal for fast retrieval without deep semantic understanding:

- Suitable for large-scale retrieval tasks.
- Prioritizes speed and low computational cost.
- Less effective for fine-grained or semantic understanding.
- Requires training on a large amount of data for new concepts and tasks, making the training process resource-intensive.

VG2VG: This method is ideal for tasks that require detailed object-level or relational understanding. It supports graph-based representations and can handle multiple graph-image representations:

- Ideal for object-level and relational understanding.
- Suitable for tasks requiring graph-based representation.
- Supports multiple graph-image representations, each focusing on task-specific graphs.
- Robust to unseen or emerging concepts, enabling zero-shot retrieval for new concepts.
- Training is fast and efficient compared to V2V training.
- High computational cost.

S2S: This matching mode strikes a balance between performance and semantic depth:

- Balances performance and semantic depth.
- Leverages scene segmentation for better object-level context.
- Require training like V2V for new concepts.
- Well-suited for complex yet scalable search tasks.

7 Conclusion

This research presents a new cross-modal retrieval framework aimed at improving text-to-image retrieval in RS by moving beyond traditional vector-based similarity methods. We conducted an in-depth investigation into existing image-caption datasets to assess their suitability for retrieval tasks. Based on this, we introduced new re-captioned datasets using a VLM and designed a human evaluation experiment to validate the quality of this newly created data. The obtained results show that human evaluators prefer the new datasets over the original ones and consider them better suited for real-world retrieval scenarios.

We also proposed a graph-based representation approach to overcome the limitations of fixed vector representations. While the initial G2G matching underperformed compared to the baseline, we identified several challenges in the graph generation pipeline from images. Despite this, the method remains promising, with significant opportunities for future work to address its limitations. In the meantime, our hybrid VG2VG approach demonstrated significant performance improvements over the standard V2V baseline, especially when leveraging trained graph embeddings and the proposed GMN. VG2VG performed particularly well on fine-grained datasets, which contain richer, more diverse, and detailed captions. This emphasizes the value of combining fine-grained graph-based representations with global vector embeddings.

Additionally, we introduced a new scene-based matching method. This alternative does not require graph construction or training, yet still outperformed V2V baseline. In some cases, it also surpassed graph-based matching in the original datasets due to the limitations in caption quality that can affect graph generation. Our findings suggest that the hybrid VG2VG approach, particularly with accurate graph generation, holds the most promise for future cross-modal retrieval systems when captions are long enough and detailed. Additionally, scene-based matching (S2S) offers a cheaper alternative that is also better than V2V matching.

Limitation and Future work A major limitation found in this research is the significant underperformance of the G2G matching methods compared to V2V. This underperformance can be attributed to several challenges, including the difficulty of VLMs in generating fine-grained graphs, as well as a lack of graph diversity in the datasets. The generated graphs often missed small details and inferred attributes from general priors rather than actual visual evidence.

The findings suggest several promising paths for future research, such as:

- **Development of Robust Open Vocabulary SGG pipelines:** There is a clear need for more advanced pipelines capable of producing fine-grained and grounded graphs for RS imagery. Future work should focus on enhancing the accuracy and details of scene graph extraction.
- **Exploration of Agentic Retrieval Systems:** Further research should focus on agentic retrieval systems that generate diverse modular representations such as vectors, graphs, and scene-based embeddings. Then, the agent dynamically ranks and selects the representation that best suits each query's needs.
- **More Hybrid Approaches:** Integrating hybrid approaches that combine scene-based and graph-based embeddings could improve retrieval systems.
- **Learnable Scene Segmentation Models:** We demonstrated earlier that S2S (Scene-to-Scene) matching is sensitive to the image segmentation technique used. Future research could explore approaches where both image and text segmentation are implemented as learnable modules.

References

[1] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 384–392, 2019.

[2] Yakoub Bazi, Laila Bashmal, Mohamad Mahmoud Al Rahhal, Riccardo Ricci, and Farid Melgani. RS-LLaVA: A large vision-language model for joint captioning and question answering in remote sensing imagery. 16(9):1477.

[3] Weizhi Chen, Jingbo Chen, Yupeng Deng, Jiansheng Chen, Yuman Feng, Zhihao Xi, Diyou Liu, Kai Li, and Yu Meng. LRSCLIP: A vision-language foundation model for aligning remote sensing image with longer text.

[4] Qimin Cheng, Haiyan Huang, Yuan Xu, Yuzhuo Zhou, Huanying Li, and Zhongyuan Wang. Nwpu-captions dataset and mlca-net for remote sensing image captioning. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

[5] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[7] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*, 2024.

