

Quantization of Vision-Language Model for Remote Sensing Visual Grounding

Student Name:

Shamma Alblooshi (100045387)

Suhail Almarzouqi (100045490)

Tasneim Aldhanhani (100048765)

Instructor:

Dr. Hasan Al Marzouqi

Course:

Vision Language Models – COS 794

29 APRIL 2025

Outline

01 Problem Definition and Motivations

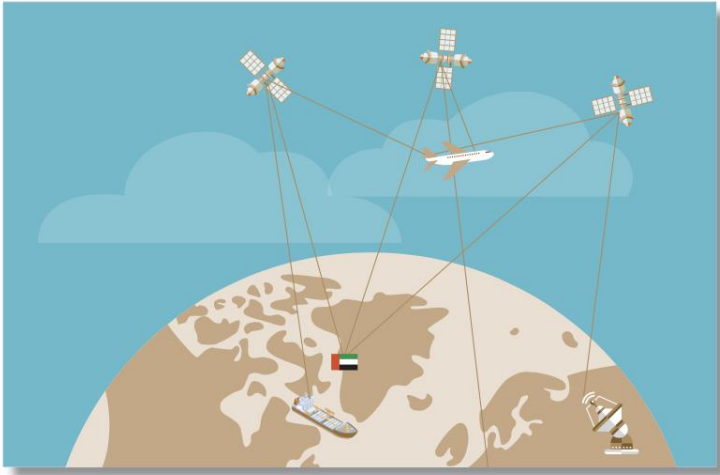
02 Methodology

03 Results and Discussion



Problem Definition and Motivations

Why RSVG?



- Earth observation satellites and UAVs generate massive high-resolution RS imagery.
- Extracting actionable insights is critical for:
 - ✓ Disaster response
 - ✓ Urban planning
 - ✓ Agriculture monitoring
 - ✓ Military applications.
- Visual Grounding enables flexible, open-world object localization from natural language queries beyond static object categories.
- RSVG supports real-time, dynamic scene understanding for diverse remote sensing platforms.

