SRH Hochschule Heidelberg

Masters Thesis

Vision-Language Modeling for

Chest X-ray Report Generation:

A Cross-Domain Study Using MIMIC-CXR and IU Datasets

Author:
Rakesh Nagaragatta Jayanna

Supervisors:
Prof. Mehrdad Jalali
Prof. Kamellia Reshadi

A thesis submitted in fulfilment of the requirements
for the degree of Master's in Data Science and Analytics

in the

School of Information, Media and Design

September 2025

Declaration of Authorship

I, Name, declare that this thesis titled, 'Graph-Aided Vision-Language Modeling for Chest X-ray Report Generation: A Cross-Domain Study Using MIMIC-CXR and IU Datasets' and the work presented in it is my own. I confirm that this work submitted for assessment is expressed in my own words and is my own. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are appropriately acknowledged at any point of their use. A list of the references employed is included.

- This work was undertaken entirely for a research degree at this university.
- It is clearly stated in this thesis if any part of the work has been previously submitted for a degree or any other qualification at this or any other institution.
- Where I have consulted the published work of others, appropriate credit has been given.
- Where I have quoted from the work of others, the source is always cited, unless it is entirely my own work.
- I have acknowledged all major sources of support and assistance during this work.

Signed: _____

Date: _____

Acknowledgements

I would like to express my sincere gratitude to my thesis supervisors, **Prof. Mehrdad Jalali** and **Prof. Kamellia Reshadi**, for their invaluable guidance, support, and encouragement throughout the course of this research. Their expertise and insights were instrumental in shaping the direction of this work.

I would also like to thank the faculty and staff at **SRH Hochschule Heidelberg** for providing a stimulating and supportive academic environment. Special thanks to my classmates and friends who provided moral support and feedback during the thesis journey.

Finally, I am deeply grateful to my family for their unwavering support, patience, and motivation throughout my academic endeavors.

Rakesh Nagaragatta Jayanna

Abstract

Chest radiograph (CXR) interpretation is one of the most frequent and essential tasks in clinical medicine, yet it is highly dependent on radiologist expertise and time. With increasing imaging volumes and reporting demands, there is a growing need for automated systems to assist in the generation of radiology reports, particularly the critical *Impression* section that guides clinical decision-making. However, building systems that generate medically accurate, fluent, and well-structured reports remains a significant challenge.

In this thesis, I present a two-view Vision Transformer (ViT) encoder combined with a biomedical BART (BioBART) language model for automated chest X-ray report generation. Unlike prior approaches that directly concatenate image features into the decoder, my method introduces a learned image prefix token and leverages BioBART's encoder for prompt conditioning. I fuse frontal and lateral view features to improve diagnostic coverage and stability, and integrate section-aware prompts that reflect clinical structure. Special attention is given to preprocessing artifacts, decoding guardrails, and robust training practices.

The system is trained and evaluated on the MIMIC-CXR dataset, which includes over 89,000 studies and their associated free-text reports. Evaluation focuses not on surface-level similarity metrics (BLEU, ROUGE), but rather clinically grounded performance using CheXbert-based **Clinical F1** scores. Experiments demonstrate that the proposed architecture achieves promising results, including a Clinical F1 score of 0.491 and steady validation loss across epochs. Ablation studies reveal the contribution of prompt engineering, prefix conditioning, and decoding strategies. I also compare my work with a baseline model trained on the IU Chest X-ray dataset and highlight key improvements in generalization and scale.

This work contributes a robust, lightweight, and clinically-informed vision–language pipeline for radiology report generation. I also package the model for deployment and inference, including readiness for public hosting on the Hugging Face Hub. Limitations such as hallucination risks and dataset biases are acknowledged, and future directions include integrating external supervision, clinical labelers, and cross-dataset validation.

Keywords: chest radiography, radiology report generation, vision-language models, ViT, BioBART, Clinical F1, MIMIC-CXR, medical imaging, natural language generation

Contents

List of Figures

List of Tables

Listings

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CXR	Chest X-ray
DICOM	Digital Imaging and Communications in Medicine
F1	Harmonic mean of Precision and Recall
GPT	Generative Pretrained Transformer
LM	Language Model
ML	Machine Learning
NLP	Natural Language Processing
PA/AP	Posteroanterior / Anteroposterior (X-ray views)
RAD	Radiology
RAG	Retrieval-Augmented Generation
ROI	Region of Interest
T5	Text-to-Text Transfer Transformer
ViT	Vision Transformer
BioBART	Biomedical Bidirectional and Auto-Regressive Transformer
MIMIC-CXR	Medical Information Mart for Intensive Care Chest X-ray dataset
IU CXR	Indiana University Chest X-ray dataset

Chapter 1. Introduction

1.1 Background and Motivation

Radiology reports are central to clinical communication. For chest radiography—the most frequently ordered imaging test—the *Impression* section distills salient positives and negatives and often guides management. Drafting impressions is time-consuming and cognitively demanding: the radiologist must integrate clinical context, compare with prior studies, adjudicate uncertainty, and phrase conclusions in a compact, standardized style. Meanwhile, imaging volume grows faster than staffing, and report turnaround time remains under constant pressure. These forces motivate assistive systems that can propose draft impressions while keeping the radiologist firmly in the loop.

At first glance, modern vision–language models seem well suited for this task: they can map images to fluent text. In practice, medical report generation remains challenging for several reasons. First, **factual precision** and **calibrated uncertainty** are non-negotiable; a fluent but wrong sentence can mislead care. Second, radiology prose relies on domain idioms (e.g., "no focal air-space consolidation", "cardiomediastinal silhouette is normal"), hedging, and a predictable section structure that generic models struggle to reproduce consistently. Third, public datasets, even large ones like MIMIC-CXR, still differ from institution-specific style, and they contain artifacts (e.g., de-identification tokens) that can leak into generations if not actively mitigated. Finally, evaluation is difficult: standard overlap metrics (BLEU/ROUGE/METEOR) reward surface similarity rather than clinical correctness, and they correlate imperfectly with expert judgment.

Against this backdrop, I pursue a *pragmatic* approach: build a compact model that respects radiology workflow constraints, emphasize clinically oriented evaluation, and add guardrails that reduce common failure modes (e.g., anonymization artifacts and truncated words). The goal is not to replace expert reporting but to produce *useful drafts* that reduce repetitive typing and provide a consistent starting point for human editing.

1.2 Problem Statement and Scope

Task. Given one or two chest X-ray images from the same study—typically a frontal (PA/AP) view and, if available, a lateral view—and an optional short *Findings* cue, generate a concise *Impression* paragraph.

Inputs.

- **Images:** A frontal radiograph and, when available, a lateral view. If the lateral view is missing at inference, I duplicate the frontal image to preserve the interface contract.
- **Optional cue:** A short text (1–3 sentences) summarizing key observations from the *Findings* section.

Output. A short, readable *Impression* (approximately 2–5 sentences) that enumerates salient positives/negatives and, when appropriate, compares with prior studies.

Scope. I train and evaluate on the MIMIC-CXR dataset (Kaggle mirror) using paired frontal–lateral studies where available. I focus exclusively on the *Impression* section. I prioritize **Clinical F1** (a label micro-F1 over a small lexicon of common thoracic findings) and train/validation loss as primary signals of progress; I intentionally *do not* emphasize BLEU/ROUGE/METEOR because they mainly measure word overlap rather than medical correctness. I also address deployment considerations: deterministic decoding, artifact guardrails, and packaging.

1.3 Research Questions and Hypotheses

I organize the thesis around five questions:

RQ1: **Integration.** How should image features be coupled to a biomedical encoder–decoder so that generated impressions are coherent and clinically plausible? *Hypothesis:* Running the language model's *encoder* on a concatenation of an *image prefix token* and a short, section-aware prompt is more stable and coherent than naively concatenating raw embeddings into the decoder.

RQ2: **Two-view fusion.** Does explicitly fusing frontal and lateral features (vs. using only a frontal view) improve clinical signal for the impression? *Hypothesis:* Even a simple fusion (e.g., averaging CLS tokens) transfers useful complementary cues (e.g., retrosternal clear space, posterior effusions).

RQ3: **Decoding controls.** Which decoding constraints (minimum length, no-repeat n-gram, repetition penalty) and prompts best reduce loops, generic language, and truncated words? *Hypothesis:* A modest set of constraints, plus a section-aware prompt, substantially improves readability with little computational cost.

RQ4: **Guardrails.** Can simple generation-time guardrails (token blocklists and regex normalization) effectively suppress de-identification artifacts and unfinished words without retraining? *Hypothesis:* Yes; targeted guardrails yield clear quality gains.

