Adapting BLIP to Medical VQA with Domain Pretraining and Self-Distillation

How to improve a general vision-language model’s capabilities in the radiology domain

Motivation and Goals:

The Problem:

- General VLMs like BLIP are trained on web data, creating a domain gap with medical images
- Direct fine-tuning on small medical datasets (VQA-RAD) leads to overfitting and suboptimal performance

Our Goal:

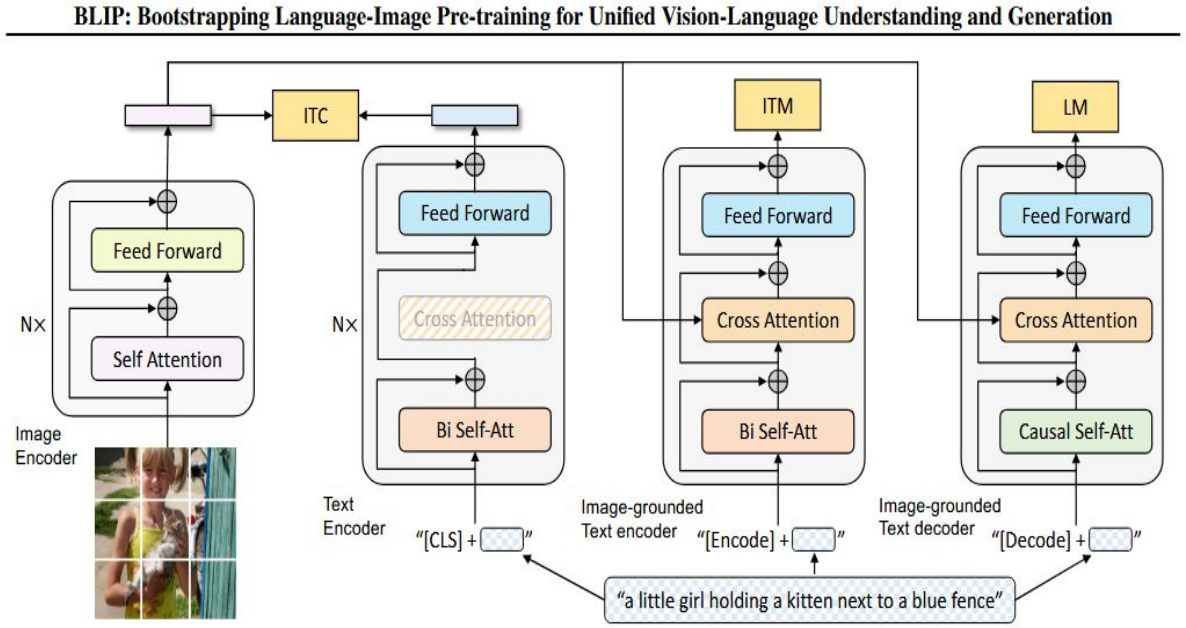
- Bridge this gap with a lightweight, effective adaptation pipeline that requires no architectural changes

Our Proposal: A two stage method combining

- Domain-Adaptive Pretraining (DAPT) on medical image-captions pairs
- Progressive Self-Distillation to stabilize training and prevent catastrophic forgetting

BLIP VLM:

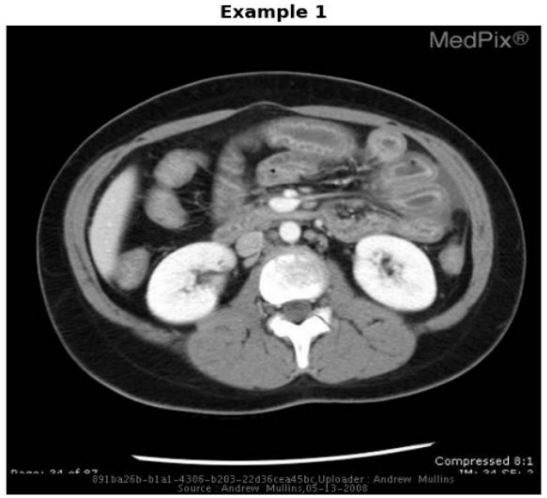
- **Architecture:** Vision Transformer (ViT) + BERT-like Encoder + Decoder
- **Features:** Operates in three modes through parameter sharing
  - Understanding (Image-Text Matching)
  - Generation (Image Captioning, VQA)
  - **Parameters:** 360M
- **Pretraining Objectives:** ITM, ITC, LM
- **Model and Weights:**
  - Original BLIP repo
  - Pretrained-only 129M Images Checkpoint



