

OmniMedVQA Diagnosis Benchmarking Report

1. Introduction

- Briefly describe the problem: diagnosis prediction from OmniMedVQA dataset using text-only, image-only, and multimodal models.
- State the dataset splits (train/val/test) and mention if stratified or random sampling was used (with justification).
- Outline the goals: compare unimodal vs multimodal approaches, evaluate performance with appropriate metrics, and assess tradeoffs in resource usage.

2. Text-Only Baseline

2.1 Model Description

In our literature review, we identified examples [1] [2] where Large Language Models (LLMs) were trained on patient questionnaires to make diagnosis. In [1], training on only text yielded 65% accuracy in diagnosis, highlighting the limitation of relying solely on textual data. I used a pretrained BERT model called BiomedBERT, which was trained on abstracts from PubMed and full text articles from PubMed Central [3]. We created a model for multiple choice using the pretrained weights from BiomedBERT. For each question, it is provided four options of answer choices and it picks the best choice. To train the model, I use the following training arguments:

Argument	Value	Description
evaluation_strategy (eval_strategy)	"epoch"	Runs evaluation after each training epoch
learning_rate	5e-5	Learning rate for the optimizer
per_device_train_batch_size	16	Number of training examples per device (e.g., per GPU)
per_device_eval_batch_size	16	Number of evaluation examples per device
num_train_epochs	2	Total number of training epochs
weight_decay	0.01	Strength of L2 regularization

2.2 Data Sampling

We filter the dataset to include only examples of questions on the disease diagnosis. As a result we have 55,387 data samples. We split the dataset into 10% test and 90% train. The train dataset was then split into 85% train and 15% validation. The dataset is stratified on modality type. We have 42380 samples for training, 7472 for validation, and 5535 for test.

2.3 Evaluation

The reported results from our training shows that the model does not generalize, achieving similar losses for both training and validation. It also achieves a very high accuracy of 0.995 in the validation set.

Epoch	Validation Accuracy	Validation Loss	Training Loss
1	0.993041	0.030241	0.036300
2	0.994781	0.021077	0.022700

Evaluating the trained model on the test set we get the following results:

Accuracy	Precision (Macro)	Recall (Macro)	F1 (Macro)
0.9906	0.9909	0.9913	0.9911

Since there are four possible options or classes the model decides on, we compute a macro average for precision, recall, and f1 scores.

2.4 Observations

The model performs very well on the text based questions. This highlights the simplicity of the dataset’s questions. Examining the questions that appear in the test set and the training dataset, many questions are very similar with similar answer options. Without any images and only the text of the questions and the options, the model performs very well making correct guesses based on similar questions that appeared in the training data. The number of unique question strings that appear in the test dataset is 2316. Of those unique questions in the test set, 1868 questions appear in the training dataset. This substantial overlap between train and test questions introduces a risk of data leakage, potentially inflating the reported accuracy and limiting the validity of generalization claims, as the model may simply memorize or recall previously seen questions rather than truly learning to generalize. Computing the accuracy only on questions that appeared in training dataset and those that did not we get:

Test Set Subset	Accuracy
Questions In Train	0.992
Questions Not In Train	0.979

The model does perform slightly worse on unseen questions. But it still performs very well as these different questions may have similar answer options to other questions or the question is similar to another question in the dataset.

3. Image-Only Baseline

3.1 Model Description

We implemented an image-only classification baseline using a **ResNet18** architecture pretrained on ImageNet. To adapt it for diagnosis classification, we replaced the final fully-connected (FC) layer to match the number of unique diagnosis labels.

Model Details:

- Backbone: `torchvision.models.resnet18`
- Pretrained Weights: ImageNet (ResNet18_Weights.DEFAULT)
- Final Layer: Replaced with a new `nn.Linear(in_features, num_classes)`
- Loss Function: `CrossEntropyLoss`
- Optimizer: Adam (`lr=1e-4`)
- Learning Rate Scheduler: `StepLR (step_size=5, gamma=0.5)`

Image Preprocessing:

- Resize to (224, 224)
- Normalization: Used ResNet18 default transforms (mean/std)
- Augmentations (train set only):
 - Random horizontal flip
 - Random rotation (± 10 degrees)

We trained for 3 epochs, using a batch size of 32.

3.2 Data Sampling

We constructed a dataset consisting only of the image and the correct diagnosis label (no text context). For each question, the correct answer label was extracted and mapped to a class index.

The same data splits from the overall pipeline were used:

- Training: 42,380 samples
- Validation: 7,472 samples
- Test: 5,535 samples

All images used were referenced from the full OmniMedVQA dataset. Class distribution was maintained implicitly (since we’re using the same stratified train/val/test splits as the other baselines).

Each data sample consists of:

- The path to the question-associated image
- The ground truth diagnosis label (as class index)

3.3 Evaluation

- Metrics: Accuracy, Precision, Recall, F1.
- Results on validation and test sets.
- Resource usage.

3.4 Observations

- Strengths (e.g., good at visually distinct pathologies).
- Weaknesses (e.g., struggles without textual context).

4. Multimodal Model (Reproduced from Literature)

4.1 Model Description

We reproduced the multimodal VQA model introduced in [4] (*Foundation Models for Generalist Medical AI*), which adapts the **OpenFlamingo** architecture to combine medical images with natural language questions and answers.