Who Cares

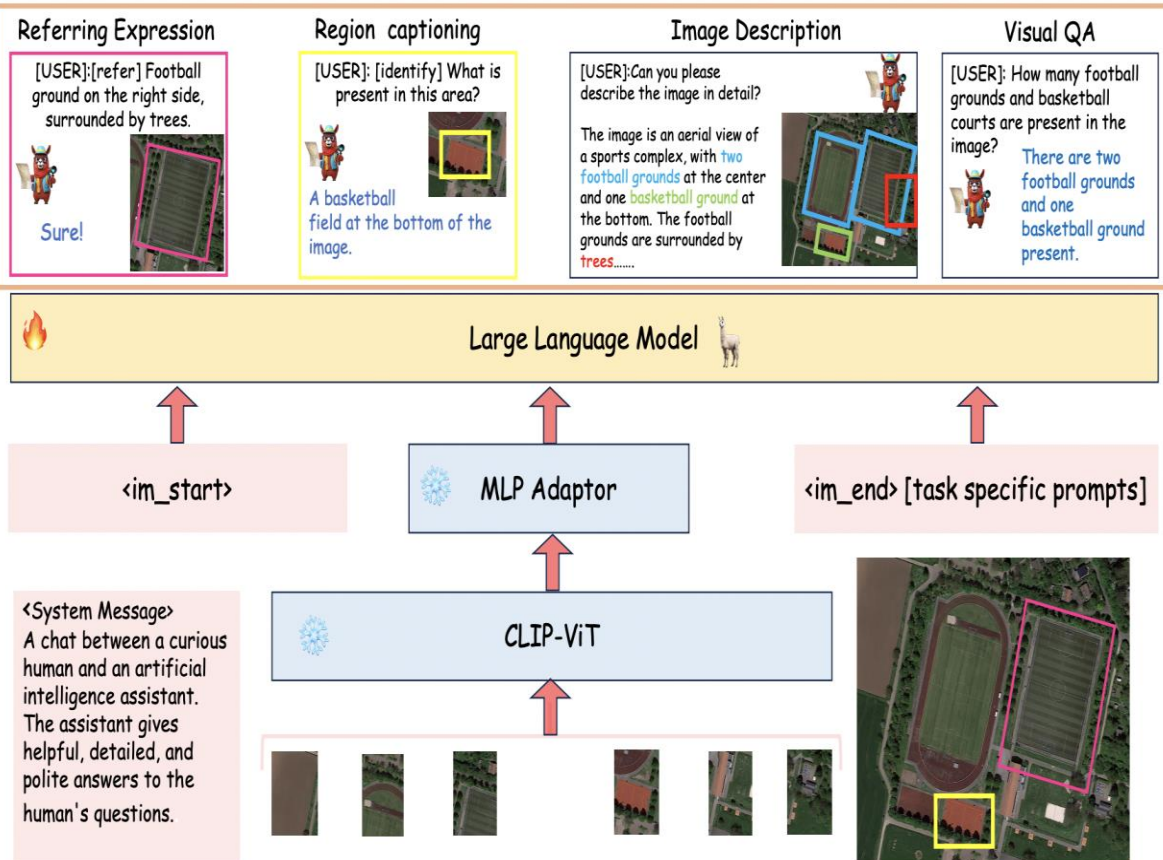


What are we trying to do?

- Quantize and optimize pretrained VLMs to improve computational efficiency for RSVG, without retraining.
 - ❑ Select and benchmark baseline VLMs (e.g., MDETR, CLIP-based detectors).
 - ❑ Integrate quantization techniques (e.g., 8-bit, 4-bit mixed precision) to compress model, speed up inference, and save memory.
 - ❑ Benchmark grounding performance before and after quantization across remote sensing datasets.
 - ❑ Analyze trade-offs between accuracy, inference speed, and model size.
 - ❑ Advance scalable, deployable multimodal AI models for RS imagery understanding.

Methodology

GeoChat Model



- Purpose: Adapt VLMs for remote sensing imagery with: Top-down views, multi-scale targets, cluttered scenes.

GeoChat Model Architecture

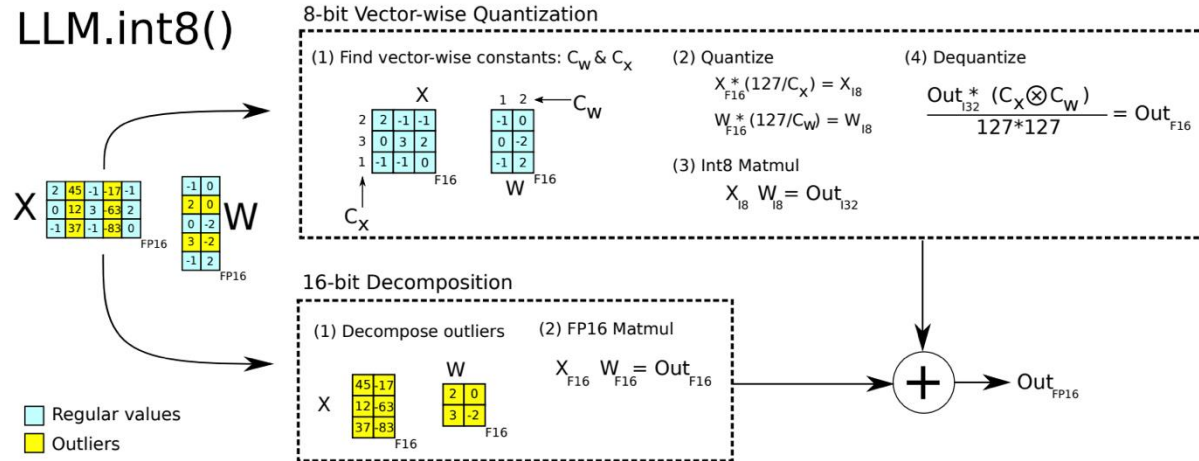
- Built upon LLaVA-1.5 framework, adapted for remote sensing.
- Visual Encoder: CLIP-ViT (enhanced for 504×504 remote sensing images).
- MLP Adaptor: Projects visual features to LLM space.
- Language Model: Vicuna-7B with LoRA fine-tuning for instruction-following.
- Task Tokens: [grounding], [identify], [refer] inserted to control output behavior.
- Spatial Representation: Bounding boxes encoded as normalized text strings (with optional rotation).

Project Objectives

- Quantize the pretrained GeoChat model to achieve efficient, scalable remote sensing visual grounding (RSVG) , improving inference speed, memory footprint, and deployability without retraining.
- Pipeline:
 1. Baseline: Evaluate GeoChat's performance in BF16 precision (Brain Float 16).
 2. 8-bit Quantization: Apply Bits and Bytes asymmetric quantization (LLM.INT8) for model compression.
 3. 4-bit Quantization: Further compress using QVLM custom techniques based on Bits and Bytes.