[8] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Towards open-vocabulary scene graph generation with prompt-based finetuning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13688, pages 56–73. Springer Nature Switzerland. Series Title: Lecture Notes in Computer Science.

[9] Han Hu, Chengkai Li, Yuanliang Yang, Jie Zhang, Suhang Wang, and Jingrui He. Llms for knowledge graph construction and reasoning. *arXiv preprint*, abs/2305.13168, 2023.

[10] Zhong Ji, Changxu Meng, Yan Zhang, Yanwei Pang, and Xuelong Li. Knowledge-aided momentum contrastive learning for remote-sensing image text retrieval. 61:1–13.

[11] Zhong Ji, Changxu Meng, Yan Zhang, Haoran Wang, Yanwei Pang, and Jungong Han. Eliminate before align: A remote sensing image-text retrieval framework with keyword explicit reasoning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pages 1662–1671. Association for Computing Machinery.

[12] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678. IEEE.

[13] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[14] Lin Li, Chuhan Zhang, Dong Zhang, Chong Sun, Chen Li, and Long Chen. Taking a closer look at interacting objects: Interaction-aware open vocabulary scene graph generation.

[15] Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models.

[16] X. Li, B. Qu, D. Tao, and X. Lu. Rsitmd: Remote sensing image–text matching dataset. In *Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, 2019.

[17] Yansheng Li, Linlin Wang, Tingzhu Wang, Xue Yang, Junwei Luo, Qi Wang, Youming Deng, Wenbin Wang, Xian Sun, Haifeng Li, Bo Dang, Yongjun Zhang, Yi Yu, and Junchi Yan. STAR: A first-ever dataset and a large-scale benchmark for scene graph generation in large-size satellite imagery.

[18] Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. Graph structured network for image-text matching.

[19] Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. RemoteCLIP: A vision language foundation model for remote sensing.

[20] Tao Liu, Rongjie Li, Chongyu Wang, and Xuming He. Relation-aware hierarchical prompt for open-vocabulary scene graph generation.

[21] Yifan Liu, Yujie Qian, Fuzheng Zhang, Yanyan Zhao, and Xuanjing Huang. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. *arXiv preprint*, abs/2404.03868, 2024.

[22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[23] 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.

[24] 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. version: 1.

[25] Dilxat Muhtar, Zhenshi Li, Feng Gu, Xueliang Zhang, and Pengfeng Xiao. LHRS-bot: Empowering remote sensing with VGI-enhanced large multimodal language model.

[26] Manh-Duy Nguyen, Binh T. Nguyen, and Cathal Gurrin. A deep local and global scene-graph matching for image-text retrieval.

[27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[28] Jiancheng Pan, Yanxing Liu, Yuqian Fu, Muyuan Ma, Jiahao Li, Danda Pani Paudel, Luc Van Gool, and Xiaomeng Huang. Locate anything on earth: Advancing open-vocabulary object detection for remote sensing community.

[29] 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.

[30] B. Qu, X. Li, D. Tao, and X. Lu. Sydney-captions: A high-resolution aerial image captioning dataset. *International Journal of Applied Earth Observation and Geoinformation*, 89:102047, 2020.

[31] Bo Qu, Xuelong Li, Dacheng Tao, and Xiaoqiang Lu. Deep semantic understanding of high resolution remote sensing images. In *Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2016.

[32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[33] Colorado J. Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-MAE: A scale-aware masked autoencoder for multiscale geospatial representation learning.

[34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[35] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

[36] João Daniel Silva, Joao Magalhaes, Devis Tuia, and Bruno Martins. Multilingual vision-language pre-training for the remote sensing domain. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, pages 220–232.

[37] João Daniel Silva, João Magalhães, Devis Tuia, and Bruno Martins. Large language models for captioning and retrieving remote sensing images.

[38] Xintian Sun, Benji Peng, Charles Zhang, Fei Jin, Qian Niu, Junyu Liu, Keyu Chen, Ming Li, Pohsun Feng, Ziqian Bi, Ming Liu, and Yichao Zhang. From pixels to prose: Advancing multi-modal language models for remote sensing.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1710.10903.

[40] 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.

[41] Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. Cross-modal scene graph matching for relationship-aware image-text retrieval.

[42] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pages 24824–24837. Curran Associates, Inc., 2022.

[43] Xingxing Weng, Chao Pang, and Gui-Song Xia. Vision-language modeling meets remote sensing: Models, datasets and perspectives.