- **Name and Citation:** OpenFlamingo-based multimodal VQA [Moor et al. 2023].
- **Original vs. Our Implementation:**
 - The original paper used **OpenFlamingo-9B**, which integrates a ViT-G/14 vision encoder with a large language model.
 - Due to time and storage constraints, we used the lighter **OpenFlamingo-3B** variant (`openflamingo/OpenFlamingo-3B`). This model follows the same architectural design but is significantly smaller and more manageable for our setup.
- **Model Components:**
 - **Vision Encoder:** Pretrained CLIP ViT-L/14.

- **Language Encoder:** Pretrained MPT-1B-RedPajama-200B.
- **Fusion Strategy:** Cross-attention layers inserted into the transformer stack (matching OpenFlamingo).
- **Parameter-Efficient Tuning:** LoRA adapters added to the language model for efficient fine-tuning.
- **Training Setup:**
 - Optimizer: AdamW ($1r=1e-5$).
 - Loss: CrossEntropy with padding mask.
 - Epochs: **3** (shortened from the original setup to save compute).
 - Batch sizes: 5 (train) / 2 (eval).
 - Runtime: **~170 minutes** for the reduced dataset run.

In summary, we scaled down the model size, number of epochs, and dataset size for feasibility, but preserved the **core architecture** (vision encoder, language encoder, fusion layers, and LoRA fine-tuning) to remain faithful to the original design.

4.2 Data Sampling

We adapted the dataset preparation strategy from the paper, with modifications for efficiency:

- **Dataset:** OmniMedVQA.
- **Splits:**
 - **90%** of data reserved for training/validation, **10%** for testing.
 - Within the training portion, an **85/15 split** was applied for train/validation.
- **Subsampling:** To save time and storage, we randomly sampled a subset of the dataset. This enabled training to complete within ~170 minutes on a single CPU while still allowing meaningful evaluation.

4.3 Evaluation

Epoch	Training Loss	Validation Loss
0	2.383	1.478
1	1.110	1.443
2	0.850	1.413

- **Metrics:** Accuracy, Precision, Recall, and F1.

Accuracy	Precision	Recall	F1
0.802	0.831	0.859	0.843

The training loss steadily decreased across epochs (from 2.383 to 0.850), showing that the model was able to effectively learn from the data. Validation loss also improved, though it plateaued after the first epoch, suggesting that further training or regularization may be required to prevent overfitting. On the evaluation set, the model achieved an accuracy of 0.802 with precision of 0.831, recall of 0.859, and F1 score of 0.843. These values indicate a balanced performance across metrics, with recall being slightly stronger than precision, which suggests the model is more effective at capturing positive cases than avoiding false positives. Given the relatively small dataset, the use of the smaller OpenFlamingo-3B model, and the limited number of training epochs, these results are promising and suggest that performance could improve with additional training time, larger data samples, or scaling to a larger model variant.

4.4 Observations

- **Comparison with Text Baseline:**

- The **text-only baseline** achieved higher direct accuracy than our multimodal model under the current constraints.
- This is expected because many questions in the dataset can be answered from textual information alone, and the multimodal model was limited by **smaller model size (3B vs. 9B)**, **dataset subsampling**, and only **3 training epochs**.
- Despite these limitations, the multimodal model still reached **80% accuracy**, which is strong given the reduced setup. With more data, longer training runs, and the full 9B model, we expect the multimodal model to surpass the text baseline, as it can leverage both visual and textual reasoning.

- **Strengths:**

- Successfully reproduced the OpenFlamingo design (vision encoder + language encoder + fusion).
- LoRA adapters enabled efficient fine-tuning on limited resources.
- Demonstrated competitive results even with reduced scale.

- **Weaknesses / Practical Challenges:**

- Underperformed relative to the text-only baseline due to reduced scale and short training.
- High compute cost (170 minutes for 3 epochs) even in reduced form.
- **Environment fragility:** Running OpenFlamingo required **exact Python and module versions** (PyTorch, Hugging Face Transformers, and dependencies).
 - * Incorrect versions prevented the model from downloading or executing.
 - * This version sensitivity made reproducibility more difficult than with the lighter baselines.

5. Conclusion

- Summarize performance comparisons between text-only, image-only, and multimodal models.
- Highlight tradeoffs: accuracy vs. resource usage vs. interpretability.
- Discuss implications for diagnosis tasks and possible future work (e.g., lighter multimodal models, interpretability improvements).

References

- [1] T. Uehara *et al.*, “Accuracy of diagnoses predicted from a simple patient questionnaire stratified by the duration of general ambulatory training: An observational study,” *International Journal of General Medicine*, vol. 7, pp. 13–19, 2013, doi: 10.2147/IJGM.S53800.
- [2] T. Tu, M. Schaekermann, A. Palepu, *et al.*, “Towards conversational diagnostic artificial intelligence,” *Nature*, vol. 642, no. 8029, pp. 442–450, 2025, doi: 10.1038/s41586-025-08866-7.
- [3] Y. Gu *et al.*, “Domain-specific language model pretraining for biomedical natural language processing.” 2020.
- [4] M. Moor *et al.*, “Med-flamingo: A multimodal medical few-shot learner,” in *Proceedings of the 3rd machine learning for health symposium*, in Proceedings of machine learning research, vol. 225. PMLR, 2023, pp. 353–367. Available: <https://proceedings.mlr.press/v225/moor23a.html>