WOA7015 Advanced Machine Learning
1/2025/2026

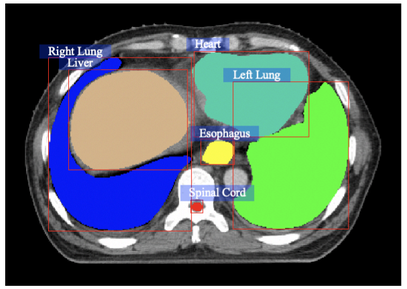
FROM CNN-LSTM TO VLM:
A COMPARATIVE ANALYSIS ON MED-VQA WITH SLAKE

PREPARED BY:
TEAM: JHERNG
HONG JIA HERNG - U2005313

LECTURER:
PROF. IR. DR. CHAN CHEE SENG

1 - Background

Why Med-VQA?



Knowledge-based:
(En) What is the function of the rightmost organ in this picture?
(Zh) 图中是否有器官属于呼吸系统?
(Are there organs in this image belonging to the respiratory system?)

Vision-only:
(En) Does the image contain left lung?
(Zh) 这张图片是关于腹部吗?
(Is this image about the abdomen?)

- Medical Visual Question Answering (Med-VQA) enables models to answer **image-grounded clinical questions**
- Useful for:
  - clinical decision support
  - improving interpretability
  - assisting radiology workflows

Why compare CNN-LSTM vs VLM?

- Traditional multimodal models treat Med-VQA as **discriminative** problem (classification)
- Vision-Language Models treat Med-VQA as **generative** problem (text generation)
- Goal:
  - understand performance gaps + tradeoffs
  - benchmarks both approaches on SLAKE (Liu et al., 2021)

2 - Objectives

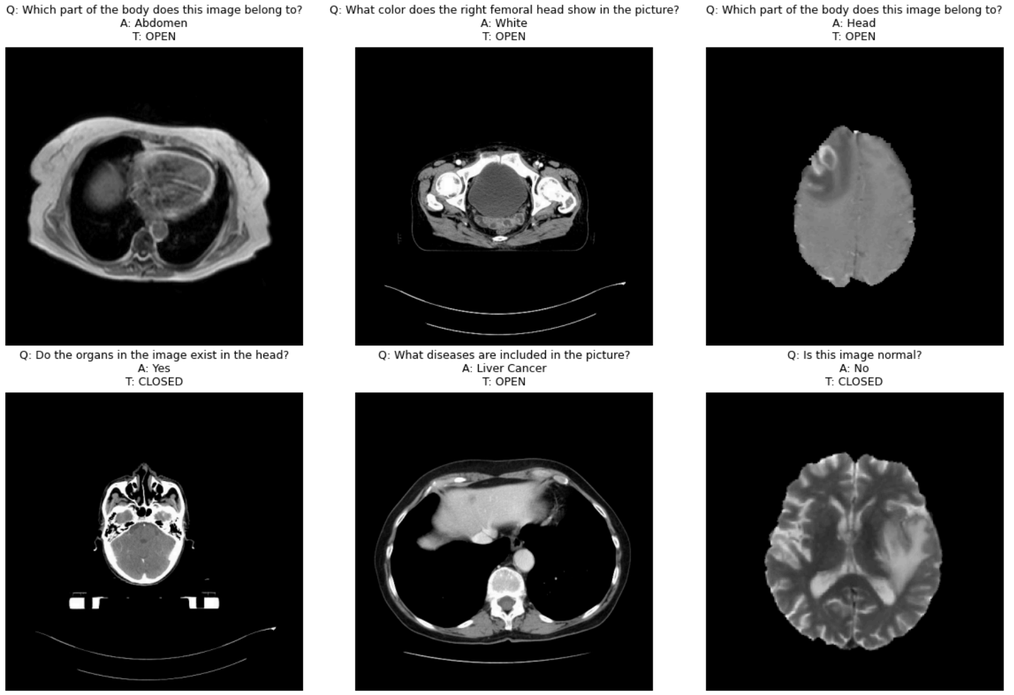
- To build and evaluate a classical multimodal baseline using a CNN for image encoding and an LSTM for question encoding.
- To benchmark a modern pretrained Vision–Language Model (BLIP-VQA) on the same dataset.
- To compare strengths, limitations, and performance of both approaches on the SLAKE Med-VQA dataset.
- To identify which modelling approach is more suitable for medical VQA problems at small data scales.

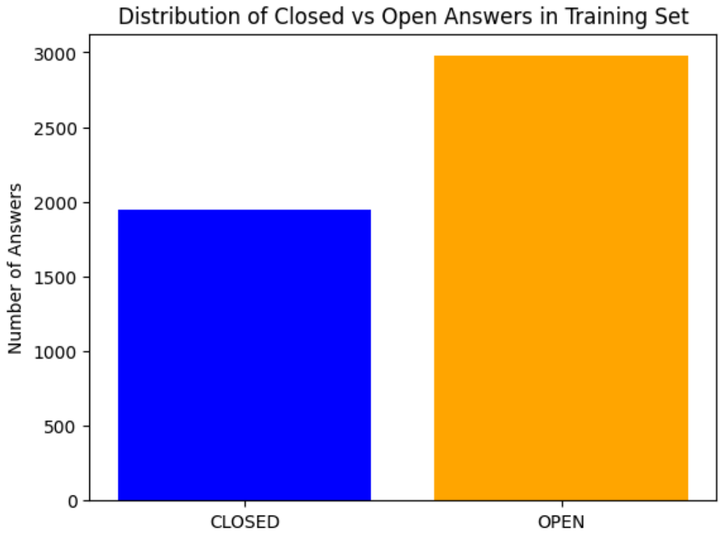
3 - EDA

Dataset Overview: SLAKE

- SLAKE (Semantically-Labeled Knowledge-Enhanced)
  - Medical images (e.g., CT, MRI, X-ray, etc.)
  - Both closed-ended questions (Yes/No, modality type, organ identification)
  - And open-ended factual questions (e.g., “What organ is enlarged?”)
  - *This study uses **English** question-answer pairs only.*

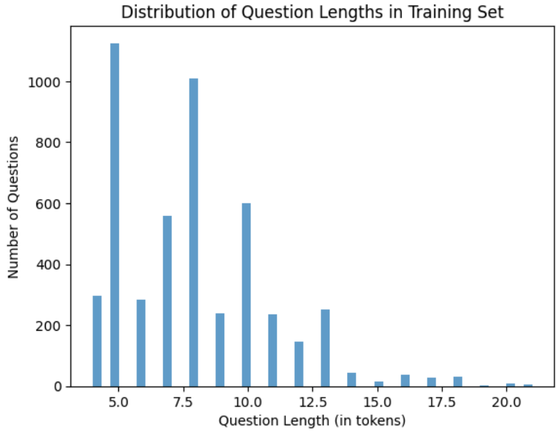
- Answer Type (English Only):
  - CLOSED answers: 1943 (39.50%)
  - OPEN answers: 2976 (60.50%)