[44] Jin-Ge Yao, Dongyan Zhao, Minghui Qiu, Xuan Su, Qingcai Chen, and Yang Liu. Extracting knowledge graphs from plain text with language models. *arXiv preprint*, abs/2502.09956, 2025.

[45] 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. 60:1–19.

[46] Zhiqiang Yuan, Wenkai Zhang, Changyuan Tian, Xuee Rong, Zhengyuan Zhang, Hongqi Wang, Kun Fu, and Xian Sun. Remote sensing cross-modal text-image retrieval based on global and local information. 60:1–16.

[47] Yang Zhan, Zhitong Xiong, and Yuan Yuan. SkyEyeGPT: Unifying remote sensing vision-language tasks via instruction tuning with large language model. 221:64–77.

[48] Wei Zhang, Miaoxin Cai, Tong Zhang, Jun Li, Yin Zhuang, and Xuerui Mao. EarthMarker: A visual prompting multi-modal large language model for remote sensing.

[49] 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.

[50] Yan Zhang, Zhong Ji, Changxu Meng, Yanwei Pang, and Jungong Han. iEBAKER: Improved remote sensing image-text retrieval framework via eliminate before align and keyword explicit reasoning.

[51] Zilun Zhang, Tiancheng Zhao, Yulong Guo, and Jianwei Yin. RS5m and GeoRSCLIP: A large scale vision-language dataset and a large vision-language model for remote sensing. 62:1–23.

[52] Yijie Zheng, Weijie Wu, Qingyun Li, Xuehui Wang, Xu Zhou, Aiai Ren, Jun Shen, Long Zhao, Guoqing Li, and Xue Yang. InstructSAM: A training-free framework for instruction-oriented remote sensing object recognition.

A Appendix

A.1 System prompts

Here, we present all the system prompts used for the VLM and LLM to generate the fine-grained data, scene graphs form images and texts, and the scenes from text.

A.1.1 System prompts for generating fine grained caption

Fine-Grained Captioning Prompt

```
You are an advanced AI model capable of understanding and
describing remote sensing images.  Your task is to generate a
detailed textual description of the image content.  Please ensure
your description is clear, concise, and captures the key elements
of the scene.  Start your response immediately by the description.
Do not include introductory phrases like 'Here is a description...'
```

A.1.2 System prompts for Image graph generation pipeline

Object Counting Prompt

You are an advanced AI model capable of analyzing satellite images. Your task is to detect clearly visible objects in the image and estimate their counts.

Instructions:

1. Only include objects that are clearly identifiable (e.g., buildings, roads, trees, cars, playgrounds, rivers).
2. Use exact counts where possible. If the count is unclear or very large, use the value -1.
3. Return your output as a valid JSON **object**, not a string.
4. The JSON must include one field: object_counts, which is a dictionary mapping each object type to its count.

Example Output:

```
{
  "object_counts": {
    "building": 5,
    "tree": -1,
    "road": 2,
    "car": 3
  }
}
```

Scene Graph Generation Prompt

You are an advanced AI model capable of analyzing satellite images. Based on the object counts and visual understanding of the image, generate a structured scene graph capturing object relationships, attributes, and an overall scene label.

Your response must follow this format:

- A valid JSON object.
- Must include the following fields:
  - object_counts (same dictionary as in the object counting step)
  - relations: a list of dictionaries with keys: [subject, relation, object, subject_id, object_id]

Graph Generation Rules:

1. Each object must be assigned a consistent ID.
2. Use meaningful spatial relationships: adjacent_to, on, inside, connected_to, etc.
3. Each object must have at least one attribute (e.g., color, shape, material).
4. Include a global scene node (e.g., Airport, Urban Area) and connect it to objects via spatial relations.
5. Avoid vague relations like next_to, and do not use the same object as both subject and object.

Example Relation Entry:

```
{
  "subject": "building",
  "relation": "adjacent_to",
  "object": "road",
  "subject_id": 0,
  "object_id": 1
}
```

A.1.3 System prompts for text graph generation pipeline

Text Graph Generation Prompt

Object-Attribute Graph Extraction Guidelines

- Extract a connected object-attribute graph from descriptive text.
- The output must include object nodes, attribute nodes, and clearly labeled relationships.
- Only use entities (e.g., car, tree, building) as subjects. Attributes can only appear as objects.
- Attributes are separate nodes linked by labeled edges (e.g., car -(color)-> red).
- Preserve the directionality of relationships as stated in the input.
- Use exact relationship phrases from the input.
- Normalize object names and avoid duplicates.
- Quantify objects using:
  - Number words for exact counts (e.g., two)
  - Quantifiers for vague counts (e.g., many)

Format

- Nodes: [object, attribute]
- Edges: [(subject, relation, object)]

A.1.4 System prompts for text scene generation pipeline

Text Scene Generation Prompt

Scene Description Extraction Guidelines

- You will receive a paragraph describing a visual scene.
- Your task is to extract concise, focused, and self-contained scene descriptions.