Datasets:

VQA-RAD

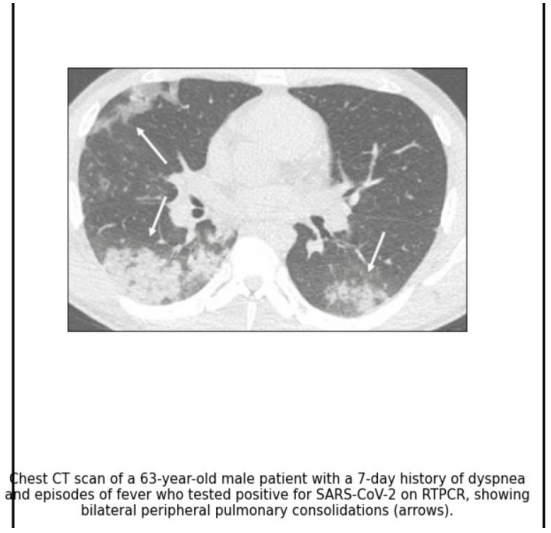
- **Type:** Medical VQA dataset pairing radiology images with clinical QAs
- **Content:** 3515 QA pairs on 315 images
- **Q/A Type:** ‘Open’ and ‘Closed’
- **Use:** Task-specific Fine Tuning and Evaluation



Q: Is the jejunal wall enlarged?
A: Yes

ROCOv2

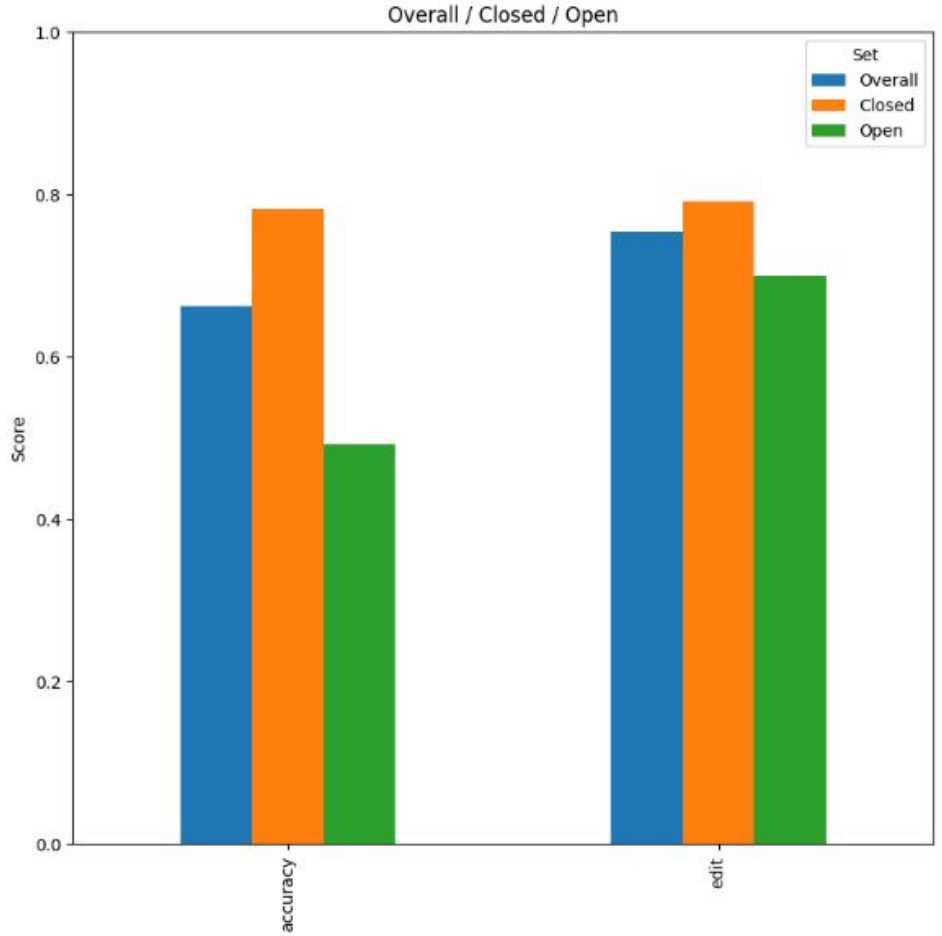
- **Type:** Large scale medical imaging dataset for multimodal learning
- **Content:** 45k (filtered by modality) radiology image-captions
- **Use:** Domain Adaptive Pretraining to learn medical visual concepts and terminology



Chest CT scan of a 63-year-old male patient with a 7-day history of dyspnea and episodes of fever who tested positive for SARS-CoV-2 on RTPCR, showing bilateral peripheral pulmonary consolidations (arrows).