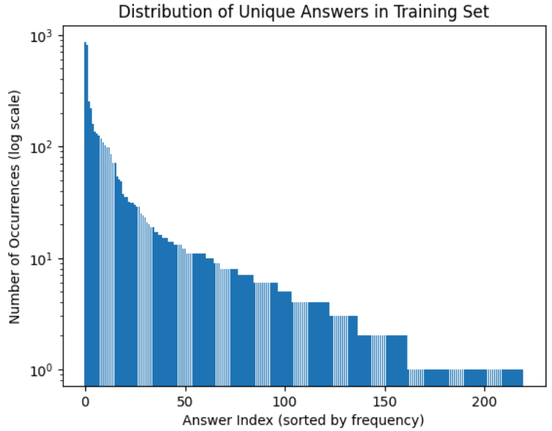


3 - EDA Highlights

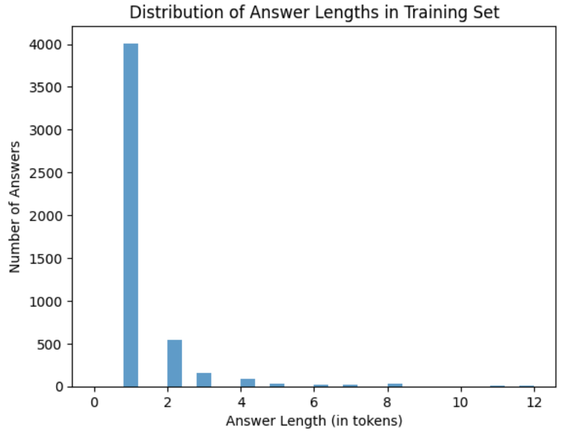
1. Question vocabulary size: 290
   - All unique tokens/words in train.



2. Max question length: 21 → 32 tokens (word-level tokenization for CNN-LSTM).
   - Why bump to 32?
     - Protect against slightly longer questions



3. Answer space for classifier (221 classes):
   - Top-K answers (K=220, i.e., all answers in train) + <unk> for unseen/rare answers.
4. Long-tailed answer distribution impacts macro-F1 and rare class performance.

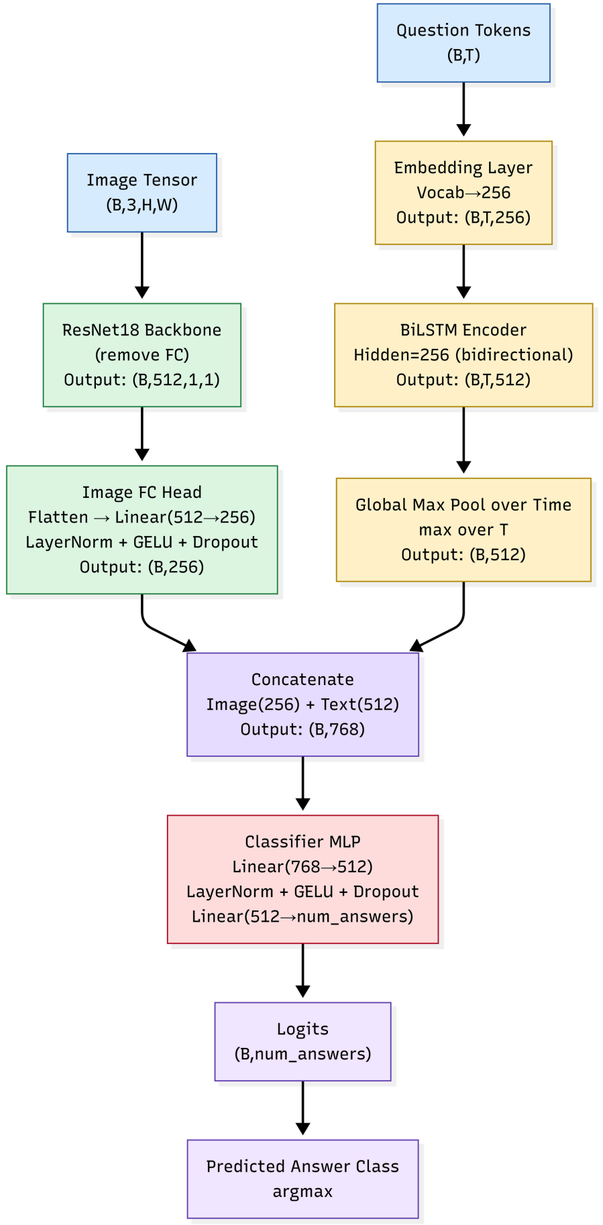


5. Max answer length: 12 (Useful info for VLM)

4 - Methods

CNN-LSTM (Discriminative)

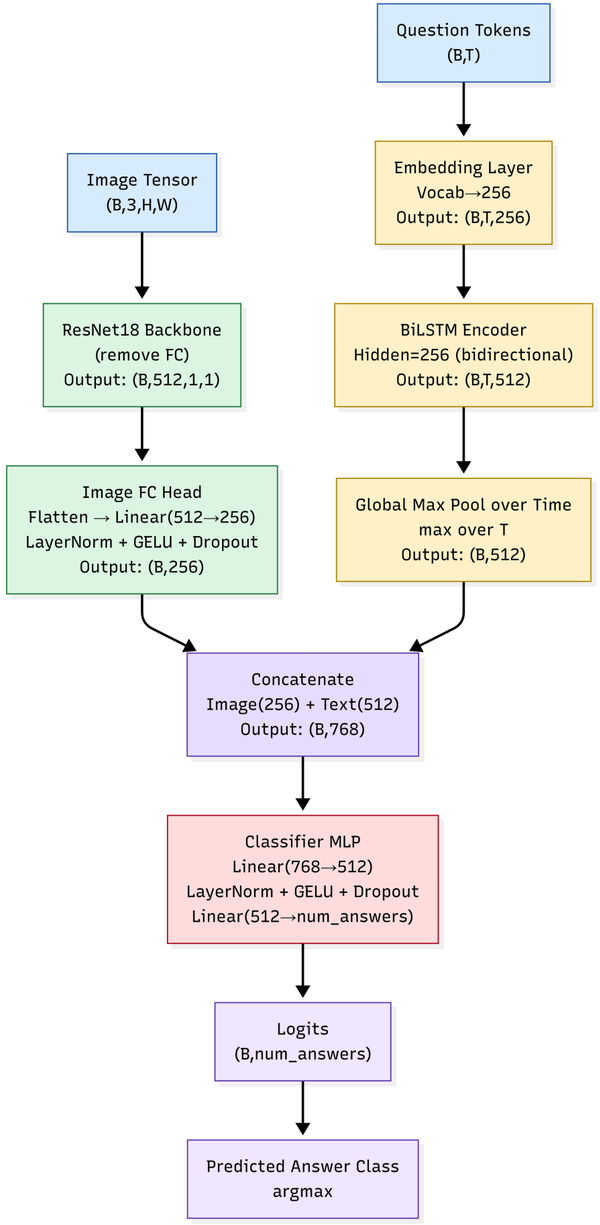
- **Formulation**
  - Input: image + question
  - Output: answer class (Top-K unique answers)
    - K=221, where 220 is #unique answers, 1 is <unk>
  - Loss: cross-entropy
- **Architecture**
  - CNN image encoder (frozen vs fine-tuned)
    - Using ResNet-18
      - Hypothesis: Small enough, won’t overfit too bad on SLAKE
  - LSTM question encoder
  - Fusion + MLP classifier head