Instructions

- If the input is a short caption with no interactions, return nothing.
- Split into at most 6 scenes with at least 2 interacting objects each.
- Do not add or infer any objects or events not mentioned.
- Preserve wording and object counts as described.
- Avoid vague words like another, etc.
- Describe interactions directly.

Example

The remote sensing image shows an airport tarmac with several aircraft. A large passenger jet is centrally located, casting a long shadow. Other smaller aircraft are parked nearby, some partially obscured by the larger jet. The tarmac surface is a light gray color, and there are some indistinct structures or equipment visible around the aircraft. The image appears to be taken from a high angle, providing a clear view of the aircraft arrangement on the ground.

Output:

- A large passenger jet casting a long shadow on the airport tarmac
- Smaller aircraft parked nearby, partially obscured by the larger jet
- Light gray tarmac surface surrounding the parked aircraft
- Indistinct structures or equipment visible around the aircraft
- A high-angle view of the aircraft arrangement on the tarmac
- The overall scene showing multiple aircraft on the airport tarmac

A.2 Full results

A.2.1 Full results for Vector Matching

We present baseline results using the vector matching mode. Table 16 compares the original and fine-grained datasets. This table contains the full results across all datasets, matching modes, and backbones.

Table 16: Retrieval performance on original vs. fine-grained datasets for various models.

Dataset	Model	Backbone	Original Dataset				Fine-Grained Dataset			
			R@1	R@5	R@10	mR	R@1	R@5	R@10	mR
FIT-RS	RS-M-CLIP	ViT-B-32	1.50	5.43	9.07	5.33	1.87	8.14	13.28	**7.76**
	RemoteCLIP	ViT-B-32	3.37	11.60	18.05	11.01	5.99	18.80	27.50	**17.43**
	RemoteCLIP	ViT-L-14	5.14	14.31	22.83	14.09	9.35	22.83	32.09	**21.42**
	GeoRSCLIP	ViT-B-32	6.64	19.18	27.50	17.77	20.02	46.59	59.12	**41.91**
	GeoRSCLIP	ViT-L-14	7.48	21.33	30.87	19.89	23.76	49.77	61.55	**45.03**
NWPU	RS-M-CLIP	ViT-B-32	9.76	31.12	46.22	**29.03**	3.37	12.67	19.97	12.00
	RemoteCLIP	ViT-B-32	3.55	12.79	21.28	12.54	6.63	20.70	31.52	**19.62**
	RemoteCLIP	ViT-L-14	4.15	14.92	24.68	14.58	9.40	27.30	39.97	**25.56**
	GeoRSCLIP	ViT-B-32	4.03	14.48	23.36	13.96	15.46	38.83	53.90	**36.06**
	GeoRSCLIP	ViT-L-14	4.41	15.24	24.37	14.67	17.46	42.95	57.56	**39.32**
RSICD	RS-M-CLIP	ViT-B-32	44.83	60.27	69.09	**58.06**	6.04	20.31	32.11	19.49
	RemoteCLIP	ViT-B-32	10.67	32.59	49.20	30.82	13.17	37.79	54.53	**35.16**
	RemoteCLIP	ViT-L-14	12.90	38.02	53.27	34.73	17.47	44.56	63.77	**41.93**
	GeoRSCLIP	ViT-B-32	8.45	25.03	38.35	23.94	23.60	53.98	70.81	**49.46**
	GeoRSCLIP	ViT-L-14	9.97	28.18	42.10	26.75	27.36	58.55	74.29	**53.40**
RSITMD	RS-M-CLIP	ViT-B-32	60.22	76.68	83.23	**73.38**	12.17	32.30	45.13	29.87
	RemoteCLIP	ViT-B-32	20.00	52.52	71.24	47.92	24.78	61.28	80.97	**55.68**
	RemoteCLIP	ViT-L-14	21.42	54.12	71.37	48.97	32.30	69.03	86.28	**62.54**
	GeoRSCLIP	ViT-B-32	12.35	38.58	55.31	35.41	38.72	79.42	91.37	**69.84**
	GeoRSCLIP	ViT-L-14	17.21	41.37	56.33	38.30	42.04	80.53	91.37	**71.31**
SYDNEY	RS-M-CLIP	ViT-B-32	22.76	54.48	81.03	**52.76**	8.62	41.38	60.34	36.78
	RemoteCLIP	ViT-B-32	15.86	54.14	74.83	**48.28**	12.07	44.83	70.69	42.53
	RemoteCLIP	ViT-L-14	13.10	47.93	65.52	42.18	10.34	51.72	75.86	**45.97**
	GeoRSCLIP	ViT-B-32	16.90	50.00	73.10	46.67	25.86	65.52	81.03	**57.47**
	GeoRSCLIP	ViT-L-14	15.86	55.17	77.59	49.54	34.48	68.97	86.21	**63.22**
UCM	RS-M-CLIP	ViT-B-32	15.90	54.67	82.67	**51.08**	11.43	28.10	48.10	29.21
	RemoteCLIP	ViT-B-32	17.52	61.14	94.10	**57.59**	17.62	58.57	81.43	52.54
	RemoteCLIP	ViT-L-14	17.43	62.38	93.62	57.81	26.19	63.81	89.52	**59.84**
	GeoRSCLIP	ViT-B-32	14.76	51.05	81.81	49.21	30.48	71.43	96.67	**66.19**
	GeoRSCLIP	ViT-L-14	16.10	54.48	86.19	52.26	33.81	79.05	97.62	**70.16**