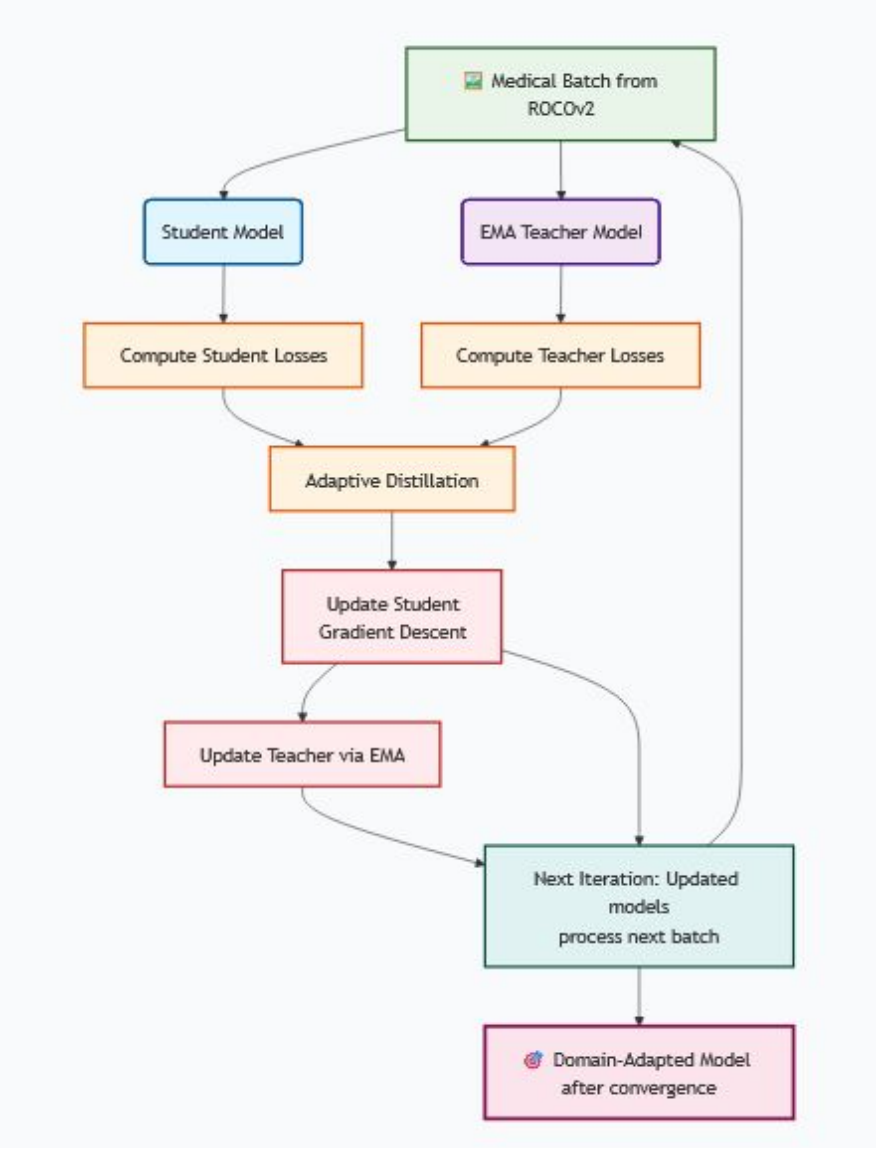
Baseline:

- **Model:** Standard BLIP architecture (ViT-B/16 + BERT-like Text Encoder + Text Decoder)
- **Training Data:** Direct fine-tuning on VQA-RAD dataset only
- **Method:** Standard fine-tuning
- **Vision:** Freezed to avoid overfitting
- **Performance:** 66.2% overall accuracy baseline
- **Key Limitation:** VQA-RAD small size
- **Purpose:** Establishes out-of-the-box BLIP performance on medical VQA
- **Evaluation:** Overall Accuracy and Edit



