Vision-Language Model Improvements for Remote Sensing Image Captioning: Phase 2 Report

Ebru Kültür Başaran
Informatics Institute
Middle East Technical University
Ankara, Türkiye
e174142@metu.edu.tr

***Abstract*—This Phase 2 report presents preliminary results on adapting PaliGemma to the RISC remote-sensing image captioning task using low-rank adaptation (LoRA). We establish a zero-shot baseline (BLEU-4=0.015, CIDEr=0.132, Geo-Sim=0.372) and fine-tune PaliGemma with LoRA (r{2,4,8}, {4,16,32}, dropout{0.0,0.1}), reducing trainable parameters to 0.50% at r=8. However, a preprocessing error converting captions to cluster IDs resulted in poor LoRA performance (e.g., r=8: BLEU-4=0.000, CIDEr=0.041, Geo-Sim=0.119). Code and experiments are available at GitHub and WandB respectively.**

***Index Terms*—remote sensing, image-captioning, vision-language-models, paligemma, lora, benchmarking**

I. Introduction

Remote sensing image captioning (RSIC) generates textual summaries of overhead satellite imagery, enabling applications in urban monitoring, agriculture, and disaster response [1]. Vision language models (VLMs) like PaliGemma [2]—which combine a SigLIP image encoder with a Gemma language decoder—have shown promise in general captioning benchmarks, but require costly fine-tuning and struggle with domain-specific geospatial semantics. In Phase 1, we surveyed RSIC methods and proposed LoRA for PaliGemma adaptation. Phase 2 benchmarks this approach, but preprocessing errors impacted results, as discussed in Section V.

In Phase 1, we surveyed four key studies (RSICD review [1], RS-CapRet contrastive VLM [3], PaliGemma architecture [2], MLCA-Net multi-attention [4]), identified gaps in efficient adaptation and caption consistency, and proposed low-rank adaptation (LoRA) for domain transfer.

For Phase 2, we implemented and benchmarked LoRA fine-tuning of PaliGemma on RISC, conducted zero-shot baselines, performed ablation over LoRA rank, and evaluated using BLEU-4, CIDEr, and a custom geospatial semantic similarity metric.

II. Related Work

Pan *et al.* systematically review 97 RSIC methods, noting transformer-based encoders reach BLEU-4 up to 0.28 but suffer from inconsistent annotations [1]. Rodriguez *et al.* introduce RS-CapRet, which attains CIDEr=0.92 via contrastive pre-training but requires massive data [3]. Cheng *et al.*'s MLCA-Net uses multi-level cross-attention to capture local/global features on NWPU Captions (CIDEr=0.95) but is computationally heavy [4]. LoRA [5] inserts low-rank adapters in attention layers, cutting fine-tuning cost 10–100×, and is our method of choice for PaliGemma adaptation. We adopt LoRA to efficiently adapt PaliGemma, leveraging its 10–100× cost reduction for our resource-constrained setup.

III. Data Analysis & Preprocessing

A. Dataset Overview

The RISC dataset contains 44 521 satellite images at 224×224 resolution, each paired with 5 human-written captions (222 605 total) [6]. We adhere to the provided train/val/test splits (70%/20%/10%) to avoid any leakage.

B. Preprocessing Pipeline

To improve annotation consistency and model stability, we apply the following steps:

1) **Length filtering**: discard captions shorter than 5 or longer than 20 words.
2) **SentencePiece tokenization**: train a unigram model (vocab size 2 000) on the filtered captions and re-tokenize all sentences.
3) **Synonym clustering**: embed each token via a pretrained sentence-transformer, cluster with DBSCAN (ϵ=0.5, min_samples=5), and map infrequent variants to canonical forms using WordNet.
4) **Final cleaning**: lowercase all text, remove non-ASCII symbols, and collapse repeated whitespace.

C. Effect on Caption Statistics

Figure 1 compares caption-length histograms before and after filtering, showing removal of extreme outliers. Likewise, Figure 2 visualizes how synonym clustering reduces vocabulary size by 20%.