We can see that the pattern is consistent, with the last column (mR) for the synthetic dataset showing modest values highlighted in bold.

A.2.2 Full resutls for VG2VG matching

We evaluate the hybrid VG2VG approach on the original and fine-grained dataset, with results shown in Table 17 and Table 18. Each row corresponds to a dataset and the vision backbone used to generate the global embedding. The goal is to compare VG2VG performance against the V2V baseline. In each row, cells are colored green if the score exceeds the V2V baseline and red if it falls below. % Above Baseline indicates the percentage of cases where our approach outperforms the baseline, while Mean % Gain represents the average improvement over all settings.

Table 17: Retrieval performance (mR) on **Original Data**

Dataset	**backbone**	**V2V**	**VG2VG - Hungarian**		**VG2VG - GMN**	
			Vanilla	**Trained**	**Vanilla**	**Trained**
FIT-RS	RS-M-CLIP	5.33	4.89	6.42	5.24	8.64
	RemoteCLIP_ViT-B-32	11.01	7.92	12.50	10.98	13.34
	GeoRSCLIP_ViT-B-32	17.77	11.54	18.99	17.78	19.92
RSICD	RS-M-CLIP	58.06	54.47	58.54	58.10	57.75
	RemoteCLIP_ViT-B-32	30.82	23.54	30.45	30.83	30.82
	GeoRSCLIP_ViT-B-32	23.94	17.24	22.79	23.92	24.23
RSITMD	RS-M-CLIP	73.38	71.50	74.09	73.21	73.45
	RemoteCLIP_ViT-B-32	47.92	38.70	46.25	47.80	48.08
	GeoRSCLIP_ViT-B-32	35.41	26.80	34.49	35.47	36.46
SYDNEY	RS-M-CLIP	52.76	49.08	52.07	52.99	52.30
	RemoteCLIP_ViT-B-32	48.28	35.40	45.63	48.05	49.20
	GeoRSCLIP_ViT-B-32	46.67	38.62	47.35	46.44	48.39
UCM	RS-M-CLIP	51.08	44.76	52.09	51.08	50.51
	RemoteCLIP_ViT-B-32	57.59	42.64	56.03	57.72	57.94
	GeoRSCLIP_ViT-B-32	49.21	34.06	46.41	48.83	51.14
% Above Baseline		-	**0%**	**46.66%**	**40.00%**	**73.33%**
Mean % gain		-	**-7.20%**	**-0.35%**	**-0.05%**	**+1.86%**

We observe the following pattern: trained embeddings consistently outperform vanilla ones, and GMN outperforms Hungarian matching. However, the improvement above the baseline is modest (+1.86% mean gain). This is expected, as in the original datasets the captions are often very generic, resulting in text graphs that lack sufficient descriptive detail to capture fine-grained relationships between scene elements. In such cases, graph-based methods are less effective. In the hybrid mode, performance tends to be more influenced by the global vector embeddings, and the limited detail in the captions constrains the contribution of the graph component. These observations align with the hypothesis that richer, more specific textual descriptions are necessary for graph-based retrieval to realize its full potential. Consequently, we expect the effects of graph modeling to be more pronounced in fine-grained datasets where caption diversity are higher.