Our Proposed Approach:

- **Domain-Adaptive Pretraining (DAPT)**
  - Train on medical image-caption pairs (ROCOv2)
  - Learn medical visual concepts & terminology
- **Progressive Self-Distillation**
  - Teacher-student framework with EMA
  - Adaptive multi-task learning
- **Task-Specific Fine-Tuning**
  - Specialize for VQA on VQA-RAD dataset
  - Frozen visual encoder preserves medical knowledge



Domain-Adaptive Pretraining:

- **Method:** Domain-Adaptive PreTraining on medical data before VQA fine-tuning
- **Dataset:** ROCOv2 radiology corpus (45K filtered image-caption pairs)
- **Training:** Continued pretraining using BLIP's original three objectives
- **Objectives:** Image-Text Contrastive (ITC), Image-Text Matching (ITM), Language Modeling (LM)
- **IterableDataset**:
- **GradientAccumulation and MixedPrecision**
- **Result:** 72.3% overall accuracy (+6.1% over baseline)
- **Improvement:** Shows domain pretraining alone bridges significant gap

Domain-Adaptive Pretraining:

Pretraining Objectives:

Visual-Text Understanding: Image-Text Contrastive (ITC) and Image-Text Matching (ITM) align images and text.

Text Generation: Language Modeling (LM) loss trains the model to generate captions

Domain-Adaptive Pretraining:

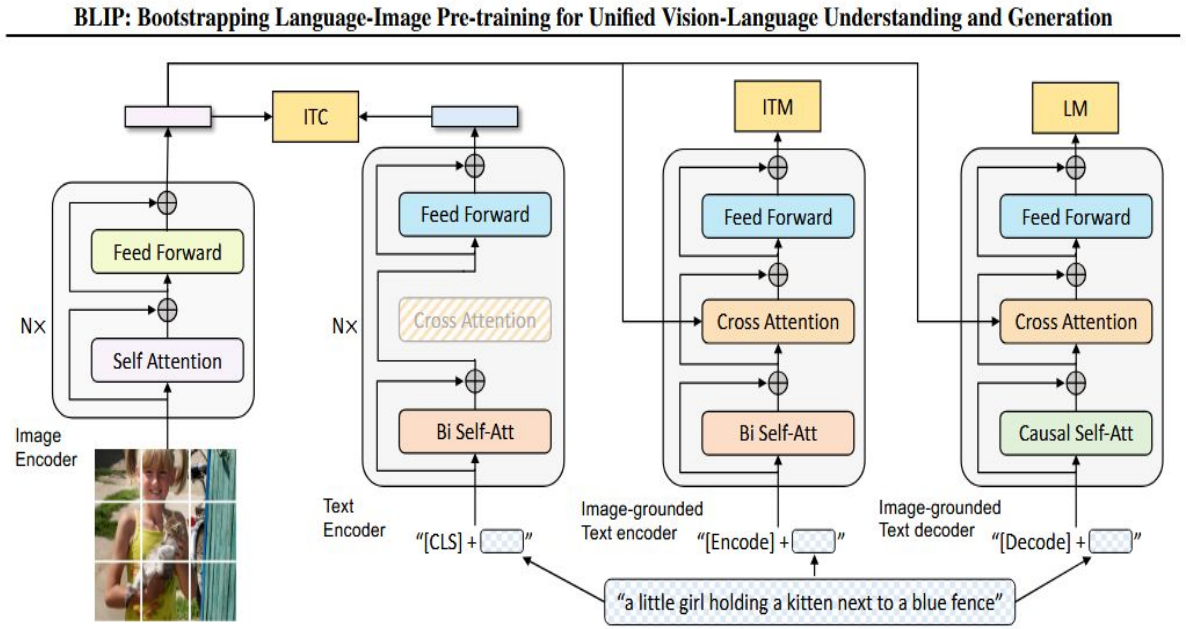
Pretraining Objectives:

$$\mathcal{L}_{ITC} = \frac{1}{2}\left(\mathcal{L}_{I\text{B}T} + \mathcal{L}_{T\text{B}I}\right)$$

$$\mathcal{L}_{ITM} = \mathbb{E}_{(I,T)}\left[y \log p + (1-y)\log(1-p)\right]$$

$$\mathcal{L}_{LM} = -\mathbb{E}_{(I,T)} \sum_{\iota=1}^{L} \log P(w_\iota | w_{<\iota}, I; \theta)$$

$$\mathcal{L}_{DAPT} = \mathcal{L}_{ITC} + \mathcal{L}_{ITM} + \mathcal{L}_{LM}$$