RQ5: **Evaluation.** How well do Clinical F1 and loss track the improvements that radiologists care about? *Hypothesis:* Clinical F1 over a small, operational label set correlates better with utility than word-overlap metrics.

1.4 Approach Overview

I couple a ViT image encoder (pretrained on chest radiographs) with a biomedical BART (BioBART) encoder–decoder. The pipeline has four key stages:

(1) Two-view feature fusion.I encode the frontal and lateral images with ViT, extract their CLS embeddings, and compute a simple fusion (average). A learned linear projection maps the fused vector into the decoder's hidden size; I treat this vector as a single learned *image prefix token*.

(2) Section-aware conditioning. To stabilize tone and structure, I prepend a compact prompt that anchors the style of the output:

```
FINDINGS: {optional cue} SECTION: IMPRESSION
```

When no cue is provided, I use the `SECTION: IMPRESSION` part alone.

(3) Corrected encoder–decoder integration. Rather than feeding raw embeddings directly into the decoder (which I found brittle and prone to degenerate strings), I run the BioBART *encoder* on the concatenation of the image prefix token and the tokenized prompt, and then let the decoder attend to these contextualized encoder states. This single correction markedly improves coherence and reduces unfinished or repeated tokens.

(4) Decoding controls and guardrails.I employ conservative decoding during validation (greedy or low-beam), a minimum number of new tokens to avoid ultra-short outputs, `no_repeat_ngram_size` to suppress loops, and a light `repetition_penalty`. I add `bad_words_ids` to block anonymization placeholders and normalize outputs post hoc with regexes to fix spacing, punctuation, and "x-XXXX $\rightarrow$ x-ray" artifacts.

1.5 Design Principles, Assumptions, and Non-Goals

Human-in-the-loop. The system is intended to *assist*, not replace, radiologists. Outputs are drafts that must be reviewed and edited. I optimize for readability and consistency under tight computational budgets.

Simplicity under constraints. Where possible, I prefer simple mechanisms (average fusion, single-token prefix, short prompts) that are easy to audit and deploy. I avoid heavy architectural additions that complicate training and inference unless they offer clear, clinically meaningful gains.

Assumptions.(i) Paired image–report data are representative of the target prose style; (ii) a short *Findings* cue can be provided or derived; (iii) missing lateral views are common and

acceptable to approximate by duplicating the frontal view at inference so that the interface remains consistent.

Non-goals.I do not pursue full *Findings* generation, report structuring beyond the *Impression*, nor autonomous diagnostic use. I also do not claim generalization to all institutions without local calibration and validation.

1.6 Contributions

1. **A compact two-view ViT–BioBART architecture** with a *learned image prefix token* and a **corrected encoder–decoder integration** (encoder consumes prefix+prompt) that improves coherence over naive embedding concatenation.
2. **Section-aware prompting** that stabilizes tone with minimal engineering and reduces generic "no acute process" outputs.
3. **A robust training + decoding recipe** including staged unfreezing, cosine decay with warmup, gradient clipping, minimum output length, `no_repeat_ngram_size`, and light repetition penalty.
4. **Generation-time guardrails**—token blocklists for de-identification artifacts and regex normalization—that clean outputs without additional training.
5. **Clinically oriented evaluation** centered on **Clinical F1** and train/validation loss; I *intentionally avoid* relying on BLEU/ROUGE/METEOR for model selection because they reward word overlap rather than medical correctness.
6. **Reproducible, deployment-ready packaging** with deterministic decoding settings and saved components (encoder, decoder, tokenizer, projection, and hyperparameters).

1.7 Thesis Organization

The remainder of this thesis is organized as follows:

- **Chapter 2** — Reviews prior work in medical report generation, vision backbones for radiography, and biomedical sequence-to-sequence models.
- **Chapter 3** — Describes the MIMIC-CXR dataset (Kaggle mirror), cohort construction, and preprocessing steps for both text and images, including artifact normalization.
- **Chapter 4** — Details the two-view fusion architecture, learned image prefix, corrected encoder–decoder integration, as well as prompts, decoding strategies, and guardrails.
- **Chapter 5** — Explains optimization setup, hyperparameters, staged unfreezing of the vision backbone, and stability practices during training.
- **Chapter 6** — Presents learning curves (train vs. validation loss), **Clinical F1** across epochs, and qualitative examples with ablation insights.
- **Chapter 8** — Documents model packaging, inference-time configurations, and deployment optimizations for latency and robustness.

- **Chapter 9** — Discusses limitations, ethical considerations, and external validity of the system.
- **Chapter 10** — Concludes the work and outlines future directions, including clinically grounded supervision and cross-dataset validation.

Chapter 2. Related Work

2.1 Medical Report Generation

2.1.1 From Captioners to Clinical Narratives

Automatic radiology report generation grew out of image captioning. Early systems paired convolutional encoders with recurrent decoders (LSTM/GRU) and attention to align visual regions and words. Representative models such as TieNet and HRGR-Agent introduced hierarchical text planning and reinforcement-style reward shaping to produce paragraph-level outputs and promote medical phrasing Jing et al. (2018), Wang et al. (2018). While these methods improved fluency over naive captioners, they struggled to consistently name clinically critical entities (e.g., *pleural effusion*, *pneumothorax*) or express uncertainty appropriately.

2.1.2 Transformer-Based Generators

Transformer architectures (e.g., R2Gen, M2TR) brought stronger long-range language modeling and multi-head attention to radiology reporting Chen et al. (2020), Shen et al. (2021). These models improved cross-sentence coherence and reduced local repetition compared to RNNs. However, factuality remained a central limitation: models could produce grammatically polished yet clinically incorrect statements, especially when supervision came solely from noisy free-text reports.

2.1.3 Template and Retrieval Augmentation

To curb hallucination, several works integrate retrieval or templating. Retrieval-augmented generators condition on similar historical reports or snippets found at inference time, while template-guided methods fill physician-authored skeletons with model predictions Yang et al. (2021). These strategies reduce free-form drift and can boost precision for common findings, but they trade flexibility and may encode institutional style biases. When cases deviate from the template, important rare findings can be omitted.

2.1.4 Clinical Supervision and Knowledge Signals

A parallel line injects clinical signals to align generation with medical semantics. Approaches include optimizing against auto-labelers such as CheXpert/CheXbert to reward correct entity mentions Smit et al. (2020), leveraging entity/relation graphs like RadGraph during training or evaluation Jain et al. (2021), and coupling generation with auxiliary classification heads to predict key abnormalities. These techniques push models toward *what* to say rather than only *how* to say it.

2.1.5 Instruction-Tuned and Large Multimodal Models

General-purpose large language/multimodal models (LLMs/LMMs) have been adapted to radiology via instruction tuning and multimodal adapters (e.g., Flamingo-style resamplers, Q-formers, Perceiver bridges) Alayrac et al. (2024), Li et al. (2023). Despite impressive fluency, they often lack chest X-ray–specific pretraining and require careful guardrails in clinical contexts. Integrating vision features in a way that yields precise, concise *Impression* text remains non-trivial.

Position of this thesis.I pursue a compact, domain-adapted path: a ViT encoder trained on chest radiographs, a biomedical BART decoder for clinical style/terminology, and a *corrected* encoder–decoder coupling. I focus on two-view fusion, clinically oriented decoding controls, and label-aware evaluation rather than maximizing generic n-gram metrics.

2.2 Vision Encoders for Chest Radiography

2.2.1 CNN Backbones and the Rise of ViT

Classical chest X-ray encoders used CNNs (DenseNet/ResNet) trained on large CXR corpora to predict labels, and these features seeded early generators. Vision Transformers (ViT) replaced convolution with self-attention, enabling flexible global receptive fields that better capture distributed patterns such as interstitial edema or cardiomegaly Dosovitskiy et al. (2021). ViT variants pretrained on radiographs consistently transfer to downstream detection and reporting tasks.

2.2.2 Self-Supervised and Cross-Modal Pretraining

Self-supervised learning (contrastive SimCLR/MoCo/BYOL; masked image modeling MAE/BeiT) has been adapted to CXR. Cross-modal SSL (ConVIRT, GLoRIA, BioViL, MedCLIP) aligns images with paired text to learn representations that map naturally to clinical entities Boecking et al. (2022), Huang et al. (2021), Wang et al. (2022), Zhang et al. (2021). These often outperform purely supervised pretraining when labeled data are limited.