Table 18 presents the results for the fine-grained datasets, where the same overall trend holds: training is necessary, and GMN remains the best-performing similarity function. In this case, the contribution of the graph component in the hybrid mode is more evident, with an average improvement of 8.30% over the baseline. This larger gain highlights the value of incorporating detailed graph structures when captions provide richer semantic and relational information. The fine-grained captions allow the graph component to capture more nuanced spatial relationships and object interactions, which enhances retrieval performance.

Table 18: Retrieval performance (mR) on **Fine-Grained Data**

Dataset	**Global embedding backbone**	**V2V**	**VG2VG-Hungarian**		**VG2VG-GMN**	
			Vanilla	**Trained**	**Vanilla**	**Trained**
FIT-RS	RS-M-CLIP	7.76	7.95	10.85	7.58	25.95
	RemoteCLIP_ViT-B-32	17.43	15.25	22.42	17.18	25.67
	GeoRSCLIP_ViT-B-32	41.91	29.87	43.84	41.94	45.34
RSICD	RS-M-CLIP	19.49	17.96	21.32	19.55	26.25
	RemoteCLIP_ViT-B-32	35.16	29.34	37.82	34.80	41.47
	GeoRSCLIP_ViT-B-32	49.46	40.84	48.68	49.53	53.73
RSITMD	RS-M-CLIP	29.87	27.29	34.59	29.79	47.68
	RemoteCLIP_ViT-B-32	55.68	46.98	59.51	55.45	60.73
	GeoRSCLIP_ViT-B-32	69.84	56.86	69.54	69.21	71.21
SYDNEY	RS-M-CLIP	36.78	32.18	41.95	37.93	45.83
	RemoteCLIP_ViT-B-32	42.53	37.35	53.45	42.53	52.15
	GeoRSCLIP_ViT-B-32	57.47	45.98	63.79	59.20	60.20
UCM	RS-M-CLIP	29.21	26.83	37.46	30.16	51.95
	RemoteCLIP_ViT-B-32	52.54	44.60	56.83	52.38	57.67
	GeoRSCLIP_ViT-B-32	66.19	52.38	65.24	66.03	70.05
% Above Baseline		-	**6.67%**	**80.00%**	**40.00%**	**100.00%**
Mean % gain		-	**-6.64%**	**+3.77%**	**+0.13%**	**+8.30%**

A.2.3 Graph collision rate in the original and fine-grained datasets

We analyze the test graphs for both the original and fine-grained datasets, by focusing on the Graph Collision Rate (GCR). To compute the GCR, we first extract all the relation triplets then sort and compute the hash (e.g., SHA256). Next, we count the number of unique hash values. The GCR is computed by dividing the number of duplicates by the total number of samples. A higher value indicates greater redundancy in the dataset, which reflects repeated or non-diverse graph captions. Note that GCR focuses on exact matching. A lower GCR value does not necessarily mean that the dataset is diverse, it means there are fewer **exact** duplicates, but many graphs may still share subgraphs. Table 19 presents the GCR for both the original and fine-grained datasets

Table 19: Graph Collision Rate (GCR) for Original and Fine-Grained Datasets.

Metric	**RSICD**	**FITRS**	**RSITMD**	**SYDNEY**	**NWPU**	**UCM**
Original GCR	30.61	0.00	7.52	48.28	50.58	56.10
Fine-Grained GCR	0.00	0.00	0.00	0.00	0.00	0.00

In the fine-grained datasets, we observe that the GCR is 0 across all datasets, indicating that the graphs generated do not have any exact duplicates, which is a significant improvement compared to the original datasets. To further analyze, we plot the graph length distribution for the number of nodes in the graphs. The distribution of graph lengths is shown in Figure 20. This plot indicates that the fine-grained graphs tend to be larger with more objects and relations, as expected.

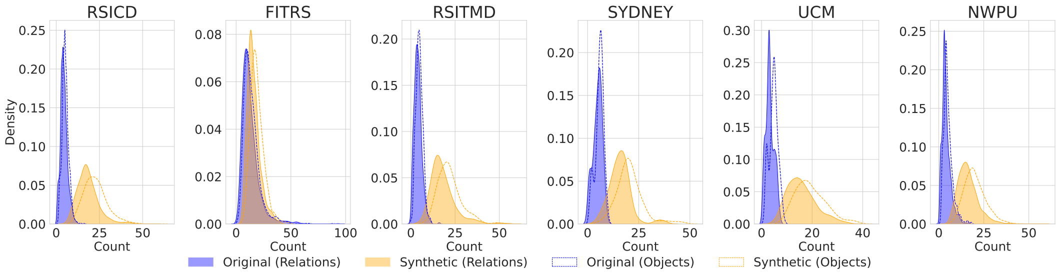


Figure 20: Graph Length Distribution: Comparison between original and fine-grained datasets.