4 - Methods

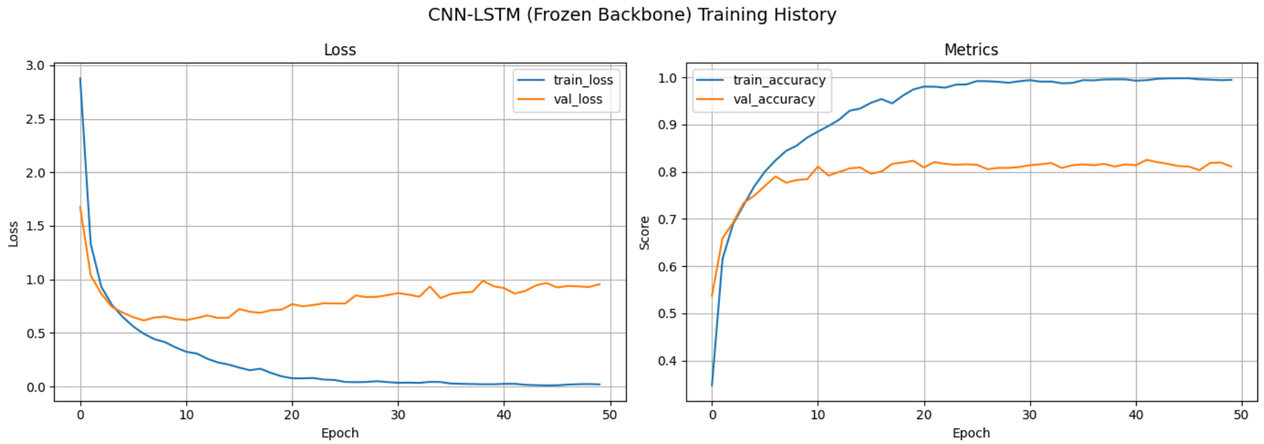
CNN-LSTM Training Setup

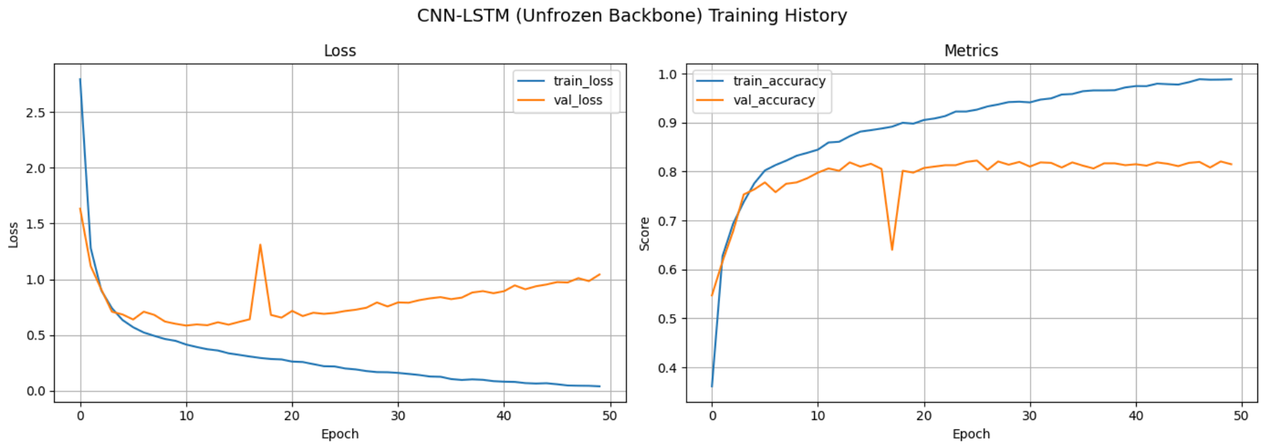
- **Preprocessing**
  - Image resize + normalization
  - Tokenize question → fixed length 32 (vocab size = 290)
  - Answer classification: 220 + <unk>
- **Training**
  - Train 2 variants, frozen backbone & full fine-tuning
  - Took ~12 mins each on A100 80GB
  - Configs:
    - Optimizer: Adam (lr=1e-3)
    - Epoch: 50
    - Model selection metric: Val Accuracy
    - Batch size: 128
  - Parameters:
    - Frozen backbone:
      - Total params: 12,943,901, Trainable params: 1,767,389
    - Unfreezed backbone:
      - Total params: 12,943,901, Trainable params: 12,943,901