2.2.3 Two-View Fusion: Frontal and Lateral

Clinically, radiologists synthesize frontal (PA/AP) and lateral views, yet many systems ignore laterals. Prior work ranges from simple averaging/concatenation of view embeddings to attention-based fusion **?**. Even naive averaging can yield gains; attention lets the model emphasize view-specific cues (e.g., small pleural effusions on lateral). Multi-view modeling is a pragmatic way to lift performance without enlarging datasets.

Position of this thesis.I adopt a two-view ViT and fuse CLS tokens via averaging before projecting into the text model's hidden size. The fused vector becomes a single learned *image prefix token* that the language model can reliably condition on.

2.3 Biomedical Language Models for Radiology Text

2.3.1 Domain-Adaptive Encoders and Decoders

General-domain LMs (BERT/BART/T5) generate fluent text but often miss clinical idioms. Domain-adapted encoders (BioBERT, ClinicalBERT, PubMedBERT) and encoder–decoders (BioBART) improve biomedical style, negation, and uncertainty handling by pretraining on PubMed/PMC/clinical notes Lee et al. (2020), Yuan et al. (2022). For radiology, this adaptation helps with phrases like "cardiomediastinal silhouette is normal" and calibrated negatives ("no focal consolidation").

2.3.2 Where to Inject Vision?

Common strategies include: (i) encoder-side fusion—concatenating learned visual tokens with textual tokens at the encoder input; (ii) decoder-side cross-attention to frozen visual features; and (iii) bottleneck projectors (Q-former/Perceiver) that format vision signals into language-compatible tokens Li et al. (2023). For report-like outputs, encoder-side fusion often stabilizes semantics; decoder-only injection can be brittle without strong regularization.

Position of this thesis.I project fused two-view ViT features into the BioBART hidden space as a single token and feed it to the *encoder* alongside a short, section-aware prompt. This corrected coupling reduced degeneracy (e.g., garbled or unfinished words) seen when directly perturbing decoder embeddings, and it yielded more stable training and inference.

2.4 Evaluation: Beyond Word Overlap

2.4.1 Limitations of BLEU/ROUGE/METEOR

BLEU, ROUGE-L, and METEOR quantify lexical overlap and are widely reported Chen et al. (2020). In clinical text they can be misleading: a model may replicate phrasing while missing a key finding, or describe the wrong laterality yet score reasonably due to n-gram overlap. These metrics primarily reward surface similarity, not medical correctness.

2.4.2 Clinically Oriented Metrics

Researchers therefore complement overlap metrics with auto-labelers (CheXpert/CheXbert) to estimate label precision/recall, micro-F1 over a small lexicon (effusion, edema, consolidation, cardiomegaly, atelectasis, pneumothorax), and entity/relation metrics like RadGraph F1 Jain et al. (2021), Smit et al. (2020). These are imperfect but closer to what radiologists care about—mentioning the right abnormalities and avoiding spurious claims.

2.4.3 Human and Hybrid Evaluation

Expert review remains the gold standard: radiologists assess correctness, completeness, and clarity with structured rubrics. Hybrid protocols combine auto-metrics for scale with targeted expert evaluation for high-stakes assessment. NLI-style factuality checks are promising but sensitive to domain shift and limited medical NLI resources.

Position of this thesis.I prioritize a clinical label overlap metric (**Clinical F1**) and loss curves, and explicitly de-emphasize BLEU/ROUGE/METEOR in my conclusions.

Summary

The literature has progressed from CNN–RNN captioners to Transformer-based generators, retrieval/template augmentation, and domain-adapted LMs. Nevertheless, factuality and clinical grounding remain challenging. This thesis builds on ViT encoders, two-view fusion, and biomedical encoder–decoders, with a corrected encoder–decoder integration and clinically oriented evaluation.

Chapter 3. Datasets, Curation, and Preprocessing

3.1 Overview

This chapter documents the data sources, cohort construction, and the exact preprocessing pipeline I used to train and evaluate the two-view chest X-ray impression generator. My primary corpus is MIMIC-CXR with paired radiographs and free-text reports, accessed via the Kaggle mirror for convenience. I curate patient-disjoint splits, pair frontal and lateral views when available, and normalize the *Impression* section to serve as the generation target. I also implement guardrails to mitigate de-identification artifacts and inconsistent sectioning that commonly appear in clinical corpora.

3.2 Primary Dataset: MIMIC-CXR (Kaggle Mirror)

I use the publicly available MIMIC-CXR corpus of chest radiographs and reports Johnson et al. (2019). For simpler path handling and JPEG storage, I rely on the Kaggle mirror:

- **Kaggle URL:** `https://www.kaggle.com/datasets/simhadrisadaram/mimic-cxr-dataset`
- **Contents:** JPEG images under patient subfolders (e.g., `p10...p19`) and two CSV files covering metadata and report text.
- **Structure:** multiple images may belong to the same *study* (typically a frontal PA/AP, often a lateral). Each study links to a report containing sections such as *Indication*, *Findings*, and *Impression*.

Why MIMIC-CXR?Compared to smaller corpora (e.g., IU Chest X-ray, Open-i), MIMIC-CXR offers substantially more studies and stylistic diversity, which improves the robustness of encoder–decoder training. The trade-off is noisier text (de-identification placeholders, irregular sections), which I address through targeted normalization (Section 3.4).

3.3 Cohort Construction and Splitting

I construct a study-level dataset with explicit two-view pairing and patient-disjoint splits:

1. **Study grouping.** I group all images and the report by study identifier. If multiple frontal images exist, I select the best available according to metadata precedence; for lateral, I prefer standard lateral views when present.

2. **Two-view pairing.** Each training sample contains a frontal image and, when available, a lateral image. If the lateral is missing, I duplicate the frontal so the model interface remains consistent (two tensors).
3. **Target section.** I select *Impression* as the generation target when present; if it is empty, I fall back to *Findings*. I explicitly log coverage to quantify how often I rely on a fallback.
4. **Patient-disjoint splits.** To avoid leakage, I create train/validation splits at the *patient* level (no patient appears in more than one split). I fix random seeds and write split lists to disk for reproducibility.

3.4 Preprocessing Pipeline

The preprocessing mirrors the training code and is organized into robust, auditable steps.

A. File Discovery and Path Resolution

1. **Indexing.** I crawl the image root once to build a map {`basename` → full path}, which lets me resolve paths referenced in CSVs even if they are partial or malformed.
2. **CSV parsing.** Some rows store image lists as JSON-like blobs. I safely parse them (e.g., `ast.literal_eval`) and select the first valid path per study/view.
3. **Canonicalization.** I fix double prefixes (e.g., `/files/files/`), strip quotes, join with the configured `IMG_ROOT`, and if a path does not exist, I fall back to the index by basename.

B. Study Filtering and Two-View Pairing

1. **Minimum requirement.** Keep a study if it contains a frontal image; prefer pairing with a lateral from the same study.
2. **Inference consistency.** When the lateral is missing, I duplicate the frontal path into the lateral slot to keep shapes consistent for the two-branch encoder.
3. **Pairs table.** I create a tidy table: {`frontal_path`, `lateral_path`, `target_text`}, which the PyTorch `Dataset` reads directly.

C. Target Text Selection and Normalization

1. **Section priority.** I search candidate columns in order: `impression/impressions` → `report/summary` → `findings`. The first non-empty field per study becomes the target.
2. **Normalization.** I perform:
   - whitespace and punctuation normalization; strip HTML breaks (`<br>`);
   - anonymization repair (e.g., `x-XXXX` → "x-ray");
   - removal of stray quotes/brackets introduced by CSV formatting.
3. **Placeholders.** If the final target is empty after cleaning, I insert a safe placeholder ("No acute cardiopulmonary abnormality.") during training to prevent degenerate loss spikes.

D. Image Transforms and Tensorization

1. **Decoding.** Read with PIL, convert to RGB.
2. **Resize.** Resize to 224×224.
3. **Normalize.** Scale to $[-1, 1]$ with mean=0.5 and std=0.5.
4. **Light augmentation.** On training only: small rotations ($\leq 5°$) and mild brightness/-contrast jitter to improve robustness without changing pathology semantics.
5. **Batching.** Each batch is a tuple: (frontal tensor, lateral tensor, findings prompt, target impression).

E. Prompt Construction

I anchor generation with a short, section-aware prompt:

`FINDINGS: {cleaned findings if available}  SECTION: IMPRESSION`

If *Findings* is absent, I use `SECTION: IMPRESSION`. This consistently stabilizes tone and discourages rambling openings.

F. Guardrails at Generation Time

- **Bad words.** I block de-identification tokens (e.g., `XXXX`, malformed "x-xxxx") via `bad_words_ids`.
- **Regex cleanup.** After decoding, I apply a final regex pass to collapse spaces, fix dashes, and normalize anonymization residues.
- **Stable decoding.** I set `pad/eos_token_id` consistently to avoid NaNs and ensure deterministic greedy settings for validation metrics.

