A Multimodal Generative AI Framework for Cancer Pathology Classification

Mayank Kapadia, Basanth Periyapatna Roopa Kumar, Nischitha Nagendran
Department of Applied Data Science
San Jose State University

***Abstract*—**

We propose a multimodal generative AI framework for cancer pathology that combines three capabilities: (i) histopathology picture classification, (ii) clinical note classification, and (iii) prompt-driven clinical captioning using retrieval-augmented generation (RAG). The approach establishes accurate vision baselines on PatchCamelyon (PCam) [7] and creates a complementing text pipeline by curating and categorizing TCGA BRCA clinical notes [8] via two distinct routes to assure trustworthy supervision. We investigate caption utility by creating image-based descriptions and constructing a lightweight caption-label classifier to assess RAG's downstream worth. Together, these components establish a uniform data and model foundation, explain interfaces for combining picture and text representations, and give an evidence-based approach to short, clinically grounded summaries. The main objective is to provide a practical, auditable procedure that improves diagnostic support by matching visual findings with structured language while still allowing for parameter-efficient adjustment and eventual inclusion into clinical decision tools.

I. Introduction

Cancer pathology includes both pixels (slides) and language (clinical notes). Reliable automation necessitates both solid unimodal baselines and proof that text-image bridges (captions) transmit useful signal. This work builds those foundations on public data and strictly reports what we ran and measured.

A. Motivation

- Manual slide inspection and free-text reporting clog pathology workflows; consistent baselines are required before implementing fused systems.
- Public data (PCam images [7], TCGA BRCA notes [8]) enables transparent, reproducible benchmarking of images, notes, and generated captions.
- We want to know whether captioning contributes measurable, transportable signal beyond image classifiers.

B. Contributions

C. Research Questions

1) **Backbone robustness on PCam.** We identify which image backbone delivers the most robust and consistent performance on PCam under our unified training/eval setup.
2) **Caption signal analysis.** We test whether BLIP-2 captions provide a consistent semantic signal (via CLIP similarity) and whether they support downstream categorization through a caption→label classifier.
3) **Lightweight fusion vs. image-only.** We benchmark a frozen-encoder fusion head (ResNet-50 + ClinicalBERT [9]) against image-only models to quantify competitiveness and trade-offs.
4) **ClinicalBERT tuning stability.** We compare baseline, SFT/TAPT, and LoRA [10]-efficient tuning on curated/balanced BRCA notes to determine which path yields the most stable classification.
5) **Limits informing next steps.** We document constraints (e.g., caption-only underperformance) that shape the design of retrieval-augmented, prompt-driven captioning in the next stage.

II. Dataset Description

A. Histopathology Images Dataset

Our experiments use the **PatchCamelyon (PCam)** histopathology patch dataset [7] via the `torchvision` `PCAM` interface. Each example is a color image patch paired with a binary label (benign vs. malignant). We *do not* relabel, rebalance, or resplit the data.

While large-scale slide-level datasets such as BreakHis [6] have been widely adopted for breast cancer histopathology benchmarking, we focus on PCam for its standardized patch-level splits and strong alignment with our computational constraints.

Split composition. We follow the official splits: **262,144** training, **32,768** validation, and **32,768** testing images. All reported metrics are computed on these fixed splits for strict reproducibility.

Use in this work. PCam supports two complementary tracks:

1) *Image–classification baselines* with EfficientNet-B0, ResNet-50, and ViT-Base/16 under a common evaluation protocol.
2) A *captioning probe*: we generate captions for the full corpus using BLIP-2 (FLAN-T5-XL), assess caption quality via CLIP similarity, and train a lightweight caption→label sequence model (FLAN-T5) to test diagnostic signal from text alone versus image-only models.

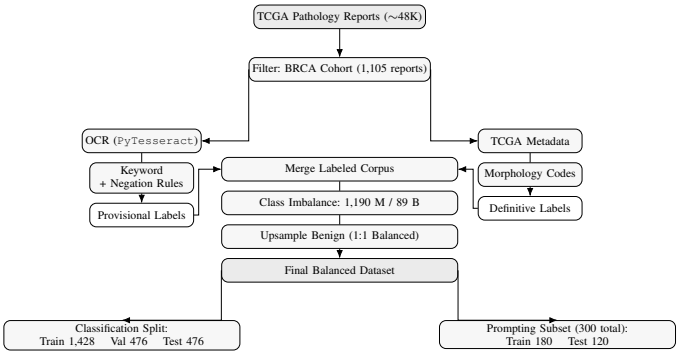

Fig. 1. Pipeline for extraction, annotation, balancing, and dataset splitting for TCGA BRCA clinical notes.