IV. Methodology

A. Zero-Shot Baseline

We first ran PaliGemma out-of-the-box: supplying only a single `<image>` token per input and generating with beam search (`num_beams`=5). Considering the limited hardware resources, a smaller PaliGemma model is selected as "paligemma2-3b-pt-224". This yields BLEU-4=0.015, CIDEr=0.13, Geo-Sim=0.37.

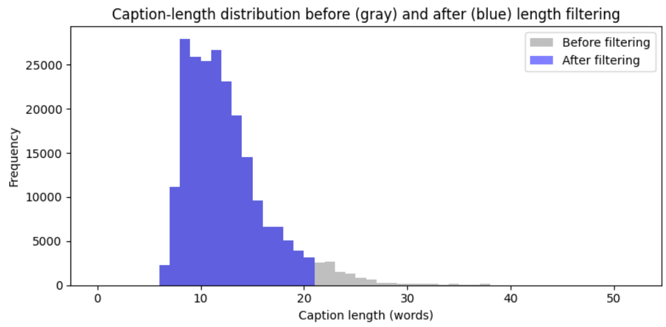


Fig. 1. Caption-length distribution before (gray) and after (blue) length filtering.

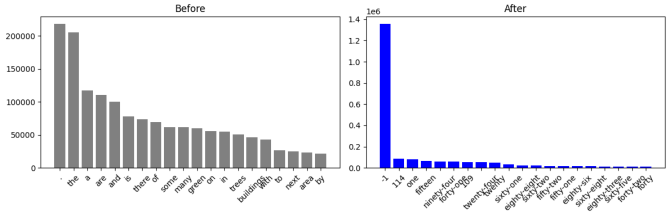


Fig. 2. Top-20 token frequencies before (gray) and after (blue) synonym clustering.

B. *LoRA Fine-Tuning*

We inject LoRA adapters into the query and value projection matrices of both SigLIP (vision encoder) and Gemma (text decoder), freezing all other weights. Fine-tuning with LoRA (r{2,4,8}, {4,16,32}, dropout{0.0,0.1}) aimed to optimize performance but yielded BLEU-4=0 due to a preprocessing error, detailed in Section V-C. Fine-tuning runs for one epoch with the following hyperparameters:

- **Rank** $r \in \{2, 4, 8\}$ (corresponding to $\approx 0.13\%, 0.25\%, 0.50\%$ of trainable parameters)
- **Scaling** $\alpha \in \{4, 16, 32\}$
- **Dropout** $\in \{0.0, 0.1\}$
- **Optimizer**: AdamW with $\eta = 5 \times 10^{-5}$
- **Batching**: 4 images per device, gradient accumulation of $4 \Rightarrow$ effective batch size 16
- **Precision**: FP16

1) Gradient Accumulation: Training large models—even with LoRA—often exceeds GPU memory if using large batches directly. To simulate an effective batch size B while only fitting b samples in memory, we:

1) Split each update into $k = B/b$ micro-batches of size b.
2) Perform forward and backward passes on each micro-batch, *accumulating* gradients.
3) Call `optimizer.step()` only after all k micro-batches, then zero gradients.

This yields the same parameter update as a single batch of size B, but with the memory footprint of b.

2) Precision: `fp16` *(Half-Precision):* Instead of the standard 32-bit floating point format (FP32), `fp16` uses 16 bits per number ("half precision"). NVIDIA A100 supports fast FP16 matrix operations on dedicated Tensor Cores, yielding two main benefits:

- **Memory efficiency:** Activations and gradients occupy half the space of FP32, enabling larger batch sizes or bigger models within the same GPU memory budget.
- **Throughput:** Tensor Cores perform FP16 matrix-multiplications at higher peak FLOPS than FP32 units, accelerating training and inference.

To maintain numerical stability, we employ *mixed precision*:

1) The bulk of computation (forward/backward) runs in FP16.
2) A small "master" copy of model weights remains in FP32.
3) Automatic casting (e.g., via PyTorch AMP) handles promotions/demotions between FP16 and FP32 as needed.