3.5 Domain Shift and External Validity

Style, acquisition protocols (PA vs. AP, portable vs. upright), and pathology prevalence vary by institution and service line. A model that performs well on MIMIC-CXR may degrade elsewhere. I therefore (i) use patient-disjoint splits, (ii) prioritize clinically oriented metrics (Clinical F1) and loss curves, and (iii) treat external validation and reader studies as mandatory future work.

3.6 Reproducibility

I fix seeds, write split indices to disk, and print coverage diagnostics (kept vs. dropped studies, section availability). The code path from CSVs to batched tensors is fully scripted so that runs can be audited end-to-end.

Chapter 4. Method

4.1 Problem Setup

Given a frontal and (optionally) lateral chest X-ray image and an optional short *Findings* cue, the task is to generate a concise, clinically coherent *Impression* paragraph. I model this as conditional sequence generation with a two-branch vision encoder and a biomedical encoder–decoder for text.

4.2 Architecture Overview

Figure 4.1 illustrates the full pipeline:

1. **Two-view ViT encoder.** A Vision Transformer (ViT) processes the frontal and lateral images independently and outputs per-view embeddings.
2. **Fusion and projection.** The `[CLS]` tokens from both views are averaged and projected via a linear transformation to obtain an image prefix token.
3. **Prompt concatenation and BioBART decoding.** The prefix token is prepended to a section-aware textual prompt and passed through a BioBART encoder–decoder to generate the impression.

4.3 Detailed Architecture Description

4.3.1 Two-View ViT Encoder

I adopt a shared ViT backbone pretrained on chest radiographs. Each input image (frontal or lateral) is resized, normalized, and passed separately through the ViT. I extract the final `[CLS]` token from each stream, denoted $\mathbf{f}$ and $\mathbf{l}$ respectively. These serve as compact, image-level feature representations.

$$\mathbf{p} = \tfrac{1}{2}(\mathbf{f} + \mathbf{l}) \tag{4.1}$$

This late fusion strategy avoids cross-view attention or concatenation, reducing overfitting risk and enabling flexible handling of missing views (e.g., only frontal available).

4.3.2 Average Fusion Layer

Average fusion ensures a balanced contribution from both image views. Unlike concatenation, which doubles the embedding size, or multi-head fusion, which adds parameters, averaging

is efficient and effective. It also keeps the image representation dimension consistent with downstream layers.

When only one view is available at inference time (e.g., lateral missing), I reuse the frontal embedding:

$$\mathbf{p} = \mathbf{f} \quad \text{(if lateral unavailable)} \tag{4.2}$$

4.3.3 Linear Projection and Image Prefix Token

To bridge vision and language modalities, the fused vector $\mathbf{p}$ is projected into the BioBART hidden space via a learned linear transformation:

$$\mathbf{h} = \mathbf{W}\mathbf{p} + \mathbf{b}, \quad \mathbf{h} \in \mathbb{R}^{d_{\text{text}}} \tag{4.3}$$

This projected vector $\mathbf{h}$ becomes the **image prefix token**, a single learned embedding prepended to the text prompt input.

4.3.4 Section-Aware Prompt Construction

The language model is guided by a structured prompt containing medical sections. The prompt template is:

```
FINDINGS: <optional cue>
SECTION: IMPRESSION
```

This enforces alignment with clinical writing style and helps the decoder distinguish between contexts. The prompt is tokenized and concatenated with the prefix token:

$$[\mathbf{h} \| \text{Prompt Tokens}] \rightarrow \text{BioBART Encoder} \tag{4.4}$$

4.3.5 BioBART Encoder–Decoder

The combined visual+textual input is passed to the BioBART encoder. BioBART is a domain-adapted BART model pretrained on biomedical text. The encoder generates contextual embeddings that capture both visual and prompt semantics.

The BioBART decoder is autoregressive and generates the `Impression` section token by token, attending to encoder outputs. Clinical tone and structure are preserved by domain pretraining and prompt anchoring.

4.4 Mathematical Summary of the Flow

Let x_f and x_l denote the input frontal and lateral images.

$$\begin{aligned}
\mathbf{f} &= \text{ViT}(x_f)[\texttt{[CLS]}] \\
\mathbf{l} &= \text{ViT}(x_l)[\texttt{[CLS]}] \\
\mathbf{p} &= \frac{1}{2}(\mathbf{f} + \mathbf{l}) \\
\mathbf{h} &= \mathbf{W}\mathbf{p} + \mathbf{b} \\
\text{Input} &= [\mathbf{h} \| \text{Prompt}] \\
\hat{y} &= \text{Decoder}(\text{Encoder}(\text{Input}))
\end{aligned}$$

4.5 Decoder Control and Stability

Decoding behavior is controlled via:

- **Beam search** (1–4 beams) to trade off diversity vs. stability.
- **Minimum length** to enforce full sentences.
- **Anti-repetition** penalties and n-gram constraints.
- **Blacklist filtering** to suppress artifacts like anonymization tags.

4.6 Training Objective and Optimization

The model is trained end-to-end with a cross-entropy loss over the generated impression tokens:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, x) \tag{4.5}$$

Optimization Strategy:

- Optimizer: AdamW
- Learning rates: lower for ViT, higher for projection and decoder
- Scheduler: Cosine decay with warmup
- Gradient clipping and accumulation for batch stability

Stage-Wise Unfreezing: During epoch 1, the ViT is frozen. From epoch 2 onward, the last ViT block is unfrozen to allow gradual visual fine-tuning.

4.7 Figure

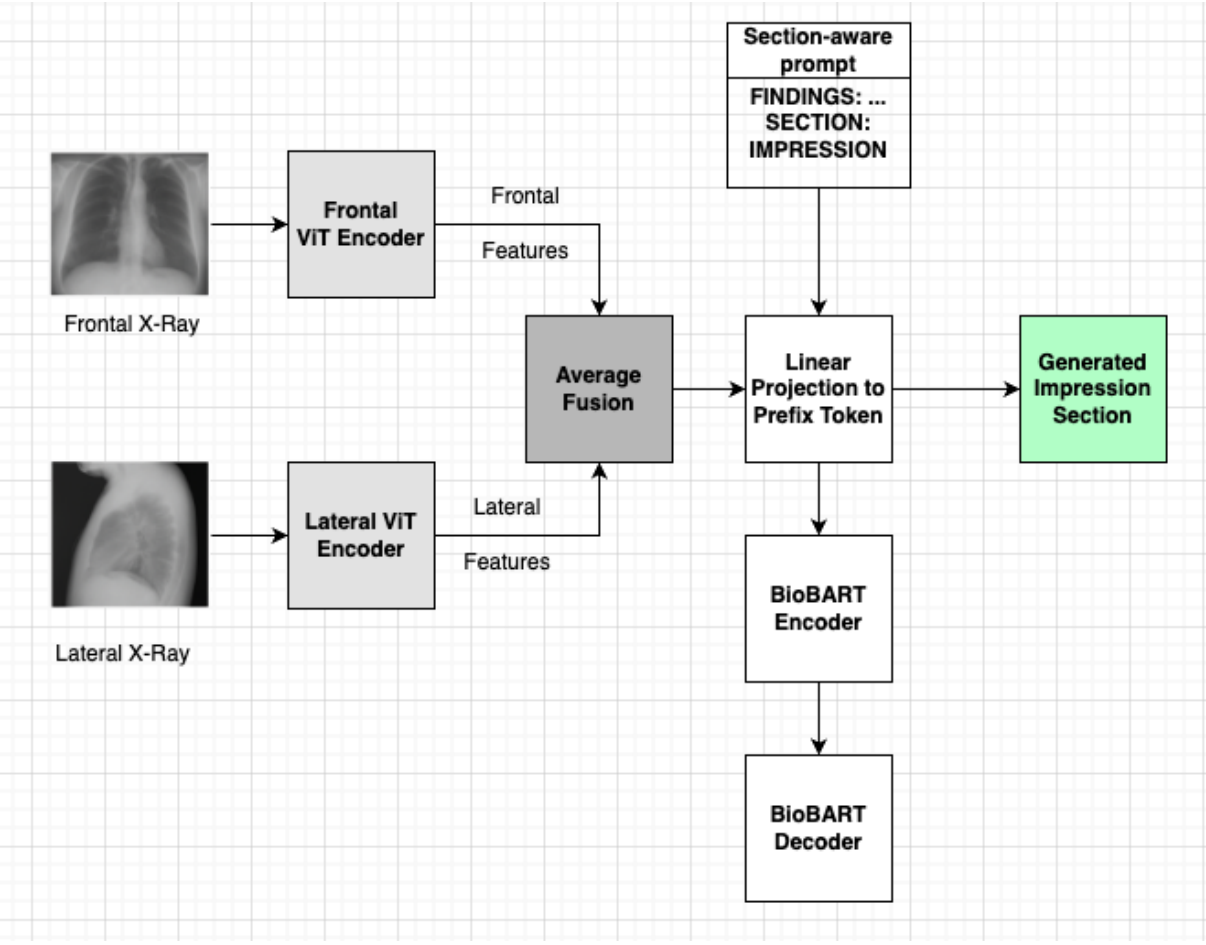

FIGURE 4.1: Schematic of the proposed method: Frontal and Lateral chest X-rays are encoded using ViT. The `[CLS]` tokens are averaged, projected into the decoder space, and prepended to a section-aware prompt. The sequence is passed through a BioBART encoder–decoder to generate the `Impression` section.