Preprocessing and integrity. The dataset is used as released, preserving labels and split membership. No additional augmentations are introduced; preprocessing is limited to standard tensor conversion and normalization required by each model family.

B. Clinical Notes Dataset

The clinical notes were provided via the Cancer Genome Atlas (TCGA) [8] portal, which is focused on the Breast Invasive Carcinoma (BRCA) cohort. 1,105 BRCA-specific pathology reports were extracted from ∼48K reports on various cancer types. Because the reports did not have labels, two annotation approaches were used.

Approach 1: After digitizing reports with `PyTesseract` OCR, diagnostic words were extracted to produce an unstructured CSV corpus. Cases were classified as benign or malignant using keyword-based rules and negation patterns (e.g., "no evidence of malignancy").

Approach 2: TCGA metadata was analyzed to retrieve morphological codes, with "/3" indicating malignant and "/0–2" indicating benign. The data revealed a considerable class imbalance (1,190 malignant vs. 89 benign), which was addressed by random upsampling of the benign class to obtain a balanced 1:1 ratio.

The final balanced dataset included 2,380 pathology reports, which were used for further tasks. For **clinical-note classification**, data were split into **train: 1,428**, **validation: 476**, and **test: 476**. Due to computational costs, a subset of **300 reports** was chosen for prompt-based caption creation, with **train: 180** and **test: 120** parameters.

Figure 1 illustrates the complete extraction, labeling, and splitting pipeline.

III. Evaluation Metrics

This section outlines the quantitative metrics employed for evaluating classification and caption generation tasks. Metrics were computed using `scikit-learn.metrics` and Hugging Face `evaluate` library to ensure consistency across all experiments.

A. Classification Metrics

For the histopathology image and clinical note classification tasks, the following metrics were used: Accuracy, Precision, Recall, F1-score, and ROC-AUC. Let TP, FP, TN, and FN denote true positives, false positives, true negatives, and false negatives, respectively.

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN} \tag{1}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

ROC-AUC was employed to measure model discrimination across decision thresholds, computed as the area under the receiver operating characteristic curve (ROC). During hyperparameter tuning, training and validation loss curves were also analyzed to monitor overfitting and convergence. The best-performing models, as determined by validation F1 and ROC-AUC, are reported in the Experimental Results section.

B. Caption Generation Metrics

For evaluating captioning performance, five standard text generation metrics were used: ROUGE-1, ROUGE-2, ROUGE-L [11], BLEU, and BERTScore$_{F1}$ [12]. ROUGE-L captures the longest common subsequence. BLEU evaluates n-gram precision with brevity penalty, and BERTScore computes contextual similarity between embeddings from a pre-trained transformer.

$$\text{ROUGE-}n = \frac{\sum_{S \in \text{Ref}} \sum_{gram_n \in S} \min(\text{Count}_{\text{gen}}, \text{Count}_{\text{ref}})}{\sum_{S \in \text{Ref}} \sum_{gram_n \in S} \text{Count}_{\text{ref}}} \tag{4}$$

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \quad BP = \min(1, e^{1-r/c}) \tag{5}$$

Where BP is the brevity penalty, r and c denote reference and candidate lengths, and p_n is the n-gram precision. BERTScore$_{F1}$ uses contextual embeddings from a transformer to compute pairwise similarity between generated and reference tokens.

Figure 2 illustrates the evaluation pipeline for both classification and caption generation modules.

IV. Methodology

A. Histopathology Classification

1) Model Overview: We benchmark three modern image backbones on PatchCamelyon (PCam) [7] using a unified, classification-only pipeline:

- **EfficientNet-B0** (convolutional baseline).
- **ResNet-50** (residual CNN baseline).
- **ViT-Base/16** (Vision Transformer with 16 × 16 patches).

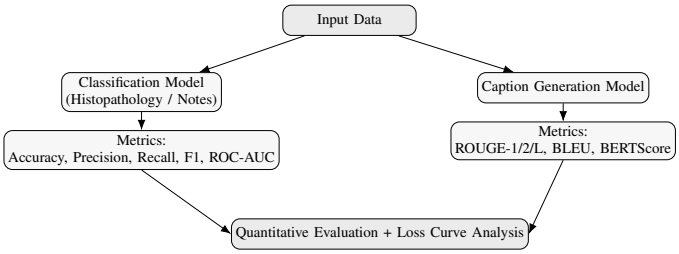

Fig. 2. Evaluation pipeline showing separate metric computation for classification and caption generation tasks.