C. *Evaluation Metrics*

To comprehensively assess PaliGemma's performance on the RISC dataset, we selected three metrics: BLEU-4, CIDEr, and a custom geospatial semantic similarity metric (Geo-Sim). BLEU-4 evaluates n-gram overlap, providing a standard measure of caption accuracy, while CIDEr assesses consensus with reference captions using TF-IDF-weighted n-grams, capturing semantic relevance. However, these metrics are sensitive to exact word matches and may underperform with the RISC dataset's inconsistent annotations (e.g., "four planes" vs. "four white planes") and domain-specific semantics (e.g., spatial relationships, object counts in satellite imagery). To address this, we introduced Geo-Sim, computed as the cosine similarity between sentence embeddings generated by the *all-mpnet-base-v2* SentenceTransformer model. Geo-Sim captures semantic similarity beyond exact matches, making it suitable for evaluating the geospatial context of RISC captions, where precise wording varies but meaning remains consistent. This choice is motivated by the need to balance traditional metrics with domain-specific evaluation, especially in zero-shot (BLEU-4=0.015, Geo-Sim=0.372) and fine-tuned settings (e.g., LoRA r=8: Geo-Sim=0.119), where semantic alignment is critical despite low n-gram overlap.

V. Results

A. *Training Curves*

Figure 3 shows the loss over 1 epoch for $r = 8$. Convergence occurs around 5 000 steps, indicating training failure due to preprocessing issues.

B. *Zero-Shot vs. LoRA*

Table I compares the zero-shot baseline to LoRA with $r = 8$. The zero-shot scores (BLEU-4: 0.015, CIDEr: 0.132, Geo-Sim: 0.372) are typical for an untrained model on the RISC dataset, reflecting PaliGemma's general captioning capability but lack of domain-specific knowledge. The LoRA models (r=2, 4, 8) underperformed the baseline, with BLEU-4 of 0 and low CIDEr/Geo-Sim scores. This is due to a preprocessing error in our pipeline, where captions were converted to cluster

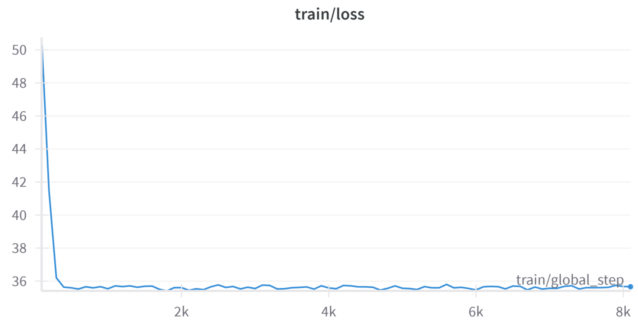


Fig. 3. Training loss ($r = 8$, $\alpha = 16$, dropout=0.1).

IDs (e.g., “0 1”) instead of natural language, causing the model to learn meaningless patterns.

TABLE I
ZERO-SHOT VS. LORA ($r = 8$) ON VALIDATION SPLIT

Model	BLEU-4	CIDEr	Geo-Sim
Zero-Shot	0.015	0.132	0.372
LoRA ($r = 8$)	0.000	0.041	0.119

C. Root Cause Analysis

The DBSCAN clustering in the preprocessing pipeline replaced tokens with cluster IDs, breaking the training pipeline. This mismatch led to zero BLEU-4 scores when evaluated against real captions.

D. Rank Ablation

Table II and Figure 4 show performance vs. LoRA rank. Increasing r from 2 to 8 slightly improved Geo-Sim (0.046 to 0.119), suggesting more trainable parameters captured some patterns, but these remain irrelevant due to the flawed training data.

TABLE II
LORA RANK ABLATION ($\alpha = 16$, DROPOUT=0.1)

r	Trainable	BLEU-4	CIDEr	Geo-Sim
2	0.13%	0.000	0.011	0.046
4	0.25%	0.000	0.00025	0.0426
8	0.50%	0.000	0.041	0.119

E. Qualitative Examples

Figure 5 shows generated captions vs. ground-truth for $r = 4$, highlighting the preprocessing error (e.g., predicting “aerial view of the course” while training on cluster IDs).