A.2.4 Full results for Scene based Matching

We report the extended table of results, including all datasets, models, and backbones, using the V2V and scene based matching in both the original and fine-grained datasets.

Orginal Datasets

Table 20: Retrieval performance on **original datasets** using V2V and Scene based matching.

Dataset	Model	V2V	V2S		S2V	S2S	
			UC	SAM		UC	SAM
FITRS	RS-M-CLIP	5.33	8.77	8.62	11.27	10.83	11.98
	RemoteCLIP_ViT-B-32	11.01	13.36	12.67	15.07	14.26	13.42
	RemoteCLIP_ViT-L-14	14.09	16.32	15.79	16.50	16.97	16.47
	GeoRSCLIP_ViT-B-32	17.77	20.93	20.96	20.37	19.47	20.15
	GeoRSCLIP_ViT-L-14	19.89	21.65	20.65	21.84	21.71	20.78
RSICD	RS-M-CLIP	58.06	61.63	61.71	61.19	60.74	60.88
	RemoteCLIP_ViT-B-32	30.82	33.84	34.84	33.76	33.78	34.44
	RemoteCLIP_ViT-L-14	34.73	35.72	36.24	37.84	35.45	35.78
	GeoRSCLIP_ViT-B-32	23.94	28.04	27.36	27.64	28.23	27.52
	GeoRSCLIP_ViT-L-14	26.75	29.93	30.04	30.98	31.16	30.52
RSITMD	RS-M-CLIP	73.38	78.91	77.61	77.20	78.29	77.13
	RemoteCLIP_ViT-B-32	47.92	49.40	51.26	51.33	49.06	50.85
	RemoteCLIP_ViT-L-14	48.97	49.53	50.72	52.55	49.46	50.21
	GeoRSCLIP_ViT-B-32	35.41	38.08	38.38	39.24	38.13	38.28
	GeoRSCLIP_ViT-L-14	38.30	40.50	40.61	41.89	40.87	40.46
SYDNEY	RS-M-CLIP	52.76	53.54	55.04	55.15	54.23	55.27
	RemoteCLIP_ViT-B-32	48.28	50.44	51.82	52.28	52.27	53.08
	RemoteCLIP_ViT-L-14	42.18	47.91	46.87	45.49	50.55	48.48
	GeoRSCLIP_ViT-B-32	46.67	50.09	49.29	49.17	51.47	49.17
	GeoRSCLIP_ViT-L-14	49.54	53.08	54.23	53.31	54.00	54.12
UCM	RS-M-CLIP	51.08	56.86	54.54	55.91	57.08	55.24
	RemoteCLIP_ViT-B-32	57.59	60.06	58.95	61.58	59.87	58.92
	RemoteCLIP_ViT-L-14	57.81	58.92	60.22	61.46	59.17	59.71
	GeoRSCLIP_ViT-B-32	49.21	50.54	51.65	53.02	50.67	51.93
	GeoRSCLIP_ViT-L-14	52.26	53.71	54.57	56.19	53.78	53.87
NWPU	RS-M-CLIP	29.03	31.89	31.9	26.24	25.3	24.94
	RemoteCLIP_ViT-B-32	12.54	16.34	16.81	14.07	14.47	14.59
	RemoteCLIP_ViT-L-14	14.58	16.88	17.47	16.16	15.3	15.54
	GeoRSCLIP_ViT-B-32	13.96	17.49	17.73	15.35	15.68	15.31
	GeoRSCLIP_ViT-L-14	14.67	18.26	18.37	15.9	16.59	16.25
% Above Baseline		-	**100.00%**	**100.00%**	**96.66%**	**96.66%**	**96.66%**
Mean % gain		-	**+2.79%**	**+2.95%**	**+3.36%**	**+3.6%**	**+3.51%**

Fine-Grained Datasets

Table 21: Retrieval performance on **fine-grained datasets** using V2V and Scene based matching.

