PeFoMed-Light: A Lightweight and Parameter-Efficient Vision–Language Architecture for Automated Medical Report Generation

Medical Report Generation using PeFoMed with LLM-based Text Reports from CheXpert Dataset

Presenters:

Ch. Himesh Raj Kumar (CS23BT005)

K. Varaprasad Reddy (CS23BT050)

Pasham Pardeev Narsi Reddy(CS22BT040)

A research project on improving automated medical report generation using parameter-efficient fine-tuning and large-scale, region-based datasets.

Deep Learning Medical AI Report Generation

CONTENTS

- Abstract
- Introduction
- Methodology
- Experiments and Results
- Evaluation Metrics
- Conclusion and Future work
- References

Abstract

- The project focuses on automating medical report generation from chest X-ray images using the PeFoMed model.
- PeFoMed was fine-tuned on a custom dataset derived from CheXpert.
- Llama was used to generate text reports corresponding to the 14 diagnostic labels in the dataset.
- The model was trained with:
  - Chest X-ray images as input, and
  - LLM-generated text reports as output.
- The main goal was to evaluate PeFoMed’s ability to generate accurate medical reports from image inputs.

Introduction

- **PeFoMed** is a **transformer-based multimodal model** designed for **medical report generation**.
- Traditional medical report generation models are **limited by high data annotation costs**.
- To overcome this, **LLM-generated synthetic text reports** aligned with **CheXpert labels** are used to **create a scalable training dataset** for PeFoMed.
- The project aims to:
    - **Verify** whether LLM-generated medical text **accurately aligns with ground truth reports**.
    - **Evaluate** model performance using **standard NLP metrics**.

What is PeFoMed? and Why PeFoMed?

- PeFoMed is a transformer-based multimodal generative model designed for medical imaging tasks like Visual Question Answering (VQA) and Medical Report Generation (MRG).
  It follows a parameter-efficient fine-tuning (PEFT) strategy so that only a small part of the model is trained, while the rest stays frozen.

- **Architecture Overview-**
- **Vision Encoder:** A pre-trained ViT (EVA) model extracts features from X-ray images (kept *frozen*).
- **Projection Layer:** Maps visual features into the LLM embedding space.
- **LLM Backbone:** A frozen LLaMA-2-Chat model, acting as a text generator.
- **LoRA Layers:** Lightweight trainable adapters inserted into the LLM to enable efficient fine-tuning.

- **Why PeFoMed?**
- Requires very little training compute
- Maintains strong generative capability
- Works well even with limited medical datasets
- Proven effective on VQA-RAD, SLAKE, PathVQA, IU-Xray

Text Generation Pipeline

- We generate Findings section from CheXpert labels using LangChain + LLaMA-7b with few-shot prompting

- **Input:** CheXpert label vector (14 labels)
- **Output:** Radiology-style Findings text.

- **Pipeline Flow:**

1.Load CheXpert labels
2.Convert them into a natural language prompt
3.LLaMA-7b generates the Findings section
4.Output checked manually + validated using CheXbert

- **Advantages:**

-Zero hallucination observed in manual check
-Radiologist-like tone preserved
-Works consistently across normal and abnormal cases

Why We Modified the Original PeFoMed Architecture

Problem in Original PeFoMed:

- The PeFoMed paper uses EVA-ViT-g Vision Encoder (a very large ViT model),its not publicly available and requires huge GPU resources.
- EVA produces visual embeddings that are already aligned to 5632 dimensions, which match LLaMA hidden size requirements.
- The original PeFoMed uses a large frozen LLaMA-Chat model for text decoding,very high VRAM (>40GB).

Resolvment of Issue in Our Implementation:

- EVA-ViT is not open-source, so we replaced it with: CLIP ViT-L/14 Vision Encoder (OpenAI)
- But CLIP produces: token embedding size = 1024, We take 4 tokens : 4 × 1024 = 4096 dimensions(5632 for EVA-ViT)
- Our Solution: To replicate the original structure: We added an extra trainable projection layer:
- CLIP output (4096)->->->(up_proj)->->->5632->->->(image_proj)->->->4096(LLaMA hidden embedding size)

Contd..

•But, Due to limited hardware resources, we were unable to load or fine-tune the complete LLaMA decoder so used a smaller text decoder(GPT-2)

GPT-2 lightweight decoder generates reports:

•Fully compatible with our low-resource hardware

•Entire pipeline trained **from scratch for our dataset**

•Base GPT-2 model weights remain **frozen**

•Only **LoRA adapters** are trained (parameter-efficient fine-tuning)

Final Result

- We successfully maintain PeFoMed architecture logic
- But with a fully open-source vision encoder (CLIP) and open source text decoder (GPT-2)
- And minimal trainable parameters (up_proj+image_proj+LoRA)

Final Model Pipeline (Modified: CLIP + Projection + GPT-2 Text Decoder)