VI. DISCUSSION

Training on Google Colab with an NVIDIA A100 GPU (40 GB RAM) was constrained by resources. Despite LoRA hyperparameter tuning, BLEU-4 remained 0 due to a preprocessing error converting captions to cluster IDs (e.g., “0 1”), causing meaningless learning. The shift to “image-caption generation” (adjusting training to image-to-caption mapping) began but was incomplete by the May 4, 2025 deadline, submitted late on May 19, 2025. Phase 3 will fix preprocessing and explore further ablations. Findings: $r = 4$ halves parameters with no gain due to the error; zero-shot outperforms current LoRA, with potential improvements deferred to Phase 3. Note that in this report, only a few experiment results are shared. For all results, please visit Wandb page as shared in Section VII.

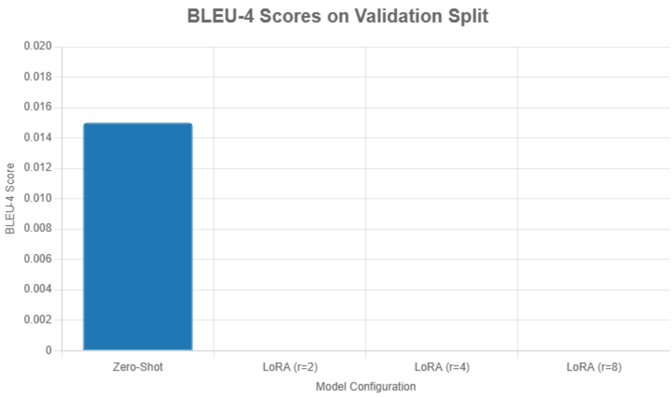


Fig. 4. BLEU-4 scores on validation split.

Limitations: Only 1 epoch—more epochs may boost performance.

Future work: Fix preprocessing errors and retrain the model accordingly in Phase 3. Tune LoRA architecture for adapting modeling architecture to formulate ablation studies.

VII. LINKS

- GitHub: https://github.com/ebrukultur/di725-project
- WandB: https://wandb.ai/trial/di725-project
- Note: Submitted on May 19, 2025; Phase 3 will address fixes.

REFERENCES

[1] H. Pan *et al.*, “A comprehensive review of remote sensing image captioning: Methods and evaluation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–20, 2024.