Progressive Self-Distillation:

- **Architecture:** Exponential Moving Average teacher guides student training
- **Key Mechanism:** Dynamic loss weighting based on relative task difficulty
- **Training:** Progressive momentum (τ: 0.95→0.995) and adaptive learning rates
- **Objectives:** Distillation losses for ITC, ITM, LM with performance-based weights
- **Result:** 74.2% overall accuracy (+2.9% over DAPT alone)
- **Advantage:** Prevents catastrophic forgetting, stabilizes multi-task learning
- **Purpose:** Demonstrates self-distillation's value for medical domain adaptation

Progressive Self-Distillation:

Student-Teacher Framework

Teacher and student models are initialised with the same weights. At each iteration, the student learns from targets and the teacher model. Afterwards, the teacher model's parameters are updated using a moving average from the student model. Higher tau means more regularised learning; lower tau, faster updates.

$$\theta_{\text{teacher}} \leftarrow \tau \cdot \theta_{\text{teacher}} + (1 - \tau) \cdot \theta_{\text{student}}$$

Progressive Self-Distillation:

Student-Teacher Framework

The student parameters updates are done by gradient descent with respect to a composed loss function given by the sum of target loss function and loss function that penalizes student models that give different outcomes from teacher model.

$$L_t^T = \text{teacher}(x), \quad L_t^S = \text{student}(x)$$

$$L_{\text{total}} = L_t^S + L_{\text{distill}}(L_t^S, L_t^T)$$

Progressive Self-Distillation:

Student-Teacher Framework

Distillation loss is given by weighted sum over MSE of student and teacher objectives. Weights are dynamically calculated to give more importance to objectives where student outputs differ more from teacher outputs, and for stability are updated using moving average

$$r_t = \frac{L_t^S}{L_t^T + \epsilon}$$

$$\hat{r}_t^{(k)} = \alpha \hat{r}_t^{(k-1)} + (1-\alpha) r_t$$

$$w_t = \frac{\exp(\hat{r}_t)}{\sum_{t'} \exp(\hat{r}_{t'})}$$

$$\mathcal{L}_{\text{distill}} = \sum_t w_t \, \text{MSE}(L_t^S, L_t^T)$$

Progressive Self-Distillation:

Distillation Loss

```
# ----- Teacher-Student Forward Passes -----
with torch.no_grad():
    teacher_loss_ita, teacher_loss_itm, teacher_loss_lm = teacher_model(images, captions, config['alpha'])

with autocast('cuda'):
    # Student forward
    student_loss_ita, student_loss_itm, student_loss_lm = model(images, captions, config['alpha'])

    # ----- Adaptive Multi-Task Distillation -----
    student_losses = {'itc': student_loss_ita, 'itm': student_loss_itm, 'lm': student_loss_lm}
    teacher_losses = {'itc': teacher_loss_ita, 'itm': teacher_loss_itm, 'lm': teacher_loss_lm}

    weights = distillation_manager.compute_weights(student_losses, teacher_losses)

    # Weighted distillation losses
    distill_loss_ita = F.mse_loss(student_loss_ita, teacher_loss_ita) * weights['itc']
    distill_loss_itm = F.mse_loss(student_loss_itm, teacher_loss_itm) * weights['itm']
    distill_loss_lm = F.mse_loss(student_loss_lm, teacher_loss_lm) * weights['lm']

    total_distill_loss = distill_loss_ita + distill_loss_itm + distill_loss_lm

    # Combined loss
    task_loss = student_loss_ita + student_loss_itm + student_loss_lm
    total_loss = task_loss + total_distill_loss
```