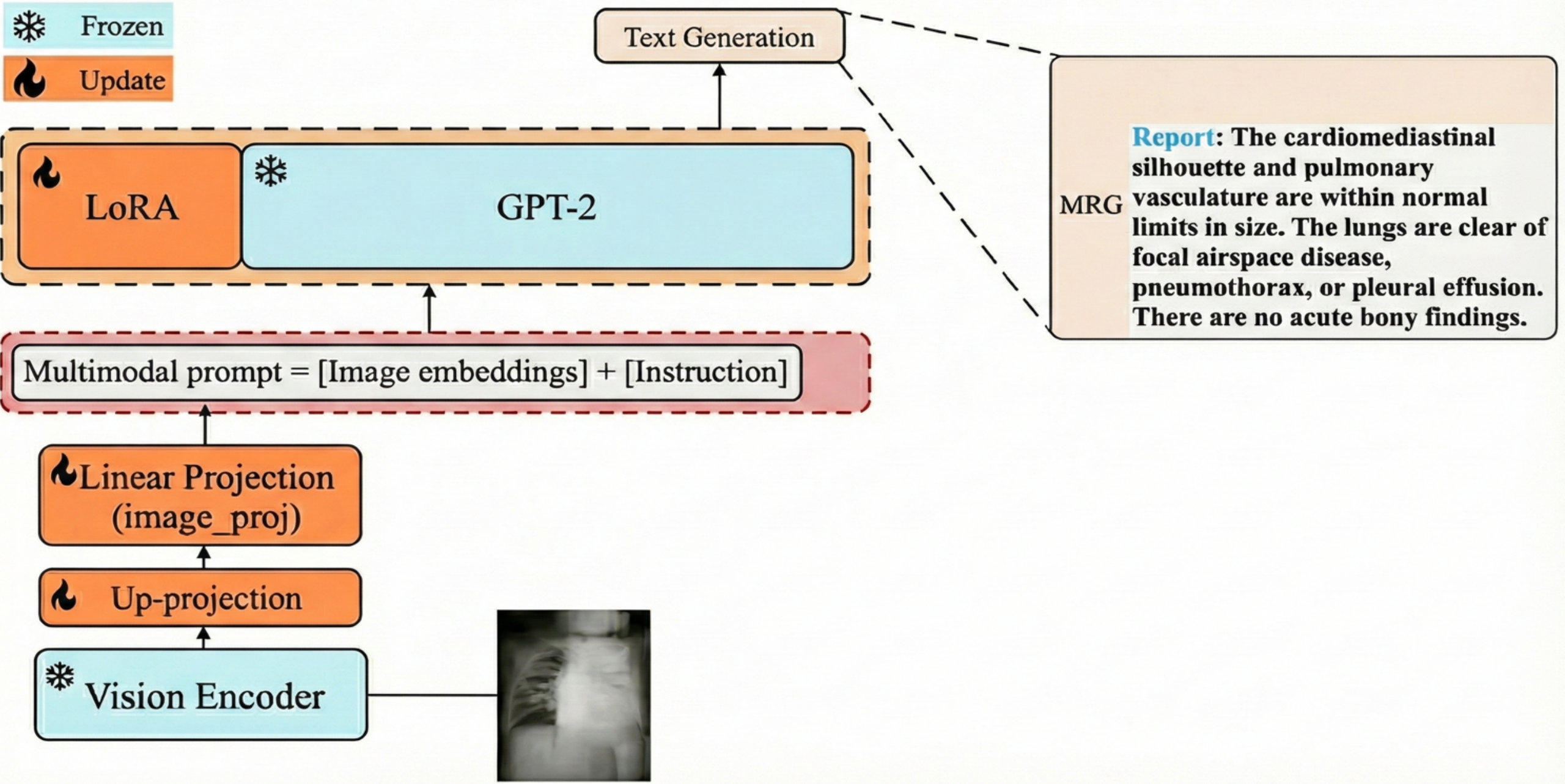
1. Image Processing
   - Input image goes into CLIP ViT-L/14 (frozen).
   - Outputs 4096-dim visual embedding.

2. Visual Feature Alignment
   - Up-projection layer: 4096 → 5632 (trainable).
   - Image-projection layer: 5632 → 4096 (trainable).

3. Multimodal Fusion
   - Aligned visual tokens are injected into the GPT-2 decoder.

4. Text Generation
   - GPT-2 generates answers/reports.
   - Base GPT-2 weights frozen.
   - Only LoRA adapters are trained.

5. Efficiency
   - Only projection layers + LoRA are trainable.
   - Achieves parameter-efficient fine-tuning.


Frozen
Update
Text Generation
LoRA
GPT-2
Multimodal prompt = [Image embeddings] + [Instruction]
Linear Projection (image_proj)
Up-projection
Vision Encoder
MRG
Report: The cardiomediastinal silhouette and pulmonary vasculature are within normal limits in size. The lungs are clear of focal airspace disease, pneumothorax, or pleural effusion. There are no acute bony findings.