Each backbone is instantiated from standard libraries and adapted to binary prediction (benign vs. malignant) with a single linear classification head. We *preserve the official PCam splits* [7] and evaluate all models under the same protocol and metrics. Decision thresholds for converting probabilities to labels are *selected on the validation set* (e.g., ResNet-50 $\tau = 0.22$, ViT-B/16 $\tau = 0.36$) and then fixed for test-time reporting. In addition to the image-only baselines, we also record a *lightweight reference* for later fusion by training a head over *frozen* encoders (ResNet-50 image features + ClinicalBERT [9] text features); this is reported for context only and does not change the image-only training setup.

2) Experimental Settings: All runs follow the same data and evaluation settings to ensure comparability:

- **Data**: PCam official splits (train 262,144, val 32,768, test 32,768). No relabeling, rebalancing, or re-splitting.
- **Preprocessing**: Minimal, backbone-appropriate transforms (tensor conversion and normalization as required by each family). No additional augmentations.
- **Objective & outputs**: Binary classification with sigmoid probabilities; final labels obtained by a threshold τ tuned on the validation set and then held fixed for test.
- **Metrics**: Accuracy, F1, and ROC–AUC computed on the fixed validation/test splits; threshold tuning uses validation only.
- **Training environment**: PyTorch on a single NVIDIA A100 (Google Colab). Model weights loaded from standard libraries; evaluation and metric computation performed offline on the saved predictions.
- **Protocol**:
  1) Train each backbone on the official training split using its standard input configuration.
  2) Select τ on the validation split to optimize F1/ROC–AUC as reported.
  3) Evaluate on the held-out test split with the selected τ; log Accuracy, F1, ROC–AUC.

This setup yields directly comparable baselines across EfficientNet-B0, ResNet-50, and ViT-Base/16, and provides the image side of the multimodal system used later alongside clinical text models and caption-based analyses.

B. Clinical Note Classification

1) Model Overview: The pipeline is consistent across both preprocessing procedures. The balanced clinical notes are used as input to a **ClinicalBERT** [9] model for benign and malignant classification. We evaluate the pretrained model as a baseline, then sequentially apply **Supervised Fine-Tuning (SFT)** and **Task-Adaptive Pretraining (TAPT)** to increase domain adaptation, followed by **LoRA** [10] to fine-tune parameters efficiently. Figure 3 illustrates the procedure.

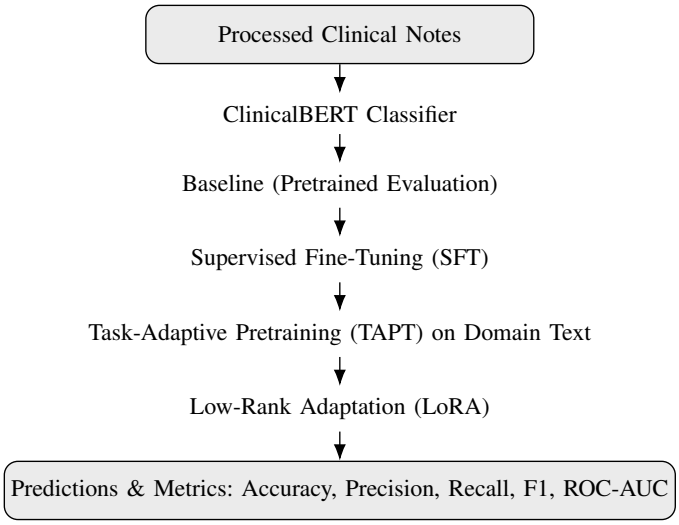

Fig. 3. Clinical note classification pipeline: sequential SFT → TAPT → LoRA training stages applied on ClinicalBERT for improved domain adaptation and efficiency.

TABLE I. SUMMARY OF EXPERIMENTAL SETTINGS FOR CLINICAL NOTE CLASSIFICATION.

Parameter	Value / Range
Hardware	A100 (Colab), RTX 5090 (SJSU)
Frameworks	PyTorch v2.9.0, Transformers v4.57.1
Libraries	NumPy, Pandas, scikit-learn, Datasets, PEFT
Batch size	16
Loss function	CrossEntropyLoss
Optimizer	AdamW
Scheduler	Linear (post-baseline)
Learning rate	[5e–6, 1e–5, 2e–5, 5e–5, 1e–4]; Best = 2e–5
Epochs	10 (early stopping: 2–4)
LoRA rank (r)	[4, 8, 16]
LoRA α	[8, 16, 32]
Trainable parameters	0.7M (vs. 135.9M full)
Metrics	Accuracy, Precision, Recall, F1, ROC-AUC
Seeds	10, 20, 42, 999, 2025

2) Experimental Settings: The experiments were conducted in two environments: SJSU GPU Lab (RTX 5090) and Google Colab (A100 GPU). Table I highlights key configurations, hyperparameters, and performance metrics tracked throughout all experiments. These configurations form the foundation for the results and comparisons presented in the following section.