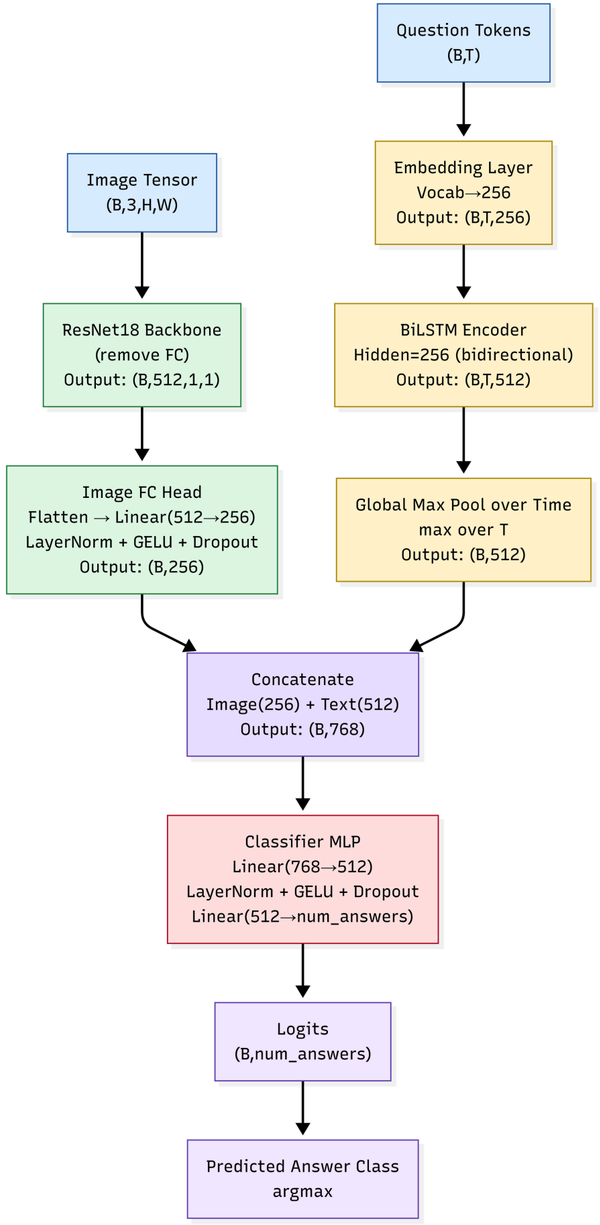


4 - Methods

CNN-LSTM Training History







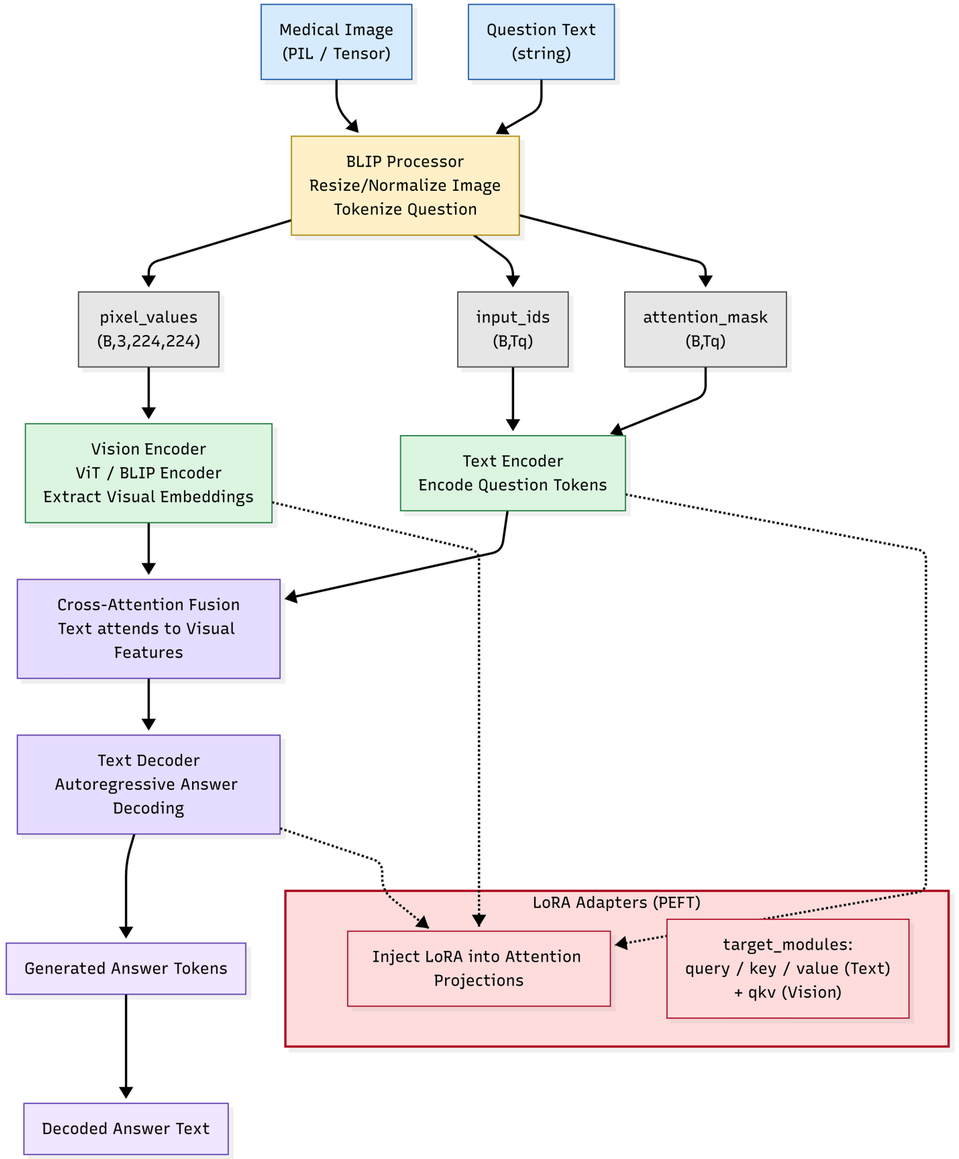
4 - Methods

BLIP (Generative) + LoRA

- **Formulation**
  - Input: image + question
  - Output: generated text answer
  - Loss: Language Modeling (LM) loss
    - Cross entropy in autoregressive fashion
    - Details hidden in transformers library

- **Why BLIP + LoRA?**
  - BLIP is lightweight compared to BLIP-2
  - LoRA fine-tunes efficiently:
    - fewer trainable params
    - faster training + lower GPU cost