Quantization Methods

8-bit quantization pipeline



Goal: In a normal 8-bit quantization you pick one single “scale” per tensor, so every element in the weight or activation shares the same mapping into $[-127, 127]$. LLM.int8() instead treats each **row** of the activation matrix and each **column** of the weight matrix as its own mini-tensor, each with its own scale factor.

Main components

8-bit Vector-wise Quantization path

- Compute per-row / per-column scales : absolute-max values per-row (C_X) and per-column (C_W)
- Quantize each row/column independently: multiply the FP16 tensor by $127 / C$ and round to Int8
- MatMul: perform Int8×Int8 → Int32 matrix multiplication
- De-quantize back to FP16: multiply the Int32 result by $(C_X \otimes C_W) / (127 \times 127)$ to recover FP16 accuracy

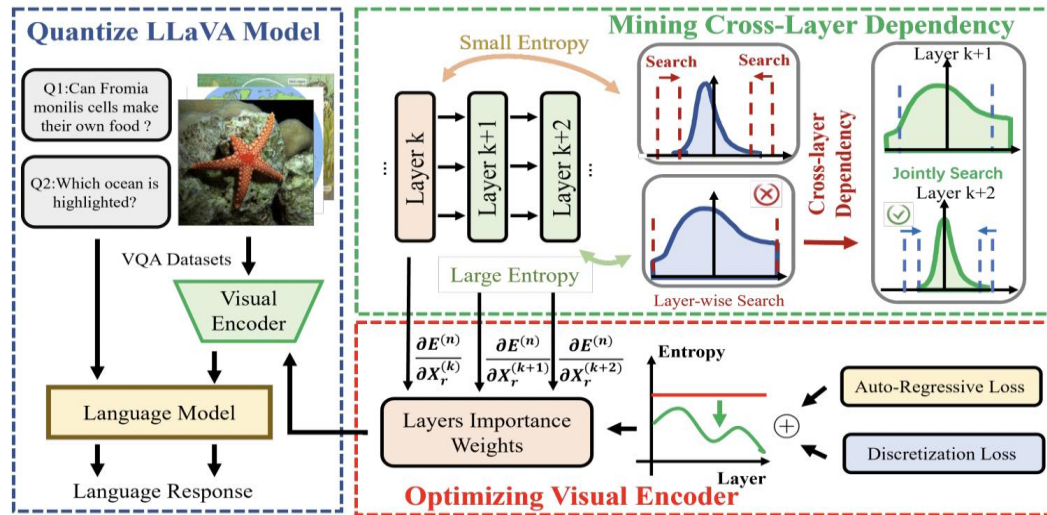
16-bit path

- Identify the small set of feature dimensions whose magnitudes exceed a threshold , denote their indices by O
- Extract from the activation X and weights W just the columns/rows corresponding to O , full-precision (FP16) mat-mul on those

Finally, you **add** the two partial outputs (FP16 from the 16-bit and de-quantized FP16 from the Int8 remainder) to get the full matrix-product result.

Quantization Methods

QVLM pipeline



Main components

Mining Cross-Layer Dependency

- They scan through layers in order. Whenever layer k and $k+1$ both have high entropy (above some threshold T), they **glue** them (and any intermediate high entropy layers) into one joint block. Wherever entropy is low, they allow a "cut" and treat that layer in its own block.
- For each block, search for the best 4-bit clipping and rounding parameters that minimize the block's output error versus its original FP output.

Optimizing the visual encoder

- Auto-regressive loss: Freeze the language model but still compute its usual next-token cross-entropy so that the whole system continues to produce sensible answers.
- Discretization (clipping) loss: encourages its activations to lie in narrow, low-dynamic-range bands (making them easier to quantize).

By back-propagating these objectives alongside the original auto-regressive language loss, the encoder learns to produce features whose quantization errors no longer cascade strongly through the network.

Goal: Convert every weight and activation in a pre-trained LLaVA model from 16-bit (or 32-bit) float down to pure 4-bit integers.

What they do :

- Leave **all** model weights frozen and choose a *small* "calibration set" of (image, question) pair, just a few dozen VQA examples.
- Search for the optimum 4-bit clipping/range parameters per block of layers (not per layer!), so that the difference between the block's 4-bit output and its original floating-point output is as small as possible.

Results and Discussion

Visual Grounding

8-bits RSVG



4-bits RSVG



[grounding] Please describe the image in detail.

In the satellite image, there are **four tennis courts** {<10><70><26><82><90>}{<21><70><37><82><90>}{<30><70><46><82><90>}{<10><80><26><92><90>} located close to each other at the bottom left of the scene. These tennis courts are likely part of a sports complex or a residential area with recreational facilities.