Training Details

```python
dapt_config = {
    'vit_grad_ckpt': True,
    'queue_size' : 256,
    'image_size' : 224,
    'alpha': 0,
    'output_dir': 'dapt_roco_psd',
    'num_epochs': 2,            # Optimal for medical adaptation
    'batch_size': 16,
    'grad_accum_steps': 4,
    'shuffle_buffer': 256,      # Larger buffer for better shuffling
    'num_workers': 2,
    'tau': 0.995,               # Slightly lower for faster teacher adaptation
    'lr': 4e-7,                 # Tuned for medical domain
    'warmup_lr': 5e-9,
    'min_lr': 1e-9,
    'weight_decay': 0.02,       # Slightly higher for regularization
    'max_grad_norm': 1.0,
    'save_every': 1,            # Save every epoch
}
```

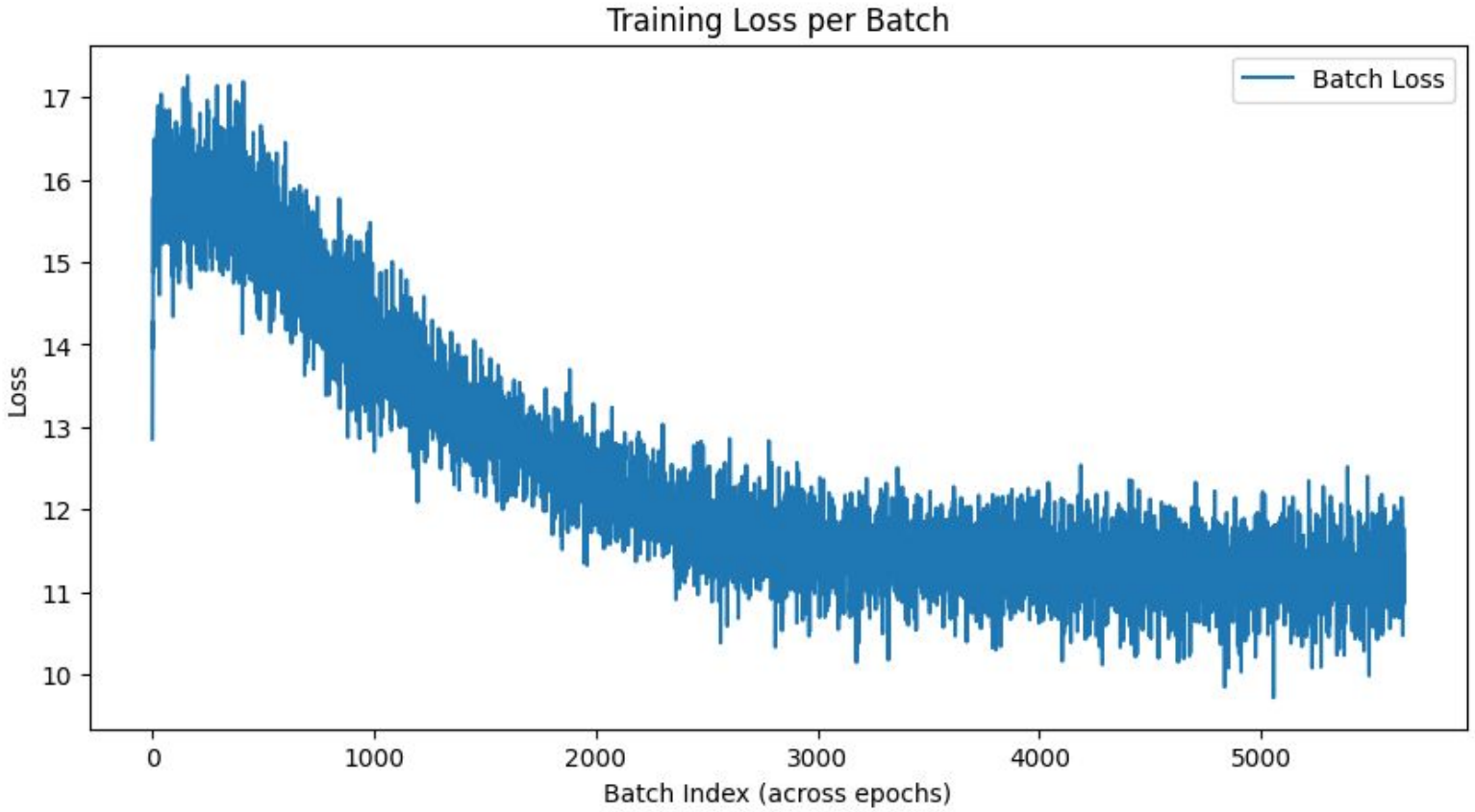
Gradient accumulation was employed to simulate a larger batch size under memory constraints.