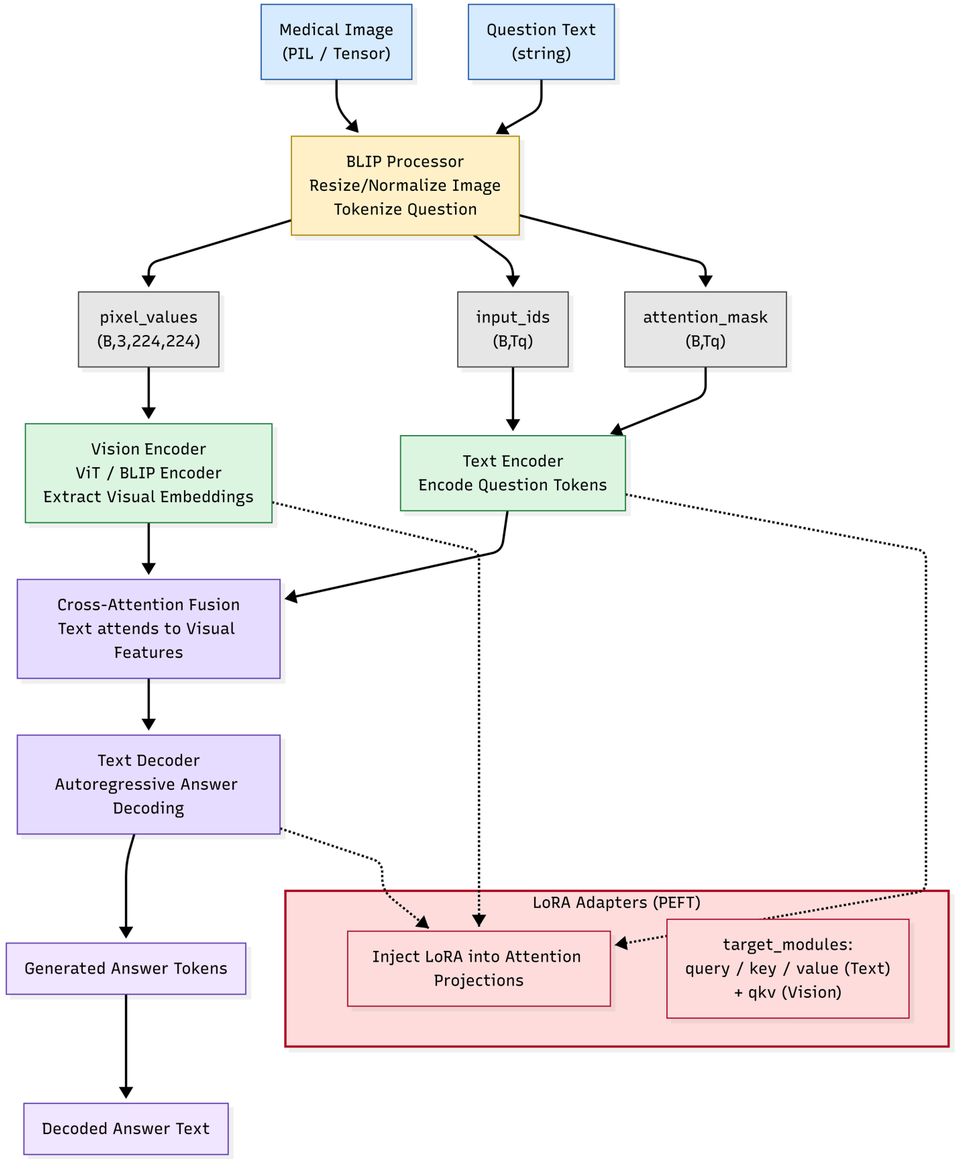
- **Adaptation Strategy**
  - Pretrained BLIP-VQA backbone
  - LoRA fine-tuning on attention projections
    - Text decoder: ***query, key, value***
    - Vision encoder: ***qkv***



4 - Methods

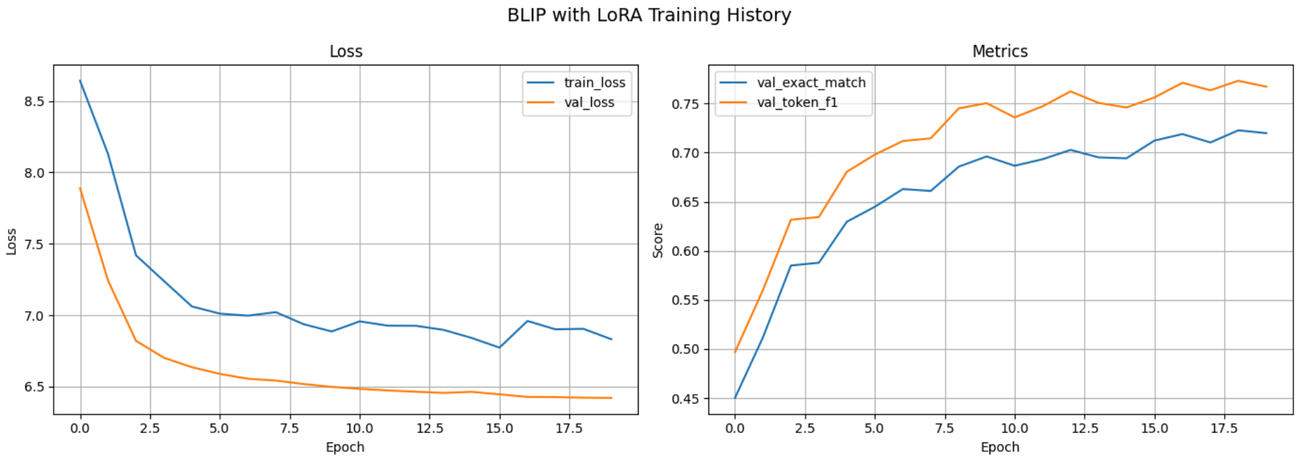
BLIP + LoRA Training Setup

- Preprocessing
  - Built-in processor handles image + text tokenization/padding.
- Training
  - Took ~52 mins each on A100 80GB
  - Configs:
    - Optimizer: AdamW (lr=1e-4)
      - standard choice for larger model like BLIP, mo stable training
    - Epoch: 20
    - Model selection metric: Val Token F1-Score
    - Batch size: 64 (GPU usage ~36/80 GBs, we can do more)
  - Parameters:
    - Total params: 363,294,524, Trainable params: 2,064,384