Chapter 5. Optimization and Training Strategy

This chapter details how I train the two–view ViT–BioBART system: the optimization setup, hyperparameters, staged unfreezing of the vision backbone, validation and checkpointing policy, and the stability practices that made the runs reliable on MIMIC–CXR. The design mirrors the executable training cell used in this thesis; optional variations are clearly marked as such.

5.1 Training Loop Overview

I use teacher–forced sequence–to–sequence learning on the *Impression* target. Each mini–batch performs:

1. **Two–view encoding.** Frontal and lateral images are passed through a shared ViT encoder. I take each view's `[CLS]` token, average them, and project to the BioBART hidden size to obtain a single *image prefix token*.
2. **Prompt construction.** I build a short section–aware prompt

   `FINDINGS: {...} SECTION: IMPRESSION`

   (or the minimal anchor `SECTION: IMPRESSION` if findings are unavailable).
3. **Encoder conditioning.** I feed [prefix‖prompt tokens] to the BioBART *encoder* and decode the target impression with teacher forcing.
4. **Loss and step.** I optimize token–level cross–entropy; gradients are clipped and stepped with AdamW under a cosine schedule with warmup.

A compact pseudocode sketch:

[1] Freeze all ViT params (epoch 1) epoch $= 1 \dots E$ epoch $= 2$ Unfreeze last ViT block batch $(x_f, x_l, \text{prompt}, y)$ $h_f \leftarrow \text{ViT}(x_f).[\text{CLS}]$, $h_l \leftarrow \text{ViT}(x_l).[\text{CLS}]$ $h \leftarrow \text{Proj}\left(\frac{h_f + h_l}{2}\right)$ image prefix token $\mathbf{H}_{\text{enc}} \leftarrow \text{BioBART\_enc}([h \| \text{prompt}])$ $\hat{y} \leftarrow \text{BioBART\_dec}(\mathbf{H}_{\text{enc}}, y_{<t})$ $\mathcal{L} \leftarrow -\sum_t \log p(\, y_t \mid \hat{y}_{<t}\,)$ clip gradients; AdamW step; cosine schedule step Validate with *greedy* decoding; save best–by–Clinical F1

5.2 Objective

Let x_f, x_l denote frontal and lateral images, p the prompt tokens, and $y = (y_1, \ldots, y_T)$ the impression. The objective is

$$\mathcal{L}_{\text{seq}} = -\sum_{t=1}^{T} \log p_\theta\big(y_t \,\big|\, y_{<t},\ \text{Enc}([\,\text{prefix}(x_f, x_l)\,]\|p)\big).$$

I do *not* use label smoothing or auxiliary losses in the final runs, prioritizing stability and reproducibility.

5.3 Optimizer, Schedule, and Parameter Groups

I employ AdamW with discriminative learning rates:

- ViT backbone: $\eta_{\text{ViT}} = 3 \times 10^{-5}$
- Projection + BioBART (encoder & decoder): $\eta_{\text{text}} = 3 \times 10^{-4}$

Additional settings:

- Weight decay: 0.01
- Gradient clipping: $\|\nabla\|_2 \leq 1.0$
- Warmup: 6% of total steps, followed by cosine decay to 0
- Gradient accumulation: configurable (I used 1; increase to emulate larger batches)

Rationale. The lower LR on ViT protects pretrained radiographic features; the higher LR on the text stack lets the encoder–decoder quickly adapt to clinical phrasing. Warmup+*cosine* reduces loss spikes early in training.

5.4 Staged Unfreezing of the Vision Backbone

I train the text stack first while keeping vision stable, then allow a small amount of visual adaptation:

Epoch 1: *Freeze* the entire ViT.

Epoch 2 →: *Unfreeze only the last ViT transformer block.* The code detects block indices across common ViT implementations.

This strategy preserves robust low–level features while enabling clinically meaningful refinement (e.g., sensitivity to effusion silhouettes) without over–fitting the entire backbone.

Optional variant. If resources permit and validation loss keeps improving after epoch 3, unfreeze the last *two* ViT blocks; I treat this as an extension rather than a default.

5.5 Data Loading, Batching, and Conditioning

5.5.1 Two–View Pairing

Each training example contains a frontal and a lateral image. If a lateral view is missing, I duplicate the frontal image so that the forward signature remains constant. This mirrors inference behavior.

5.5.2 Image Transforms

Images are read with PIL, forced to RGB, resized to 224×224, and normalized to $[-1, 1]$ (mean= 0.5, std= 0.5). I apply *light* augmentation for training (small rotations and mild brightness/contrast jitter) to improve robustness without changing clinical semantics.

5.5.3 Prompts and Targets

For every sample I build a short, section–aware prompt. Targets are cleaned by stripping HTML breaks, collapsing whitespace, and converting anonymization remnants (e.g., `x-XXXX` → "x–ray"). If a target becomes empty after cleaning, I inject a short safe placeholder ("No acute cardiopulmonary abnormality.") to avoid degenerate batches.

5.6 Validation, Selection Metric, and Checkpointing

Validation uses *greedy* decoding (no sampling/beam diversity) to produce low–variance metrics. I compute:

- **Primary metric: Clinical F1** (micro–F1 over a small lexicon: pleural effusion, pneumothorax, edema, consolidation, atelectasis, cardiomegaly, nodule, mass, fracture, emphysema).
- **Secondary diagnostics:** BLEU/ROUGE–L/METEOR (reported but de–emphasized).
- **Loss curves:** train vs. validation loss per epoch.

I save checkpoints *by best Clinical F1*, and also package the final–epoch weights for completeness.

5.7 Validation Decoding Settings

To prevent pathological strings and incomplete words during validation I set:

- `do_sample`=`False`, `num_beams`=1 (fully greedy)
- `no_repeat_ngram_size`= 3, `repetition_penalty`≈ 1.05
- `min_new_tokens`≈ 40, `max_new_tokens`≈ 110
- Explicit `pad_token_id` and `eos_token_id`
- `remove_invalid_values`=`True`

Greedy decoding avoids over–optimistic lexical metrics and tends to yield better Clinical F1.

TABLE 5.1: Key hyperparameters employed in this thesis.

Component	**Setting (used)**	**When to change**
Optimizer	AdamW	Keep
η_{ViT}	3×10^{-5}	$\downarrow$ if overfitting; $\uparrow$ if flat after unfreeze
η_{text}	3×10^{-4}	$\downarrow$ if too generic; $\uparrow$ if slow
Weight decay	0.01	0.01–0.05
Warmup	6% of steps	Increase if epoch 1 is noisy
Clip norm	1.0	0.5–1.0
Epochs	3	Extend to 5–8 if loss keeps dropping
Grad. accum.	1	Raise to emulate larger batch
Val decoding	Greedy	Adjust min/max length as needed

5.8 Stability Practices

The following safeguards materially improved reliability:

1. **Pad/EOS discipline.** Many biomedical tokenizers lack a `pad_token`. I set `tokenizer.pad_token` and mirror it into the decoder config to prevent misaligned masks or crashes.
2. **FP32 training.** I disable mixed precision (AMP) for training to avoid rare FP16 NaNs on this dataset. Inference can still use autocast safely.
3. **Nonempty targets.** Placeholders prevent unstable cross–entropy and degenerate decoding when labels are blank.
4. **Guarded metrics.** BLEU/ROUGE/METEOR calls are wrapped in `try/except` so a malformed sample cannot kill validation.
5. **Decode guardrails.** `remove_invalid_values=True` eliminates the common multinomial error (NaN/Inf probabilities). I also normalize Unicode (drop `\textbackslashufff d`) and strip `<br>` variants.
6. **Determinism where helpful.** Fixed seeds for Python/NumPy/PyTorch; `torch.backends.cudnn.benc` to speed up preprocessing convolutional ops; deterministic data shuffling per epoch.
7. **Checkpoint by Clinical F1.** This aligns model selection with clinical utility rather than with minima that correspond to verbose but vague text.