C. Caption Generation using LLMs and Prompting Techniques

1) Model Overview: We address limited caption supervision by first comparing three zero-shot LLMs (GPT-4o, GPT-3.5-turbo, Claude 3.5 Sonnet) on 300 stratified notes (150 benign / 150 malignant) using identical instructions. A fourth LLM (DeepSeek) serves as an objective judge and selects GPT-4o as the best performer; its captions are adopted as

ground truth for the **LiquidAI RAG** pipeline. RAG is then trained and fine-tuned with **LoRA** [10] (rank r, scale α) while tuning output token limit and input context length. Finally, we evaluate four prompting strategies *within* the RAG setup under identical hyperparameters: Zero-Shot, Self-Reflection, Self-Ask, and Tree-of-Thought (ToT). Detailed metric comparisons appear in the Experimental Results section.

2) Prompt Design (System & User Prompts):

a) Zero-Shot (instruction only): **System**

```
You are a board-certified pathologist. Output exactly ONE
     clinically precise sentence, only facts present in the
     note. No patient identifiers, no speculation.
```

User

```
Write ONE single-sentence caption summarizing the key
     diagnosis/findings strictly supported by the note.

NOTE:
{<clinical_note_text>}

Caption:
```

b) Self-Reflection (draft → critique → revise): **System**

```
You are a board-certified pathologist. Output exactly ONE
     clinically precise sentence, only facts present in the
     note. No patient identifiers, no speculation.
```

User

```
Task: Write a ONE-sentence clinical caption strictly
     supported by the NOTE.

Step 1 (Draft): Write a draft caption.
Step 2 (Critique): List any missing or unsupported items
     among:
- diagnosis, grade, tumor size, margins, lymph nodes, ER/PR/
     HER2, special findings
Step 3 (Revise): Produce a final ONE-sentence caption that
     fixes issues.

Return only:
Draft: ...
Critique: ...
Final: ...

NOTE:
{<clinical_note_text>}
```

c) Self-Ask (internal Q&A → final caption): **System**

```
You are a board-certified pathologist. Output exactly ONE
     clinically precise sentence, only facts present in the
     note. No patient identifiers, no speculation.
```

User

```
You are a pathology assistant helping to summarize a
     clinical note.

Step 1: Ask yourself up to 6 short, high-yield questions (
     diagnosis, grade, tumor size, margins, nodes, ER/PR/
     HER2, special findings).
Step 2: Answer each question concisely using only the NOTE.
Step 3: Using your answers, craft ONE factual clinical
     sentence as the final caption.

--- Output format (return ONLY this line) ---
Final Caption: ONE sentence only
-------------------------------------------

NOTE:
{<clinical_note_text>}
```

d) Tree-of-Thought (multi-branch reasoning → best caption): **System**

```
You are a board-certified pathologist. Output exactly ONE
     clinically precise sentence, only facts present in the
     note. No patient identifiers, no speculation.
```

User

```
You are a pathology expert trained to summarize diagnostic
     notes into a concise one-sentence caption.

Using a Tree-of-Thought approach:
- Create multiple distinct reasoning branches covering:
  (1) Diagnostic details (diagnosis, grade, size)
  (2) Pathological context (margins, nodal status, ER/PR/
       HER2)
  (3) Summary optimization (factual accuracy, concision,
       clinical tone)
- Internally compare branches and resolve inconsistencies.
- Important: DO NOT show your reasoning or branches.
- Output ONLY the final single-sentence caption that best
     summarizes the NOTE.

NOTE:
{<clinical_note_text>}

Final Caption:
```

e) LLM-as-a-Judge (DeepSeek) for model selection: **System**

```
You are a senior pathologist and clinical language expert.
You will evaluate automatically generated one-sentence
     pathology captions.
Judge only the factual accuracy and clarity of each caption
     based on the given clinical note.
Always stay objective and consistent.
Assign numeric scores from 1 to 5 for each evaluation
     criterion and select a single overall winner.
```