Dataset	Model	V2V	V2S		S2V	S2S	
			UC	SAM		UC	SAM
FITRS	RS-M-CLIP	7.76	10.77	10.95	16.23	18.03	17.31
	RemoteCLIP_ViT-B-32	17.43	17.87	19.03	20.65	20.40	19.65
	RemoteCLIP_ViT-L-14	21.42	21.15	21.99	24.14	22.68	22.46
	GeoRSCLIP_ViT-B-32	41.91	41.17	43.29	34.56	38.83	36.86
	GeoRSCLIP_ViT-L-14	45.03	42.76	44.54	39.73	42.79	40.95
RSICD	RS-M-CLIP	19.49	23.79	23.49	31.78	33.12	31.81
	RemoteCLIP_ViT-B-32	35.16	33.95	34.92	37.06	37.27	37.03
	RemoteCLIP_ViT-L-14	41.93	42.97	44.04	42.30	42.49	41.94
	GeoRSCLIP_ViT-B-32	49.46	52.34	53.04	43.07	48.46	46.51
	GeoRSCLIP_ViT-L-14	53.40	55.57	56.36	45.63	52.21	50.27
RSITMD	RS-M-CLIP	29.87	32.98	33.20	50.24	51.27	49.50
	RemoteCLIP_ViT-B-32	55.68	52.30	55.40	58.57	56.43	56.73
	RemoteCLIP_ViT-L-14	62.54	61.01	63.44	60.71	59.31	58.94
	GeoRSCLIP_ViT-B-32	69.84	70.82	72.29	62.70	65.06	63.88
	GeoRSCLIP_ViT-L-14	71.31	72.07	72.51	65.06	68.82	66.98
SYDNEY	RS-M-CLIP	36.78	39.06	40.21	46.53	46.53	45.95
	RemoteCLIP_ViT-B-32	42.53	47.68	46.53	43.65	51.70	48.25
	RemoteCLIP_ViT-L-14	45.97	46.53	47.68	51.70	52.27	54.00
	GeoRSCLIP_ViT-B-32	57.47	61.47	62.05	56.87	62.62	61.47
	GeoRSCLIP_ViT-L-14	63.22	66.07	67.22	62.05	67.22	67.22
UCM	RS-M-CLIP	29.21	34.63	35.59	48.44	48.44	49.08
	RemoteCLIP_ViT-B-32	52.54	51.77	52.57	59.40	54.32	56.86
	RemoteCLIP_ViT-L-14	59.84	57.65	60.35	62.89	59.71	61.62
	GeoRSCLIP_ViT-B-32	66.19	66.70	67.97	65.74	67.49	69.40
	GeoRSCLIP_ViT-L-14	70.16	71.14	72.57	69.24	73.84	72.57
NWPU	RS-M-CLIP	29.03	31.89	31.9	26.24	25.3	24.94
	RemoteCLIP_ViT-B-32	12.54	16.34	16.81	14.07	14.47	14.59
	RemoteCLIP_ViT-L-14	14.58	16.88	17.47	16.16	15.3	15.54
	GeoRSCLIP_ViT-B-32	13.96	17.49	17.73	15.35	15.68	15.31
	GeoRSCLIP_ViT-L-14	14.67	18.26	18.37	15.9	16.59	16.25
NWPU	RS-M-CLIP	12	14.6	15	17.95	16.87	17.43
	RemoteCLIP_ViT-B-32	19.62	20.42	22.2	20.5	19.93	20.05
	RemoteCLIP_ViT-L-14	25.56	24.8	26.5	24.52	25.59	26.54
	GeoRSCLIP_ViT-B-32	36.06	37.5	38.54	34.03	34.06	33.23
	GeoRSCLIP_ViT-L-14	39.32	39.95	40.89	37.7	39.46	35.42
% Above Baseline		-	**68.00%**	**88.00%**	**53.33%**	**66.66%**	**66.66%**
Mean % gain		-	**+1.03%**	**+2.26%**	**+2.62%**	**+3.31%**	**+2.84%**

Based on Table 20 and Table 21, we can observe the same pattern: across all models and datasets, there is no case where the V2V baseline is the best approach. There is always at least one green cell in each row. We also see that V2S-SAM has the highest percentage above the baseline. However, for S2S-UC, we have a better mean percentage gain (+ 3.31% for fine-grained datasets and +3.6% for the original), even though some values are lower than the baseline. This means that S2S has the highest mean value overall, even if it is influenced by many negative values. Therefore, we can consider S2S as the best approach. From the original datasets, we have more green value in S2S compared to the fine-grained datasets (96.66% vs 66.66%). In the original datasets, captions are more generic, with fewer scenes, making S2S very comparable to V2S. In contrast, in the fine-grained datasets, there are more scene-specific captions, which affects the matching process.