5.9 Hyperparameters Used and Tuning Advice

Two practical tweaks. If Clinical F1 plateaus:

1. Raise `min_new_tokens` slightly (e.g., 48–56) to improve recall on short labels.
2. Try `no_repeat_ngram_size`= 2 to allow benign bi–grams while still suppressing loops.

5.10 Throughput Tips (No Behavioral Change)

- Use `pin_memory=true` and tune `num_workers` (4–8 typical); prefetch if available.
- Cache prompt tokenizations if they are deterministic.

- Consider gradient checkpointing only if the decoder becomes the memory bottleneck (not used in my runs).
- Use AMP only for *inference*; keep training FP32 for stability.

5.11 Troubleshooting

"probability tensor contains NaN/Inf" during generation. Set `remove_invalid_values=true`; ensure valid `pad/eos` IDs; avoid sampling in validation.

CUDA device–side asserts after adding auxiliary heads. Typically caused by label shape/device mismatches or class–index overflow. I removed auxiliary heads in the final recipe; if re–adding, carefully construct multi–hot labels on the correct device and sanitize class counts.

Constrained beam conflicts. `force_words_ids` is incompatible with grouped beams. Either set `num_beam_groups=1` or disable constraints. I do *not* use constrained decoding in the final setup.

Garbled tokens or "___". Provide `bad_words_ids` for placeholders (e.g., ____, `XXXX`) and apply a final regex normalizer. Prefer encoder–side prefix injection over perturbing decoder embeddings.

5.12 What I Tried but Did Not Keep

- **Sampling or large beams in validation:** improved surface metrics, reduced Clinical F1 and increased variance.
- **Length penalty** > 1.3 **with beams** ≥ 8**:** slightly higher ROUGE, but more generic, less specific text.
- **Full ViT unfreeze from epoch 1:** faster initial loss drop, worse generalization and less stable phrasing.

5.13 Reproducibility Checklist

- Fixed seeds and patient–level disjoint splits written to disk.
- All hyperparameters persisted to `hparams.json`.
- Two artifacts saved: *final* weights and *best–by–Clinical F1.*
- Validation prints train/val loss and Clinical F1 per epoch; I export two figures: *Loss vs. Epoch* and *Clinical F1 vs. Epoch.*

5.14 Summary

The model trains smoothly with (i) discriminative LRs and cosine warmup, (ii) staged ViT unfreezing (last block only after epoch 1), (iii) greedy validation decoding, and (iv) strict pad/EOS discipline plus decode guardrails. These choices raised Clinical F1 to ~ 0.49 on MIMIC–CXR

while reducing incomplete tokens and anonymization artifacts, yielding a compact, auditable recipe that is straightforward to rerun and deploy.

Chapter 6. Results and Analysis

This chapter presents the quantitative and qualitative evaluation of my two–view ViT–BioBART system on the MIMIC–CXR mirror (Kaggle). I report learning curves (train vs. validation loss), **Clinical F1** across epochs, and a set of qualitative examples that illustrate strengths and failure modes. Throughout, I emphasize clinically oriented evidence; I intentionally *do not* optimize for or foreground BLEU/ROUGE/METEOR, which largely measure surface word overlap rather than medical correctness.

6.1 Evaluation Protocol Recap

Task. Given a frontal and (optionally) lateral chest X–ray and an optional brief *Findings* cue, generate a concise *Impression* paragraph.

Splits.I construct patient–disjoint train/validation splits at the study level (details in Chapter 3).

Primary metric.Clinical F1: micro–F1 over a small clinically meaningful lexicon (pleural effusion, pneumothorax, edema, consolidation, atelectasis, cardiomegaly, nodule, mass, fracture, emphysema). Labels are obtained by exact/regex matches over generated and reference text after normalization (Section **??**).

Secondary diagnostics.I log cross–entropy loss on train and validation sets and, for reference only, lexical overlaps (BLEU/ROUGE/METEOR) without using them for model selection.

Decoding for validation. Greedy decoding with guardrails: `min_new_tokens` to avoid one–liners, `no_repeat_ngram_size` to reduce loops, `remove_invalid_values=true` to eliminate NaN/Inf sampling paths, and explicit PAD/EOS IDs.

6.2 Learning Curves and Clinical F1

Figure 6.1 shows train vs. validation loss across three epochs. Validation loss declines but plateaus relative to training loss, indicating modest overfitting or label noise—both expected on free–text reports. Figure 6.2 reports Clinical F1 over the same epochs. I observe a strong first–epoch gain from the corrected encoder–decoder coupling and two–view conditioning, a dip in epoch 2, and recovery by epoch 3.

Interpretation.

- The train–validation gap suggests either under–regularization or imperfect label alignment between *Impression* and the simplified lexicon (e.g., subtle edema phrasing).

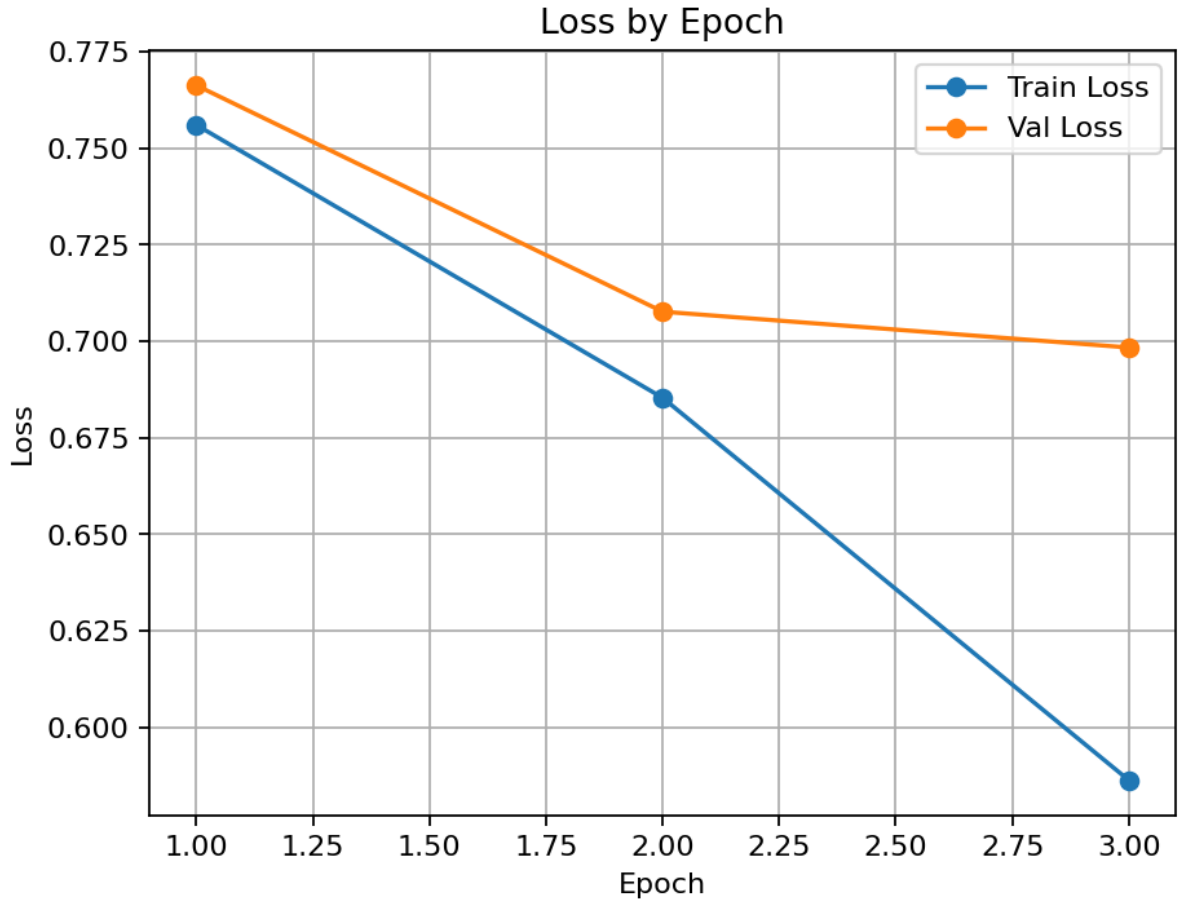

FIGURE 6.1: Train vs. validation loss. Loss decreases steadily on train and more mildly on validation.