User

```
You are given a clinical note and 3 automatically generated
     captions.
Score each caption (A, B, C) from 1 to 5 for:
1) Clinical Correctness
2) Evidence Grounding
3) Relevance & Specificity
4) Clarity & Conciseness
Then choose one overall winner: A, B, C, or Tie.

CLINICAL NOTE:
<<<
{<clinical_note_text>}
>>>

CAPTION A (GPT-4o):
{<cap_gpt4o>}

CAPTION B (GPT-3.5-turbo):
{<cap_gpt3.5>}

CAPTION C (Claude 3.5 Sonnet):
{<cap_claude>}

Return STRICT JSON ONLY in this schema:
{
  "A": {"correctness": x, "grounding": x, "relevance": x, "
       clarity": x},
  "B": {"correctness": x, "grounding": x, "relevance": x, "
       clarity": x},
  "C": {"correctness": x, "grounding": x, "relevance": x, "
       clarity": x},
  "winner": "A|B|C|Tie"
}
```

TABLE II. SUMMARY OF EXPERIMENTAL SETTINGS FOR LIQUIDAI RAG CAPTION GENERATION.

Parameter	Configuration / Range
Hardware	A100 (Colab), RTX 5090 (SJSU GPU Lab)
Frameworks	PyTorch v2.9.0, Transformers v4.57.1
Libraries	NumPy, Pandas, scikit-learn, Datasets, PEFT
Dataset size	300 (180 training / 120 testing)
Output tokens	[40, 60, 100]; Best = 60
Context window (max_token)	[512, 768, 1024, 2048, 5000, 10000, full text]; Best = 204
LoRA parameters	$r \in \{4, 8, 16\}$; $\alpha \in \{16, 32, 64\}$ (paired)
Learning rate (LR)	[2e–5, 1e–5, 1.5e–5, 7e–6]
Epochs	20 (early stopping not triggered)
Batch size	4 (LoRA fine-tuning: 1)
Prompting methods	Zero-Shot, Self-Reflection, Self-Ask, Tree-of-Thought
Evaluation metrics	ROUGE-1/2/L, BLEU, BERTScore-F1

3) Experimental Settings: The experiments were conducted in two environments: SJSU GPU Lab (RTX 5090) and Google Colab (A100 GPU). Table II highlights key configurations, hyperparameters, and performance metrics tracked throughout all experiments. These configurations form the foundation for the results and comparisons presented in the following section.

V. EXPERIMENTAL RESULTS

A. Histopathology Classification Results

Protocol. All image models were trained and evaluated on the official PCam splits (train 262,144 / val 32,768 / test 32,768), with family-specific normalization and identical optimization settings. Decision thresholds were selected on the validation set when applicable; EfficientNet-B0 is reported with the default threshold (0.50). We report Accuracy, F1, and ROC–AUC on the test split. **Results.** Table III summarizes the test performance of three backbones. ViT-Base/16 delivers the most consistent performance across all metrics; ResNet-50 is a close second; EfficientNet-B0 remains a strong lightweight baseline even without threshold tuning.

TABLE III. PCAM TEST PERFORMANCE OF IMAGE CLASSIFIERS (THRESHOLDS TUNED ON VALIDATION UNLESS NOTED).

Model	Thr.	Acc	F1	ROC–AUC
EfficientNet-B0	0.50	0.8760	0.8627	0.9508
ResNet-50	0.22	0.8871	0.8818	0.9500
ViT-Base/16	0.36	0.8898	0.8852	0.9601

Analysis. (1) *Ranking:* ViT-Base/16 > ResNet-50 > EfficientNet-B0 by ROC–AUC, with small but consistent gains in Acc/F1 as well. (2) *Thresholding:* simple threshold tuning (ResNet-50, ViT-B/16) yields measurable improvements over a fixed 0.50 threshold (Eff-B0), particularly on F1. (3) *Stability:* validation-selected checkpoints generalize well to the test split, indicating limited overfitting under our recipe. (4) *Implication:* these fixed numbers establish robust image baselines to be used unchanged in subsequent fusion and caption-aware experiments.

B. Clinical Note Classification Results

Approach 1 (PyTesseract OCR) had inferior performance and unstable accuracy (Accuracy = 0.69, F1 = 0.48), therefore it was eliminated from further investigation. We focus on **Approach 2** (AWS Tesseract), which produced technically consistent outputs and improved classification stability.