8-bits Visual Grounding

[grounding] Please describe the image in detail.

In the satellite image, there are **two basketball courts** {<22><10><42><26><90>}{<30><10><50><26><90>} located close to each other at the top of the scene. They appear to be adjacent to each other, possibly forming a single sports complex.

4-bits Visual Grounding

8-bits RSVG



4-bits RSVG



[grounding] Please Describe the image in detail.

In the satellite image, there are two **gray Boeing 747 airplanes** {<16><69><30><85><73>}{<30><60><46><76><73>} positioned close to each other at the bottom.

8-bits Visual Grounding

[grounding] Please describe the image in detail.

In the satellite image, there are two **gray Boeing 747 airplanes** {<14><68><28><84><20>}{<30><58><46><74><21>} positioned close to each other at the bottom.

4-bits Visual Grounding

Key Findings

Table 1: Grounding description task.

Model	acc@0.5	acc@0.25
BF16 Model	11.7	33.9
Quantized 8-bits	17.5	41.9
Quantized 4-bits	17.9	41

- **Quantization improved performance:** Reducing model precision introduced slight regularization, stabilizing outputs and improving generalization. As a result, both 8-bit and 4-bit quantized models outperformed the original BF16 baseline.
- **Counting accuracy generally preserved:** 4-bit and 8-bit models correctly counted major objects (e.g., airplanes, basketball courts), though the 8-bit model overcounted tennis courts. Minor scene differences appeared under the same query.
- **Minor localization shifts:** 4-bit models showed slight bounding box looseness but maintained coherent grounding and descriptions.

Results and Discussion

Zero-shot scene classification



Classify the image in the following classes: Rural, Urban

Urban

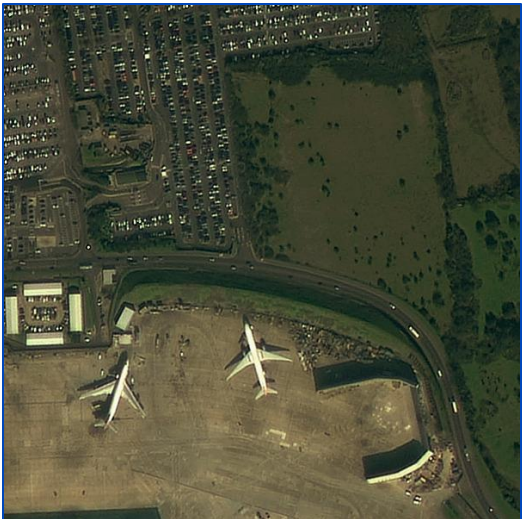
8-bits Scene Classification **[Correct]**

Classify the image in the following classes: Rural, Urban

Rural

4-bits Scene Classification **[Misclassified]**

Zero-shot RSVQA



Where are the airplanes located and what is their type?

The airplanes are located at the bottom of the image, one is a Boeing 747 and the other is a Boeing 777.

8-bits RSVQA **[Full Answer]**

Where are the airplanes located and what is their type?

The airplanes are located at the bottom left of the image. The type of the airplanes is not specified in the image.

4-bits RSVQA **[Partial Answer]**

Key Findings

Table 2: Zero-shot scene classification

Model	UCMerced	AID
BF16 Model	84.43	72.03
Quantized 8-bits	84.00	72.1
Quantized 4-bits	83.95	71.70

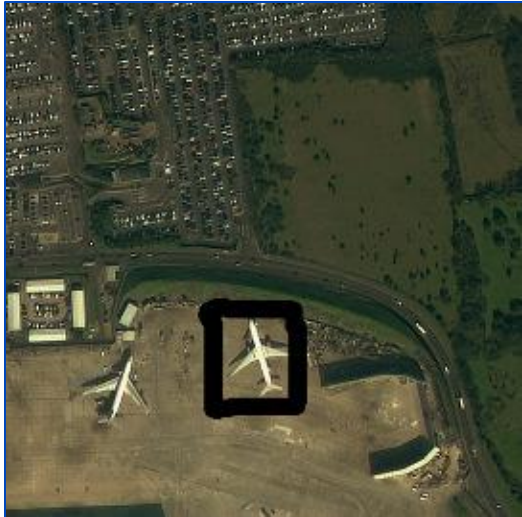
Table 3: Zero-shot RSVQA

Method	Presence	Comparison	Rural/Urban	Avg. Accuracy
BF16 Model	91.09	90.33	94.00	91.80
Quantized 8-bits	90.86	90.00	93.00	91.20
Quantized 4-bits	91.13	90.42	92.00	91.18