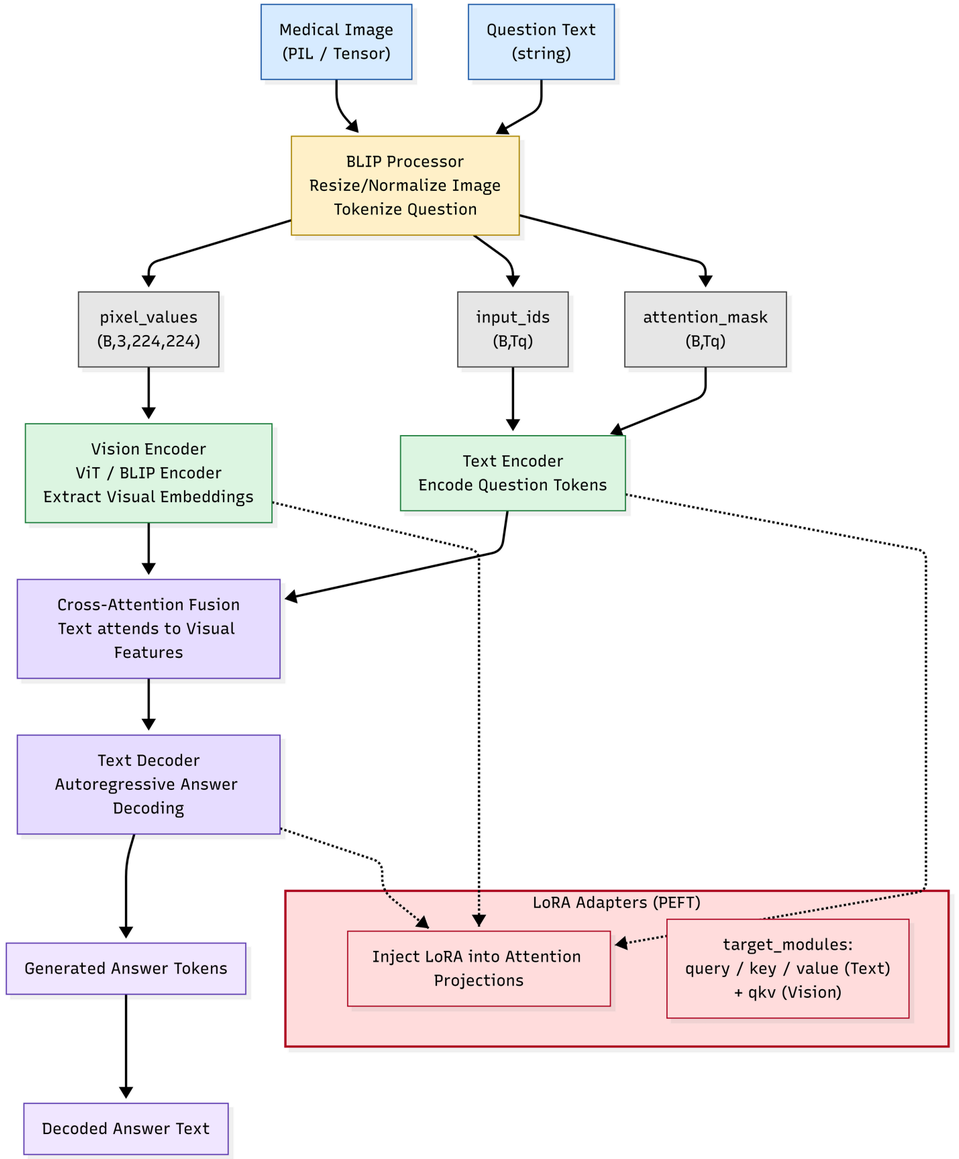


4 - Methods

BLIP + LoRA Training History



- If enough resources, we should train more, model is far from overfitting yet



5 - Results

Evaluation Metrics

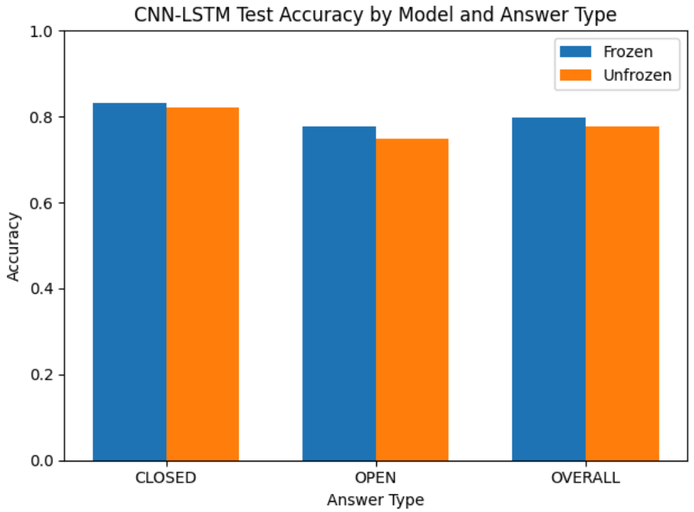
- **CNN-LSTM**
  - **Accuracy***
  - Top-5 Accuracy: GT appears within top-5 predicted classes
  - Macro-F1 (Overall / OPEN / CLOSED)
    - F1-score averaged equally across classes (sensitive to rare answers)

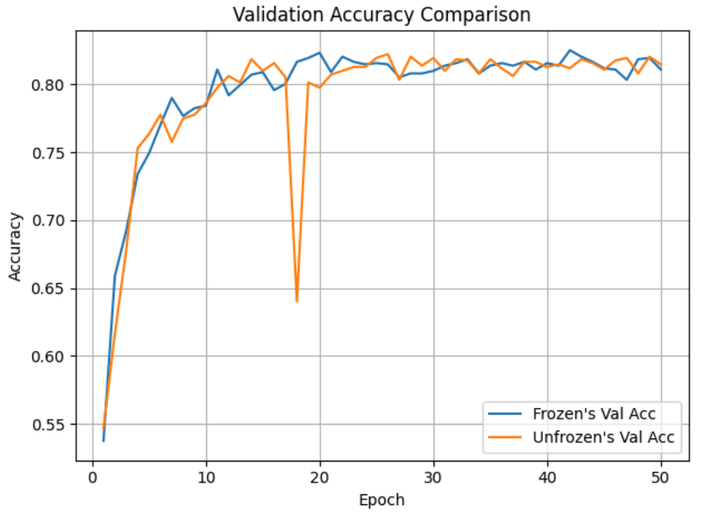
- **BLIP**
  - **Token-F1 (Overall / OPEN / CLOSED)***
    - Measures token overlap between pred and GT
  - Exact Match (normalized)
    - 1 if pred matches GT after text normalization
  - OPEN only:
    - BLEU: N-gram precision-based similarity, 0-1, higher better
    - ROUGE-L: Longest common subsequence overlap, 0-1, higher better
    - BERTScore: Semantic similarity (“CT” vs “computed tomography”, 0-1, higher better

- **Fair Cross-Model Comparison**
  - To compare discriminative vs generative:
    - Exact Match
    - Token-F1

* key metric

5 - Results
CNN-LSTM





	model	subset	accuracy	macro_f1	top5_accuracy
0	CNN-LSTM (Frozen)	OVERALL	0.7983	0.4758	0.9576
1	CNN-LSTM (Frozen)	OPEN	0.7767	0.4718	0.9318
2	CNN-LSTM (Frozen)	CLOSED	0.8317	0.6095	0.9976
3	CNN-LSTM (Unfrozen)	OVERALL	0.7766	0.4350	0.9576
4	CNN-LSTM (Unfrozen)	OPEN	0.7473	0.4238	0.9318
5	CNN-LSTM (Unfrozen)	CLOSED	0.8221	0.6600	0.9976

- **Accuracy (Key metric):**
  - CNN-LSTM (Frozen Backbone) wins in all *answer_type*.
  - CNN-LSTM does better in CLOSED (0.83), decent in OPEN (0.78) too.
- **Top-5 acc:**
  - Top-5 accuracy is much higher than Top-1 accuracy (0.96 vs 0.80).
  - Indicates correct answers are often ranked highly but not selected as top prediction.