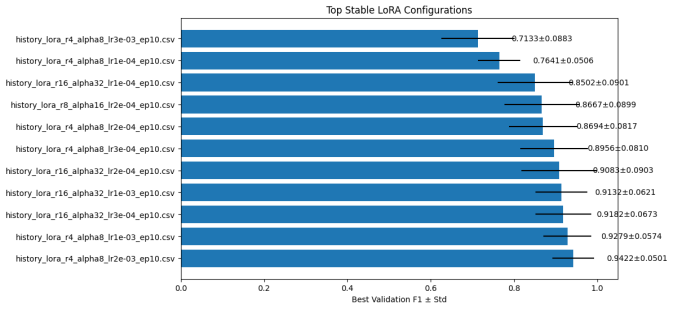

Fig. 4. Top stable LoRA configurations for Approach 2 (AWS Tesseract) showing validation F1 stability across learning rates and parameter settings.

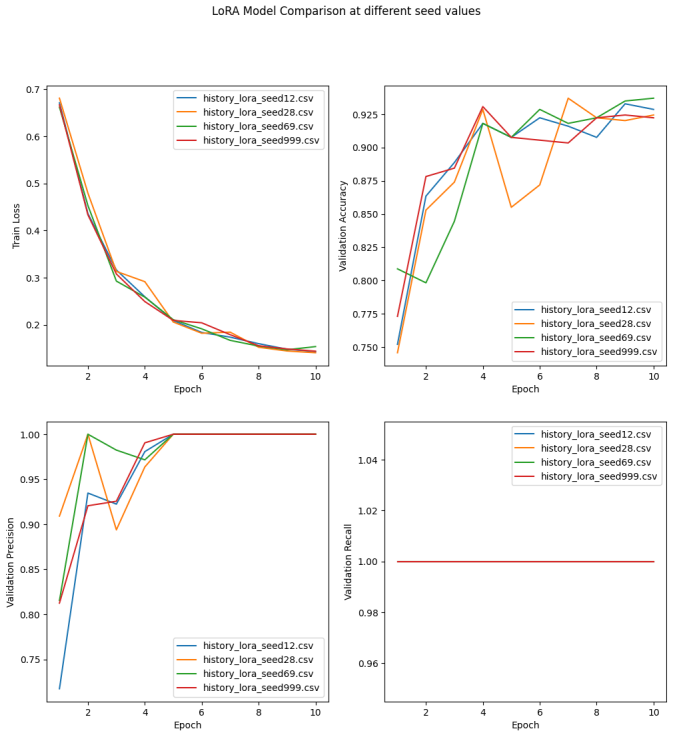

Fig. 5. LoRA model stability across different random seeds confirming reproducible convergence under the optimal configuration.

Figure 4 shows the top LoRA configuration stability across several learning rates and rank-scale combinations, while Figure 5 reveals that LoRA fine-tuning is robust across multiple random seeds. Figures 6 and 7 compare the performance of baseline, TAPT, and LoRA-TAPT models on the test set. They highlight consistent improvements following domain-adaptive pretraining and parameter-efficient fine-tuning.

The optimum configuration (**rank** $r = 4$, $\alpha = 8$, **learning rate = 2e–3, 10 epochs**) had the highest stable validation F1-score (0.94 ± 0.05) and was chosen for clinical note classification in the multimodal fusion experiments.

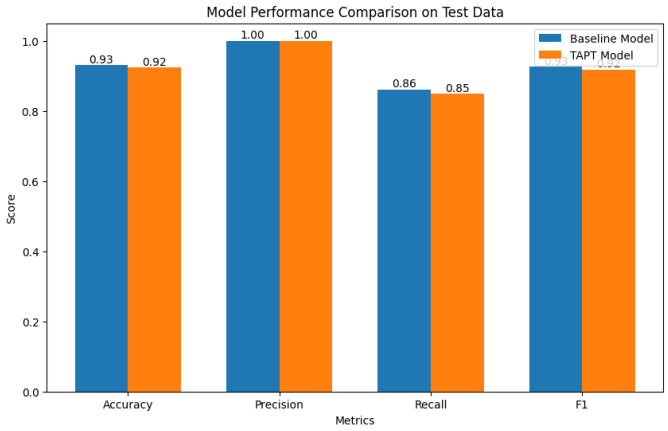

Fig. 6. Performance comparison of baseline and TAPT models on test data for Approach 2 (AWS Tesseract).