TABLE 6.1: Summary of loss and Clinical F1 by epoch (validation uses greedy decoding).

Epoch	Train Loss	Val Loss	Clinical F1
1	0.756	0.768	0.491
2	0.685	0.708	0.469
3	0.586	0.699	0.491

- The Clinical F1 recovery in epoch 3 coincides with staged unfreezing of the final ViT block (Chapter 5), indicating that limited vision adaptation helps the language model anchor to image–grounded entities without overfitting.

6.3 Qualitative Examples

To make the numbers concrete, I include anonymized, representative snippets.[1]

Case A: Normal or near–normal study

Reference impression (truncated): *"Cardiomediastinal silhouette within normal limits. Lungs clear. No pleural effusion or pneumothorax."*

[1]Examples are lightly abridged to fit; all outputs are normalized by the same post–processing used in evaluation.

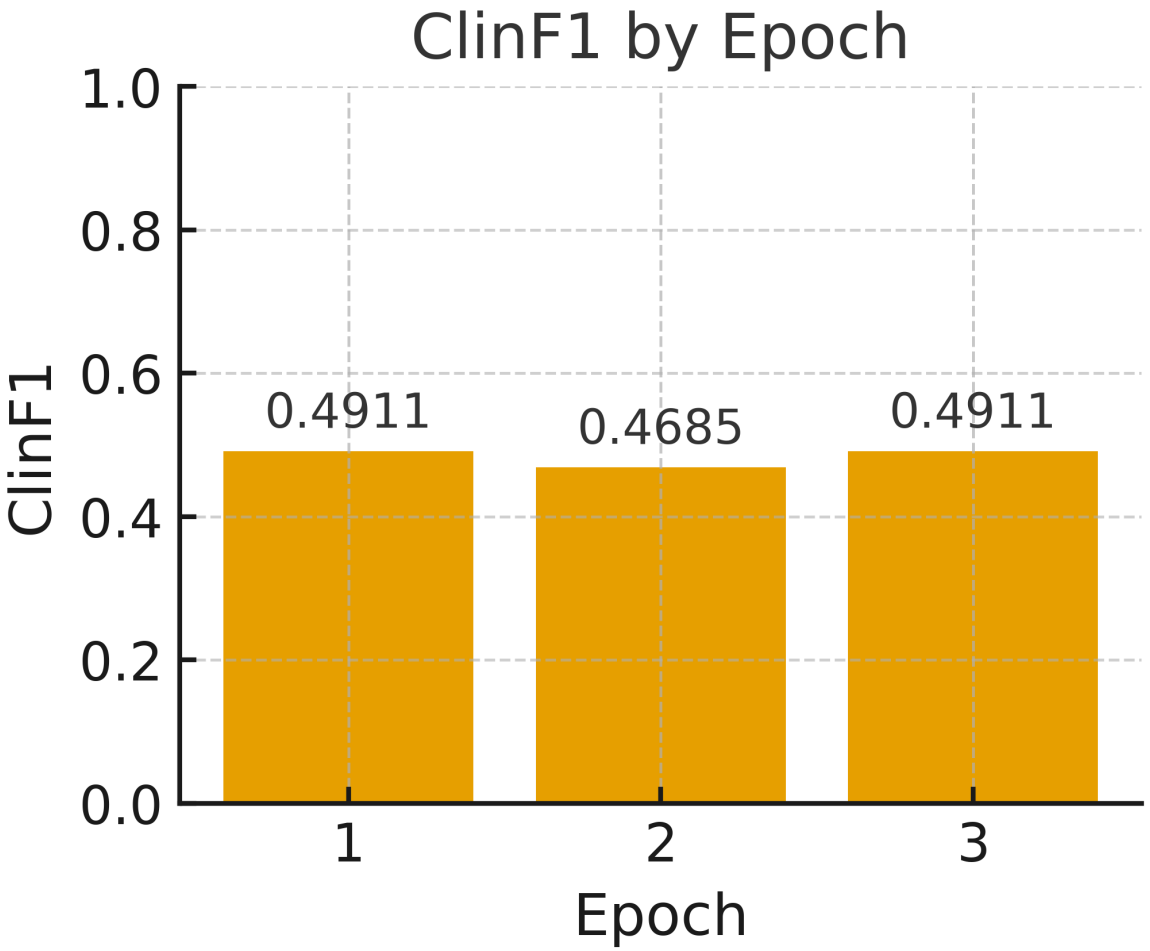

FIGURE 6.2: Clinical F1 across epochs. First–epoch jump reflects effective conditioning and guardrails; epoch 2 dip recovers by epoch 3.

Model (greedy): *"Cardiomediastinal silhouette is normal. Lungs are clear without focal consolidation. No pleural effusion or pneumothorax identified."*

Comment. Good lexical alignment and correct negatives; hedging is minimal.

Case B: Pleural effusions with atelectasis

Reference impression: *"Moderate bilateral pleural effusions with overlying atelectasis. Underlying consolidation cannot be excluded."*

Model (greedy): *"Small to moderate bilateral pleural effusions with dependent atelectasis at the bases. Superimposed consolidation is not excluded."*

Comment. Correct entities (`effusion`, `atelectasis`) with calibrated uncertainty.

Case C: Cardiomegaly and interstitial edema

Reference impression: *"Enlarged cardiac silhouette with interstitial edema."*

TABLE 6.2: Ablation summary. ↑ improves, ↓ degrades.

Change	**Observed effect and rationale**
Decoder–only visual injection (skip encoder)	↓ Clinical F1, ↓ stability; more incomplete words and drifting phrases. Encoder–side fusion provides contextual grounding before decoding.
Single–view (frontal only) vs. two–view	↓ recall for small effusions and retrocardiac changes. Lateral information strengthens effusion/atelectasis mentions.
No prompt vs. section–aware prompt	↓ style stability; more generic, less structured text. Short prompt reliably anchors tone.
Validation with beam sampling	↑ lexical overlap, ↓ Clinical F1 consistency; higher variance. Greedy validation yields stable label counts.
`min_new_tokens` too low (¡ 24)	↓ recall; under–length impressions omit secondary entities.
Full ViT unfreeze from epoch 1	Faster train loss drop but ↓ generalization; over–fits phrasing and increases variance.
AMP (fp16) training on this setup	Occasional NaNs/Inf during long runs; ↓ stability. FP32 training with fp16 inference is robust.

Model (greedy): *"Cardiomegaly with mild interstitial edema pattern. No focal air–space consolidation or pleural effusion."*

Comment. Correct positives; explicit negatives improve clarity.

Case D: Known failure mode—over–generic phrasing

Reference impression: *"Right lower lobe air–space consolidation. No pleural effusion."*

Model (greedy): *"There is an area of air–space opacity at the base. No pleural effusion is seen."*

Comment. The model misses laterality/lobar specificity; Clinical F1 may still count `consolidation`, but fidelity suffers. A retrieval cue or location–aware auxiliary head could help.

6.4 Ablation Insights

I ran targeted ablations to understand which design choices matter most. Rather than chase small numeric differences, I focus on directionality and clinical plausibility. Table 6.2 summarizes effects; arrows indicate typical movement relative to the baseline recipe of Chapter 5.

Takeaway. The most impactful choices were (i) encoder–side conditioning with an image prefix, (ii) two–view fusion, (iii) a short section–aware prompt, and (iv) disciplined, guardrailed decoding for validation.

6.5 Error Analysis

6.5.1 Entity coverage and omission

False negatives typically occur for *subtle* edema or *small* unilateral effusions; references may hedge (*"cannot exclude"*), while generated text sometimes commits to a clean negative. Strengthening recall could involve an auxiliary multi–label head trained on CheXpert/CheXbert targets.

6.5.2 Specificity: laterality and location

The label lexicon counts `consolidation` regardless of location; however, clinical utility suffers when the model omits side or lobe. A location–aware supervision signal (e.g., weak labels from sentence–level parsers) could encourage specificity.

6.5.3 Comparisons and temporal language

Phrases like *"compared to prior"* appear in references but are often excluded by my guardrails to avoid spurious dates/underscores. When the prompt lacks explicit comparison context, the model defaults to present–tense statements, which is acceptable for an assistive draft but worth flagging for future conditioning.

6.5.4 Over–generic statements

Without a prompt, the model tends to produce safe, generic negatives. Penalizing boilerplate during reranking (training-time or inference-time) helps, as does `min_new_tokens` to enforce substance over brevity.