5 - Results

BLIP (LoRA)

	model	subset	exact_match	token_f1	bleu	rougeL	bertscore_precision	bertscore_recall	bertscore_f1
0	BLIP (LoRA)	OVERALL	0.6927	0.7364	-	-	-	-	-
1	BLIP (LoRA)	OPEN	0.6202	0.6919	0.0311	0.7259	0.9194	0.9136	0.9162
2	BLIP (LoRA)	CLOSED	0.8053	0.8053	-	-	-	-	-

- **Token F1-score (Key metric):**
  - BLIP is doing better in CLOSED (0.81), significantly worse in OPEN (0.69)
  - Lower OPEN due to flexible free-form answers
- Impressive **exact match** (0.81) on CLOSED too

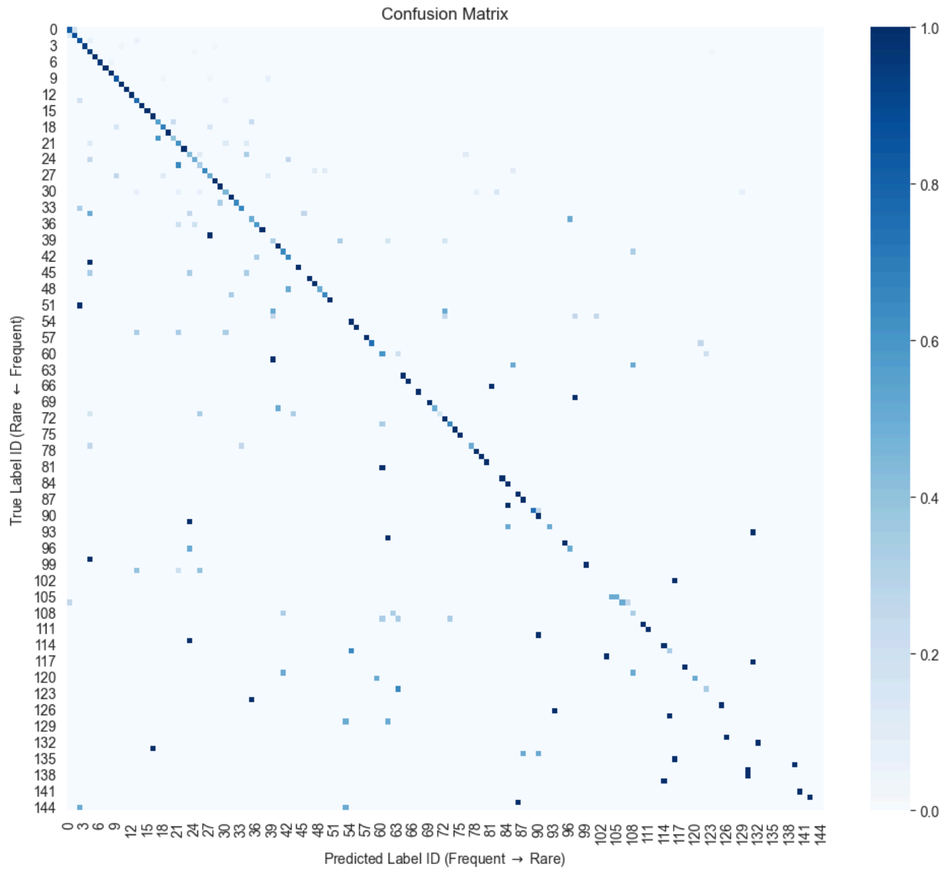
5 - Results

Long-tail Analysis (Best CNN-LSTM)

- Per-class accuracy **correlates positively** with class frequency.
- Frequent classes (e.g., modality/yes-no) **dominate** accuracy.
- Highlights limitations of fixed-vocabulary classification in Med-VQA.
- *Note:*
  - The acc spike at the tail of the distribution is due to having only 1 sample for testing

5 - Results

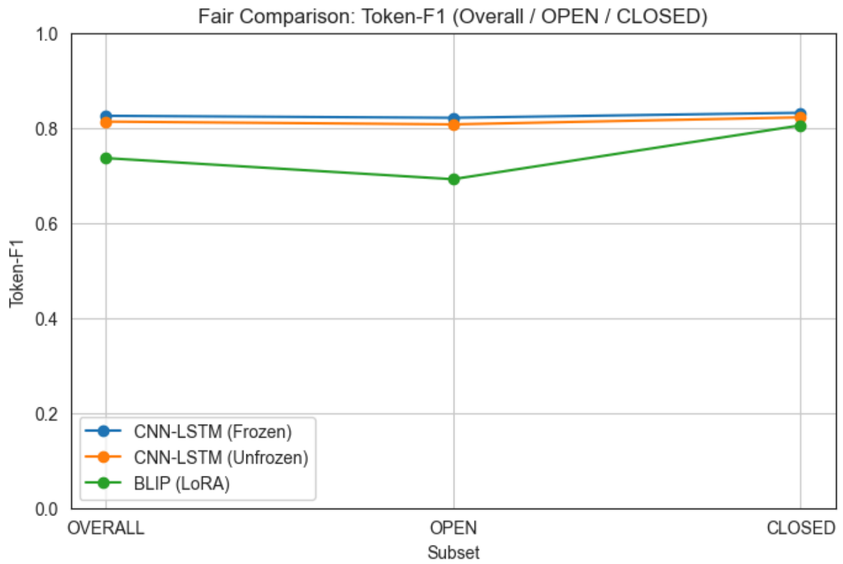
Confusion Matrix (Best CNN-LSTM)

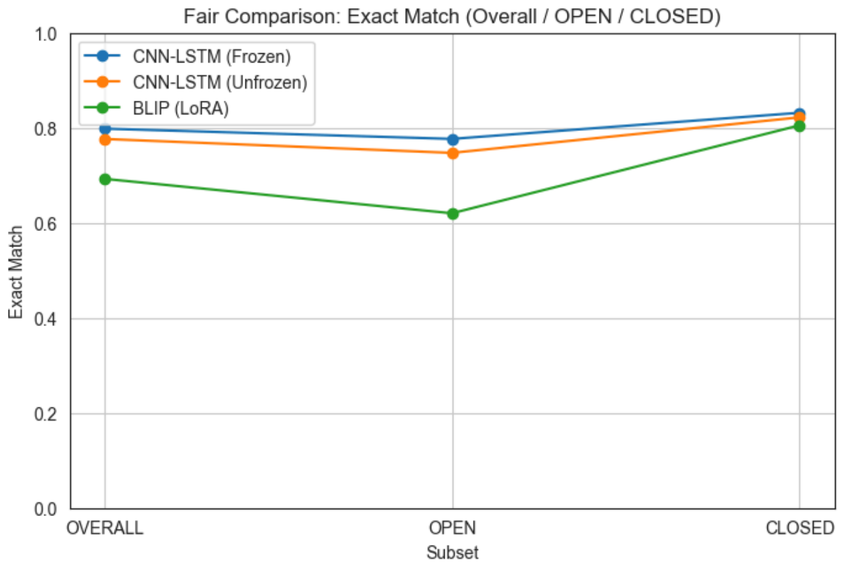


- Upper left correlates more because more training examples; lower right, more noisy, not as accurate due to less training examples
- In overall, discriminative model is making good prediction
  - **Accuracy (OVERALL):** 0.7983
  - **Accuracy (CLOSED):** 0.8317
  - **Accuracy (OPEN):** 0.7767