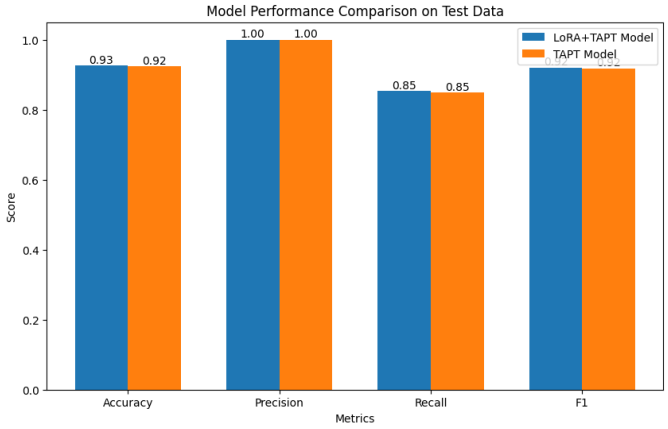

Fig. 7. Performance comparison of LoRA-TAPT and TAPT models under Approach 2 showing incremental gains in accuracy and F1.

TABLE IV. DEEPSEEK JUDGING RESULTS ACROSS 300 SAMPLES.

Model	Wins (out of 300)
GPT-4o	204
GPT-3.5-turbo	28
Claude 3.5 Sonnet	28
Tie	40

C. Caption Generation using RAG and Prompting Techniques – Experimental Results

1) Ground-Truth Selection (GPT-4o via DeepSeek Evaluation): To establish reliable ground-truth captions for RAG-based training, three zero-shot LLMs—**GPT-4o**, **GPT-3.5-turbo**, and **Claude 3.5 Sonnet**—were compared using identical prompts on 300 stratified notes (150 benign / 150 malignant). A fourth model, **DeepSeek**, evaluated all captions on four axes: Clinical Correctness, Evidence Grounding, Relevance & Specificity, and Clarity & Conciseness. As shown in Table IV, GPT-4o was consistently ranked the highest and selected as the **ground-truth source** for subsequent RAG fine-tuning.

TABLE V. TOKEN LENGTH SENSITIVITY ANALYSIS.

Token	R1	R2	RL	B	BF1
40	0.236	0.113	0.183	0.076	0.647
60	**0.264**	**0.127**	**0.196**	**0.076**	**0.674**
100	0.243	0.116	0.194	0.072	0.676

TABLE VI. CONTEXT LENGTH ABLATION (MAX_TOKEN VS. EFFICIENCY).

MaxTok	r	α	EvalLoss	Time/epoch
512	16	64	2.454	1m11s
768	16	64	2.399	1m11s
1024	16	64	2.389	1m22s
2048	**16**	**64**	**2.379**	**1m49s**
5000	16	64	2.365	2m18s
10000	16	64	2.366	3m02s
Full	16	64	2.366	2m39s

TABLE VII. ZERO-SHOT PROMPTING RESULTS (LORA-RAG).

LR	VLoss	R1	R2	RL	B	BF1
2e–5	2.05	0.317	0.164	0.251	0.106	0.804
1e–5	2.20	0.356	0.184	0.264	0.127	0.816
1.5e–5	2.12	0.355	0.184	0.271	0.129	0.835
7e–6	2.28	**0.363**	**0.180**	**0.276**	**0.106**	**0.843**

TABLE VIII. SELF-REFLECTION PROMPTING RESULTS (LORA-RAG).

LR	VLoss	R1	R2	RL	B	BF1
2e–5	2.05	0.256	0.105	0.191	0.056	0.838
1e–5	2.20	0.218	0.084	0.168	0.055	0.820
1.5e–5	2.12	0.247	0.100	0.187	0.060	0.833
7e–6	2.29	**0.219**	**0.083**	**0.169**	**0.050**	**0.816**

TABLE IX. SELF-ASK PROMPTING RESULTS (LORA-RAG).

LR	VLoss	R1	R2	RL	B	BF1
2e–5	2.06	0.300	0.151	0.237	0.090	0.697
1e–5	2.21	0.218	0.101	0.170	0.061	0.657
1.5e–5	2.12	0.227	0.105	0.178	0.063	0.635
7e–6	2.29	**0.216**	**0.103**	**0.170**	**0.080**	**0.685**

2) Metric Abbreviations: For compactness, the following abbreviations are used throughout this section: **R1:** ROUGE-1, **R2:** ROUGE-2, **RL:** ROUGE-L, **B:** BLEU, **BF1:** BERTScore-F1.

3) Token Length Sensitivity: The token-length ablation examined caption outputs of 40, 60, and 100 tokens. As shown in Table V, the 60-token configuration produced the best overall balance between content coverage and conciseness.