Mathematics Behind PeFoMed

- **Visual Embedding (ViT EVA)**:Chest X-ray image → split into patches → patch embeddings E_{img}

$$E_{img} = \mathrm{ViT}(X)$$

- **Token Grouping:**4 consecutive visual tokens are concatenated to reduce computation.

$$G_i = \mathrm{Concat}(E_{4i}, E_{4i+1}, E_{4i+2}, E_{4i+3})$$

- **Projection into LLM Space:**A single linear layer maps visual features into GPT-2 hidden space.

$$Z_i = W_p G_i + b_p$$

- **LoRA Update Rule (Parameter-Efficient Fine-Tuning):**Instead of updating the full weight W,LoRA learns low-rank matrices- $W' = W + BA, \quad \text{where } A \in \mathbb{R}^{r \times d}, B \in \mathbb{R}^{d \times r}$

- **Prompt Fusion:**Multimodal prompt = [Image embeddings] + [Task token] + [Instruction]

$$T = [Z_1, Z_2, ..., Z_n, [\mathrm{REPORT}], \mathrm{Instruction}]$$

- **Autoregressive Text Generation:** GPT-2 predicts next token

$$y_t = \arg\max P(y_t | T, y_{<t})$$

Methodology

Our approach uses the PeFoMed framework, a parameter-efficient method for fine-tuning multimodal large language models for medical report generation.

Dataset Preparation

- Base dataset: CheXpert (14 medical condition labels).
- Each image-label pair was used as input to LLaMA to generate descriptive text reports.
- Verified 100 randomly selected samples manually — the generated reports were accurate, label-consistent, and hallucination-
So proceeded with generating for around 13k samples

Data Split

- Total samples: 10868(after cleaning)
- Split into Train: 9761, Validation: 907, Test: 200

Model Training

- Input: X-ray image
- Output: LLM-generated text report
- Trained for 3 epochs using the custom dataset.

Experimental Design

We conduct a comparative study to evaluate the impact of large-scale, multigranular pre-training on medical report generation quality.

Implementation Details

- Frameworks:PyTorch, HuggingFace Transformers
- Hardware Apple M-series GPU (MPS backend)
- Training Duration\~ 30 mins per epoch processed efficiently
- LLM Used LLaMA for text generation
- PeFoMed Checkpoint Loaded pretrained weights and fine-tuned for 3 epochs

Evaluation Metrics:

- **ROUGE-L —** measures textual overlap between generated and reference reports.
- **METEOR —** evaluates precision, recall, and alignment of phrases.
- **CIDEr —** measures consensus between generated reports and human-written reports.
- **Exact Match (EM)** — checks if generated reports exactly match reference reports.

Results

Training Logs

Epoch 1:Avg Loss: 0.9069
Validation Metrics: ROUGE-L 0.1447, METEOR 0.1146, CIDEr 0.0966, EM 0.0000

Epoch 2:Avg Loss: 0.4718
Validation Metrics: ROUGE-L 0.1589, METEOR 0.1042, CIDEr 0.1003, EM 0.0000

Epoch 3:Avg Loss: 0.4718
Validation Metrics: ROUGE-L 0.1712, METEOR 0.1180, CIDEr 0.1125, EM 0.0000

- **Report Completeness**: More comprehensive coverage of all clinically relevant findings.

Evaluation Results

ROUGE-L 0.129
METEOR 0.095
CIDEr 0.173
EM 0.000

Observations

Across three epochs, the model showed consistent improvements in ROUGE-L, METEOR, and CIDEr. This indicates that the multimodal alignment and LoRA fine-tuning strategy successfully learned image-conditioned report generation.

Comparison with Prior Medical Report Generation Models

Method	Type	METEOR ↑	ROUGE-L ↑	CIDEr ↑
R2Gen [3]	Non-LLMs	0.211	0.377	0.438
BiomedGPT [54]	LLMs	0.146	0.302	0.360
PeFoMed (paper)	LLMs	0.157	0.286	**0.462**
PeFoMed-Light (Ours)	LLMs	**0.095**	**0.129**	**0.173**

:hetic

ed reports,

on-free, and

Method	Type	METEOR ↑	ROUGE-L ↑	CIDEr ↑
R2Gen [3]	Non-LLMs	0.211	0.377	0.438
BiomedGPT [54]	LLMs	0.146	0.302	0.360
PeFoMed (paper)	LLMs	0.157	0.286	**0.462**
PeFoMed-Light (Ours)	LLMs	**0.095**	**0.129**	**0.173**

Conclusion and Future work

The LLM-generated reports from LLaMA proved highly reliable and aligned with the 14 CheXpert labels. PeFoMed successfully learned to map X-ray images to meaningful clinical text.

Future work includes:

- Expanding dataset size
- Training on llama as decoder (if resource provided)
- Replacing vision encoder with Vmamba
- Adding MoE on text decoder and vision encoder part

References

- **JinlHe, PeFoMed**: Medical Report Generation Framework
- **GitHub**: [https://github.com/jinlHe/PeFoMed]
- **CheXpert Dataset** — Stanford ML Group
- **Llama LLM** — Text generation for medical data augmentation
- **GPT-2 — Text decoder, [https://GPT-2-v2.github.io]**
- **CLIP ViT-L/14 — Vision encoder, [https://arxiv.org/pdf/2103.00020]**