5 - Results

Comparison: CNN-LSTM vs BLIP





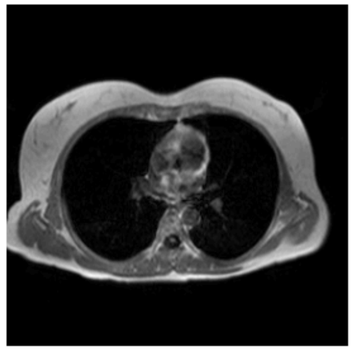
- CNN-LSTM still works better than BLIP
  - CNN-LSTM (Frozen) > CNN-LSTM (Unfrozen) > BLIP (LoRA)
- Performances are
  - consistent among OVERALL / OPEN / CLOSED
  - consistent among Token F1-score & Exact Match

	model	subset	exact_match	token_f1
0	CNN-LSTM (Frozen)	OVERALL	0.7983	0.8253
1	CNN-LSTM (Unfrozen)	OVERALL	0.7766	0.8131
2	BLIP (LoRA)	OVERALL	0.6927	0.7364
3	CNN-LSTM (Frozen)	OPEN	0.7767	0.8212
4	CNN-LSTM (Unfrozen)	OPEN	0.7473	0.8073
5	BLIP (LoRA)	OPEN	0.6202	0.6919
6	CNN-LSTM (Frozen)	CLOSED	0.8317	0.8317
7	CNN-LSTM (Unfrozen)	CLOSED	0.8221	0.8221
8	BLIP (LoRA)	CLOSED	0.8053	0.8053

5 - Results

Qualitative Analysis: CNN-LSTM vs BLIP)

Sample ID: 228
Type: OPEN



Question:

Which part of the body does this image belong to?

Ground Truth: Abdomen
CNN-LSTM: abdomen ✓
BLIP + LoRA: abdomen ✓

Sample ID: 501
Type: OPEN

Question:

What organ system is pictured?

Ground Truth: Chest
CNN-LSTM: chest ✓
BLIP + LoRA: chest ✓

Sample ID: 51
Type: OPEN

Question:

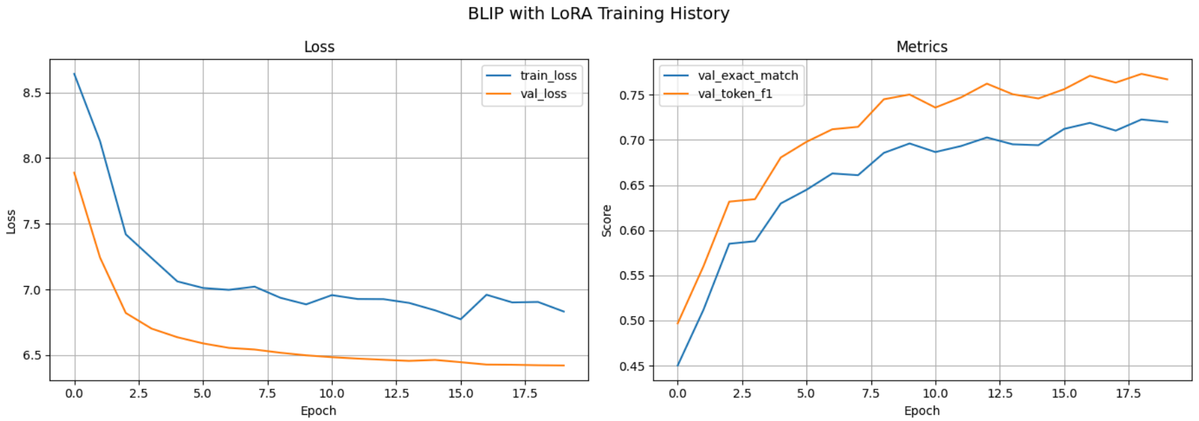
Where is/are the abnormality located?

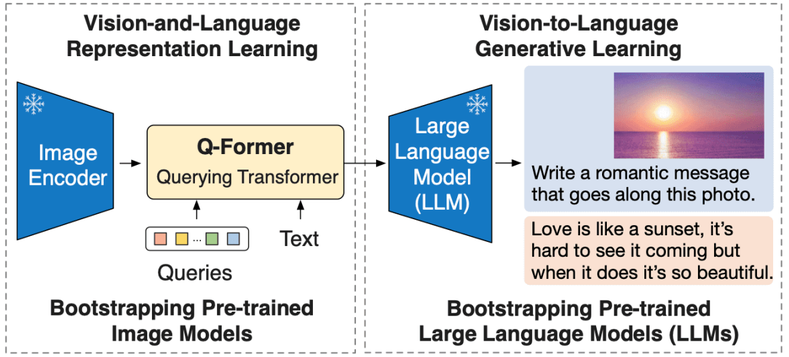
Ground Truth: Left Lung, Lower Right
CNN-LSTM: center ×
BLIP + LoRA: right lung ×

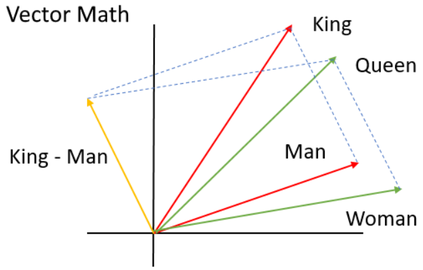
- **Common failure cases:**
  - Asking location, fine-grained anatomical terms, etc.
- **General Ideas:**
  - BLIP often produces semantically correct answers even with different wording (synonyms).
  - CNN is constrained to Top-K answer and may output <unk> for rare answers.

6 - Discussion

- **How to improve?**
  - BLIP can be trained longer (if given enough resources), far from overfitting
  - Try BLIP-2 or stronger VLMs, just 20 epochs of BLIP showed promising results
  - CNN-LSTM can utilize Word2Vec, GloVe, BERT as text encoder (semantic embeddings), without needing to train a question vocab embeddings from scratch







7 - Conclusion

- CNN–LSTM provides a strong, simple baseline for CLOSED questions at small data scale.
- BLIP + LoRA achieves the high BERTScore (>0.9) telling us that the predicted answers are semantically making sense.
- Top-5 vs Top-1 gap + lower macro-F1 indicate long-tail challenges for discriminative models.
- VLM fine-tuning with LoRA is an effective approach for medical VQA under limited compute.
- To see the results in action/reproduce the results:
  - https://github.com/hongjiaherng/woa7015-medvqa.git

THANK YOU

› End Slide