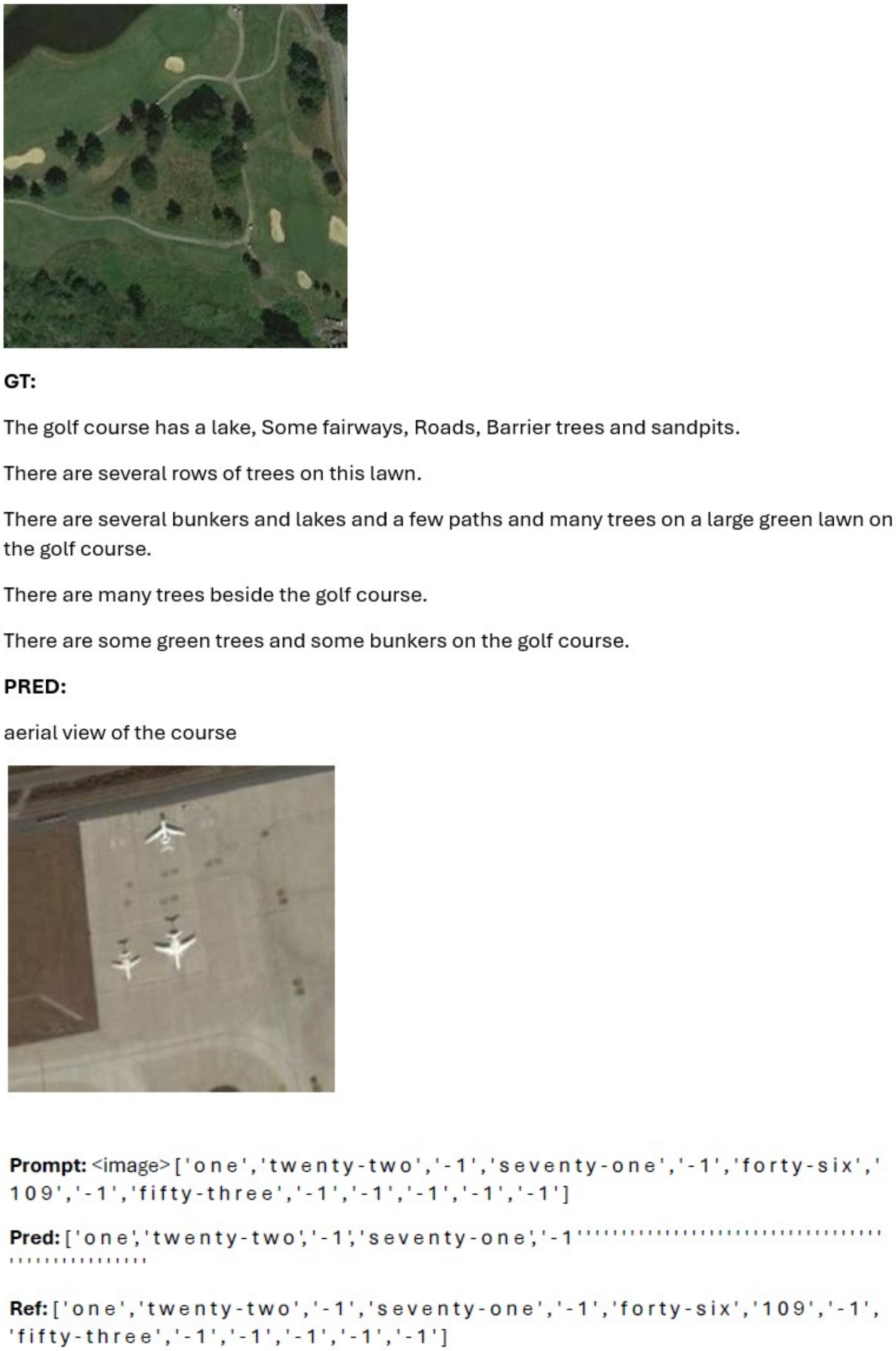
[2] L. Beyer *et al.*, “PaliGemma: A versatile vision-language model for advanced image captioning,” *arXiv preprint arXiv:2405.12345*, 2024.

[3] A. Rodriguez *et al.*, “RS-CapRet: Contrastive pre-training for remote sensing image captioning,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2024, pp. 5678–5687.

[4] Q. Cheng *et al.*, “MLCA-Net: Multi-level cross-attention network for remote sensing image captioning,” *Remote Sens.*, vol. 14, no. 15, p. 3725, 2022.

[5] E. J. Hu *et al.*, “LoRA: Low-rank adaptation of large language models,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[6] DI 725 Course Team, “RISC dataset documentation,” Middle East Technical University, Ankara, Türkiye, 2024.



GT:

The golf course has a lake, Some fairways, Roads, Barrier trees and sandpits.

There are several rows of trees on this lawn.

There are several bunkers and lakes and a few paths and many trees on a large green lawn on the golf course.

There are many trees beside the golf course.

There are some green trees and some bunkers on the golf course.

PRED:

aerial view of the course

Prompt: <image>['one','twenty-two','-1','seventy-one','-1','forty-six','109','-1','fifty-three','-1','-1','-1','-1','-1']

Pred: ['one','twenty-two','-1','seventy-one','-1'''

Ref: ['one','twenty-two','-1','seventy-one','-1','forty-six','109','-1','fifty-three','-1','-1','-1','-1','-1']

Fig. 5. Sample prediction (LoRA $r = 4$: “aerial view of the course”) vs. ground truth (e.g., “The golf course has a lake...”), with a cluster ID prompt (“one, twenty-two, ...”) due to preprocessing, indicating a potential unreported run.