- **Scene classification robustness preserved:** 8-bit and 4-bit models maintained near-baseline accuracy on UCMerced and AID, with negligible drops (<0.5%).
- **Minor misclassifications under 4-bit:** While 8-bit models correctly classified urban vs rural scenes, 4-bit models occasionally misclassified under ambiguity, reflecting slight sensitivity at extreme compression.
- **Stable zero-shot RSVQA performance:** 8-bit and 4-bit models matched or slightly exceeded BF16 in question answering tasks.
- **Partial answers at 4-bit:** 4-bit models sometimes omitted detailed types (e.g., "Boeing 747") in answers.
- **Minimal overall impact:** Compression preserved strong zero-shot generalization for both scene classification and RSVQA.

Results and Discussion

Region level captioning



What is the object present here?

The object present here is a Boeing 747 airplane.

8-bits Region Captioning **[Correct]**

What is the object present here?

The object present here is a Boeing 747 airplane.

4-bits Region Captioning **[Correct]**

Key Findings:

- **Correct region-level grounding:** Both 8-bit and 4-bit models correctly captioned the object (Boeing 747 airplane) despite compression.
- **Performance drop after quantization:** Quantized models showed lower ROUGE-1, ROUGE-L, and METEOR scores compared to the BF16 baseline, reflecting reduced fluency and lexical match.
- **Minimal difference between 8-bit and 4-bit:** 4-bit models preserved similar captioning quality to 8-bit, with only slight additional score degradation.
- **Content understanding preserved:** Although textual overlap metrics decreased, the main semantic content (object identity) remained accurately captured across quantization levels.

Future Development:

- **Enhance quantization techniques:** Investigate more advanced quantization strategies to further reduce model size while minimizing performance loss.
- **Address detailed reasoning degradation:** Explore methods such as selective fine-tuning or knowledge distillation to mitigate partial answer generation and improve fine-grained reasoning after compression.
- **Improve spatial grounding precision:** Develop calibration or post-processing methods to reduce minor bounding box shifts observed after aggressive quantization.
- **Adapt quantization for new RS tasks:** Extend quantization experiments to additional remote sensing tasks such as change detection

Table 4: Region level captioning performance

Model	ROUGE-1	ROUGE-L	METEOR
BF16 Model	87.3	87.2	83.9
Quantized 8-bits	73.8	73.8	63.1
Quantized 4-bits	73.3	73.3	62.6

Thank You

Reference

1. Y. Zhan, Z. Xiong, and Y. Yuan, "Rsvg: Exploring data and models for visual grounding on remote sensing data," IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp.1–13, 2023.
2. Y. Zhou, M. Lan, X. Li, Y. Ke, X. Jiang, L. Feng, and W. Zhang, "Geoground: A unified large vision-language model for remote sensing visual grounding," 2025. [Online]. Available: <https://arxiv.org/abs/2411.11904>
3. Hugging Face, "Optimum: A Guide to Quantization," 2024, available at: [texttthttps://huggingface.co/docs/optimum/en/concept_guides/quantization](https://huggingface.co/docs/optimum/en/concept_guides/quantization), Accessed: Feb.10, 2025.
4. M. Grootendorst, "A Visual Guide to Quantization,"<https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-quantization>, 2024, accessed: Feb. 10, 2025.
5. L. Tunstall, L. von Werra, and T. Wolf, Natural Language Processing with Transformers, Revised Edition. O'Reilly Media, 2022. [Online]. Available:<https://www.oreilly.com/library/view/natural-language-processing/9781098136789/>
6. K. Kuckreja, M. S. Danish, M. Naseer, A. Das, S. Khan, and F. S. Khan, "Geochat: Grounded large vision-language model for remote sensing," 2023. [Online]. Available:<https://arxiv.org/abs/2311.15826>
7. C. Wang, Z. Wang, X. Xu, Y. Tang, J. Zhou, and J. Lu, "Q-vlm: Post-training quantization for large vision-language models," 10 2024.
8. T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Llm.int8(): 8-bit matrix multiplication for transformers at scale," 2022. [Online]. Available: <https://arxiv.org/abs/2208.07339>
9. G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 7, pp. 3965–3981, 2017.
10. Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ser. GIS '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 270–279. [Online]. Available: <https://doi.org/10.1145/1869790.1869829>