Weight decay was applied for regularization.

Gradient clipping was used to maintain training stability and prevent exploding gradients

Domain-Adaptive Pretraining:

Training Loss

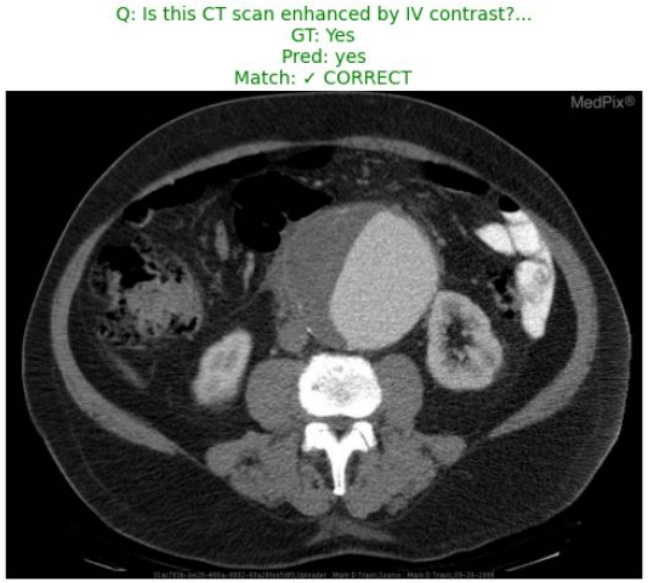


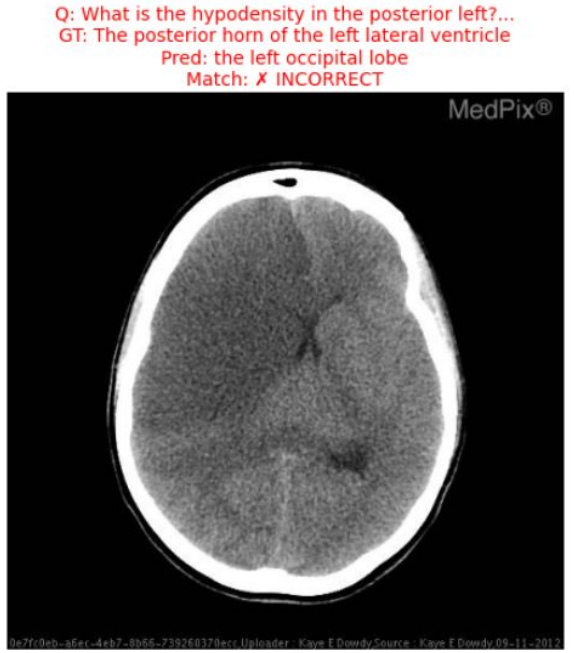
Results:

Table 1. Performance Comparison on VQA-RAD Test Set

Method	Overall Acc	Closed Acc	Open Acc	Overall Edit	Closed Edit	Open Edit
Baseline	0.662	0.782	0.492	0.754	0.792	0.700
DAPT	0.723	0.815	0.592	0.806	0.825	0.778
DAPT + Self-Distill	**0.742**	**0.845**	**0.597**	**0.828**	**0.854**	**0.792**

- Exact Match Accuracy: Strict binary metric (normalized text) for answer correctness
- Normalized Edit Similarity: Flexible metric for partial credit on phrasing variations





Conclusion and Comments:

- **Computational Constraints:**
  - 30 h/week Kaggle Tesla P100 GPU access (used mixed precision)
  - 16 GB VRAM (used gradient checkpointing, accumulation steps, etc.)
  - 50 GB Disk Space (just enough for ROCOv2)
- **Possible Extensions:**
  - Broader Medical Applications
  - Experiment with BLIP-2 (2B Params)
  - Denoise ROCOv2 captions like CapFilt
- **Key Takeaways:**
  - DAPT effectively handle the web-medical domain shift
  - Self-distillation stabilizes training and provide extra performance boost
  - This approach makes low-resource medical VQA more accessible (just used Kaggle free tier)

Resources:

- GitHub Salesforce Repo: https://github.com/salesforce/BLIP
- BLIP Paper: https://arxiv.org/abs/2201.12086
- OSF VQA-RAD: https://osf.io/89kps/
- ROCOv2 from HF: https://huggingface.co/datasets/eltorio/ROCOv2-radiology