4) Context Length Ablation: Context length (max_token) was tested from 512 to full-text inputs. As shown in Table VI, the 2048-token configuration offered the best trade-off between accuracy and computational efficiency.

5) Prompting Technique Performance:

a) Zero-Shot Prompting: Zero-shot prompting served as the baseline for all subsequent experiments. As shown in Table VII, LoRA-RAG achieved the best performance at LR = 7e–6 with R1 = 0.363, RL = 0.276, and BF1 = 0.843.

b) Self-Reflection Prompting: This technique introduced iterative critique and revision stages. While it enhanced factual correctness, it produced moderate gains over the baseline.

c) Self-Ask Prompting: Self-Ask prompting relied on a question–answer reasoning chain. Results in Table IX show improved interpretability but minor quantitative gains.

d) Tree-of-Thought (ToT) Prompting: Tree-of-Thought (ToT) reasoning achieved the highest overall scores, improving

TABLE X. TREE-OF-THOUGHT (TOT) PROMPTING RESULTS (LORA-RAG).

LR	VLoss	R1	R2	RL	B	BF1
2e–5	2.05	0.385	0.198	0.293	0.108	0.859
1e–5	2.21	0.389	0.201	0.300	0.122	0.854
1.5e–5	2.11	**0.409**	**0.214**	**0.320**	**0.130**	**0.862**
7e–6	2.28	0.379	0.191	0.281	0.111	0.852

TABLE XI. BEST PERFORMANCE COMPARISON ACROSS PROMPTING TECHNIQUES.

Technique	R1	R2	RL	B	BF1
Zero-Shot	0.363	0.180	0.276	0.106	0.843
Self-Reflection	0.247	0.100	0.187	0.060	0.833
Self-Ask	0.227	0.105	0.178	0.063	0.635
Tree-of-Thought	**0.409**	**0.214**	**0.320**	**0.130**	**0.862**

lexical overlap and semantic alignment while maintaining clarity and brevity.

6) Comparative Summary: Table XI summarizes the best-performing configurations for each prompting technique. Tree-of-Thought (ToT) achieved the highest overall lexical and semantic alignment, validating reasoning-augmented caption generation.

7) Discussion: Tree-of-Thought prompting exhibited superior performance (R1 = 0.409, RL = 0.320, BF1 = 0.862) under the LoRA-RAG setup (r = 16, α = 64, LR = 7e–6, max_len = 2048). This confirms that structured reasoning enhances factual completeness and semantic alignment in medical captioning.

VI. CONCLUSION

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] D. Horowitz, *End of Time*. New York, NY, USA: Encounter Books, 2005. [E-book] Available: ebrary, http://site.ebrary.com/lib/sait/Doc?id=10080005. Accessed on: Oct. 8, 2008.

[3] D. Castelvecchi, "Nanoparticles Conspire with Free Radicals," *Science News*, vol.174, no. 6, p. 9, September 13, 2008. [Full Text]. Available: Proquest, http://proquest.umi.com/pqdweb?index=52&did=1557231641&SrchMode=1&sid=3&Fmt=3&VInst=PROD&VType=PQD&RQT=309&VName=PQD&TS=1229451226&clientId=533. Accessed on: Aug. 3, 2014.

[4] J. Lach, "SBFS: Steganography based file system," in *Proceedings of the 2008 1st International Conference on Information Technology, IT 2008, 19-21 May 2008, Gdansk, Poland.* Available: IEEE Xplore, http://www.ieee.org. [Accessed: 10 Sept. 2010].

[5] "A 'layman's' explanation of Ultra Narrow Band technology," Oct. 3, 2003. [Online]. Available: http://www.vmsk.org/Layman.pdf. [Accessed: Dec. 3, 2003].

[6] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1455–1462, 2016. doi:10.1109/TBME.2015.2496264

[7] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling, "Rotation equivariant CNNs for digital pathology," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 210–218, 2018. doi:10.1007/978-3-030-00934-2_24

[8] The Cancer Genome Atlas Research Network, "Comprehensive molecular portraits of human breast tumours," *Nature*, vol. 490, no. 7418, pp. 61–70, 2012. doi:10.1038/nature11412

[9] K. Huang, J. Altosaar, and R. Ranganath, "ClinicalBERT: Modeling clinical notes and predicting hospital readmission," in *Proc. Clinical Natural Language Processing Workshop*, pp. 72–78, 2019. arXiv:1904.05342

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685

[11] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, pp. 74–81, 2004. Available: https://aclanthology.org/W04-1013/

[12] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. International Conference on Learning Representations (ICLR)*, 2020. arXiv:1904.09675