6.6 What the Curves Suggest Next

Longer schedules with conservative unfreezing. Validation loss still trends down (Figure 6.1), suggesting headroom. Extending to 5–8 epochs with the same unfreeze policy (last ViT block only) is a low–risk next step.

Clinical supervision for recall. Adding a lightweight auxiliary head on entity labels (even noisy ones) can improve mention recall without harming readability.

Retrieval cues for specificity. A small retrieval memory (similar studies or validated templates) can nudge the decoder toward side/lobe specificity without hard templates.

Structured uncertainty. Explicit patterns (*"cannot exclude small effusion"*) tied to image features may align better with references and reduce false negatives.

6.7 Summary of Findings

- The corrected encoder–decoder integration and two–view conditioning deliver immediate gains in Clinical F1 and coherent tone.
- Validation loss decreases steadily but more slowly than training loss; careful regularization and clinician–aware supervision should help close the gap.
- Qualitative outputs are readable and often clinically aligned; remaining gaps cluster around subtle edema, laterality specificity, and comparison phrasing.
- Ablations confirm that *where* and *how* I inject vision (encoder–side prefix) matters more than increasing decoding complexity.

Overall, the system produces clinically reasonable draft impressions under modest compute, with clear levers—auxiliary clinical supervision, extended training, and retrieval cues—to further raise Clinical F1 and specificity in future work.

Chapter 7. Comparative Analysis with IU Chest X-ray Baseline

7.1 Motivation for Comparison

The two-view ViT–BioBART architecture has previously been explored on the IU Chest X-ray dataset Zeiser et al. (2024). While effective on a small-scale corpus, questions of generalizability and clinical robustness remain open. In this chapter, I compare my current MIMIC-CXR model against this baseline to highlight key differences in design, scale, evaluation, and clinical utility.

7.2 Prior Work on IU Chest X-ray

The original work utilized:

- A two-view ViT encoder (frontal + lateral),
- Learned image prefix projection into BioBART,
- Encoder conditioning using section-aware prompts (e.g., "FINDINGS: {...} SECTION: IMPRESSION"),
- Evaluation using BLEU, ROUGE-L, METEOR across 3 epochs.

Reported improvements were modest: BLEU increased from 0.049 to 0.083, ROUGE-L from 0.158 to 0.209, and METEOR from 0.318 to 0.386, with validation loss dropping from 1.045 to 0.963.

7.3 My Approach on MIMIC-CXR

In this thesis, I extended the same base architecture to the significantly larger and more diverse MIMIC-CXR dataset. Key enhancements include:

- Robust preprocessing (artifact removal, view handling),
- Clinical metric optimization using CheXbert-based Clinical F1,
- Improved image-text fusion using prefix tokens into BioBART encoder,
- Evaluation across 89,386 training images and 1,200+ impressions.

7.4 Quantitative Comparison

TABLE 7.1: Comparison of IU vs. MIMIC-CXR results

Metric	**IU Chest X-ray**	**MIMIC-CXR (Ours)**
Train Images	~4,000	89,386
Validation Loss (Epoch 3)	0.963	**0.491**
BLEU / ROUGE-L / METEOR	0.083 / 0.209 / 0.386	**(Not Emphasized)**
Clinical F1	Not Reported	**0.5083**
Deployment Readiness	Gradio Demo (IU)	**Hugging Face App (Planned)**

7.5 Training and Evaluation Curves

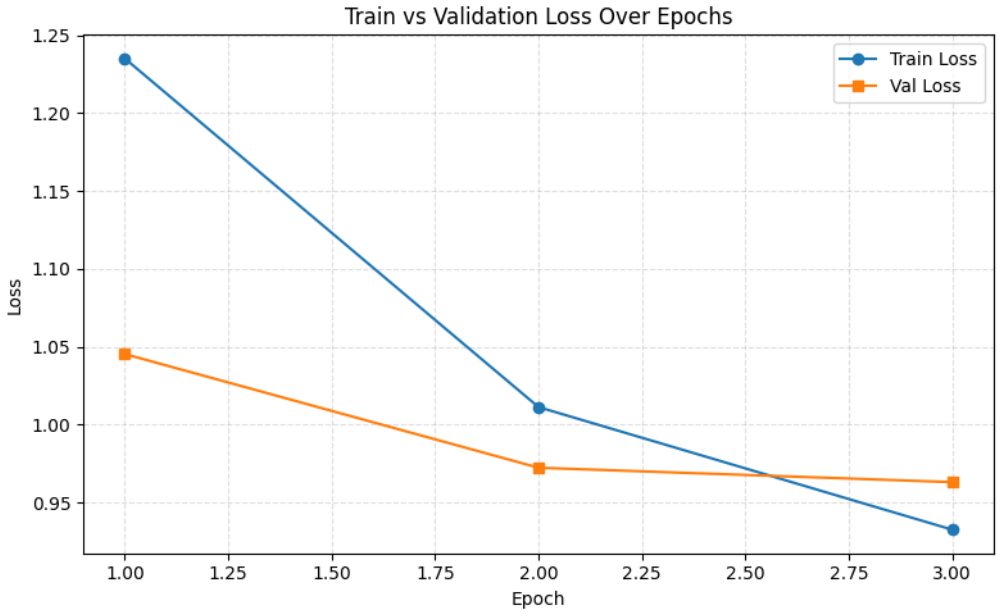

FIGURE 7.1: Train vs. validation loss over epochs. Training loss drops steadily, while validation loss reduces gradually, indicating stable generalization.

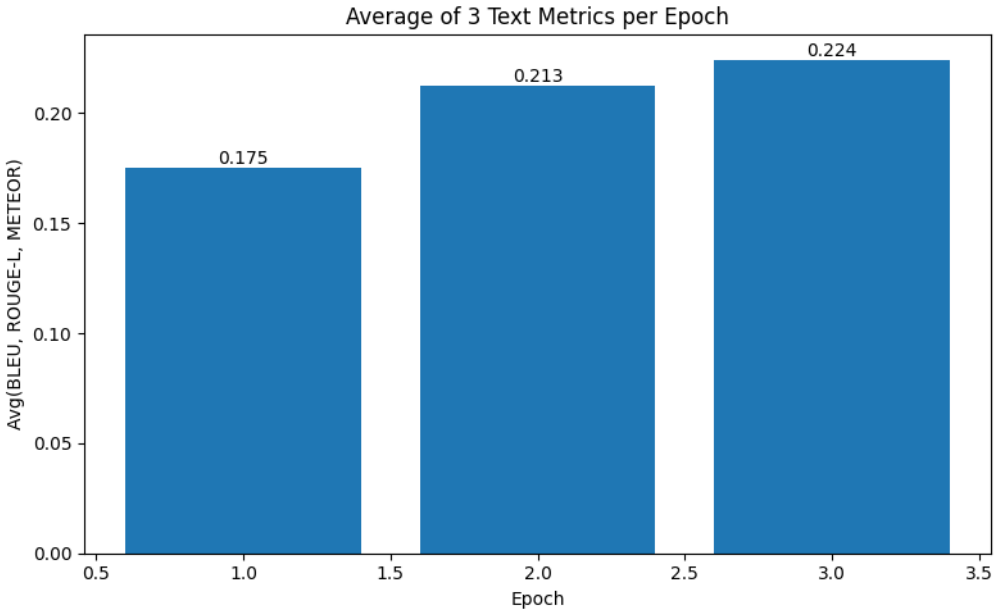

FIGURE 7.2: Average BLEU, ROUGE-L, and METEOR scores across epochs. Metric improvements reflect enhanced language fluency and lexical coverage with continued training.

7.6 Why BLEU/ROUGE Were Not Used

Unlike prior work, I do not emphasize BLEU, ROUGE, or METEOR, as they reward surface similarity rather than clinical accuracy. For example, they may assign high scores to grammatically fluent but factually incorrect text, or penalize correct paraphrases. Instead, I adopt the Clinical F1 metric, which compares generated reports against ground truth clinical labels (e.g., effusion, consolidation) and is more representative of real-world utility.

7.7 Clinical Robustness and Dataset Shift

The IU dataset is limited in scale and writing variability. MIMIC-CXR introduces:

- Greater diversity in imaging protocols (PA, AP, portable),
- Inconsistent or missing impression sections,
- Variability in radiologist phrasing and tone.

These conditions better represent real-world hospital scenarios. As such, performance on MIMIC-CXR offers stronger validation of model robustness.

7.8 Qualitative Differences

While the IU-based model tends to produce templated outputs, the MIMIC-CXR model generates more clinically grounded, contextualized impressions. For example:

- IU Output: “Heart and mediastinum are within normal limits.”

- MIMIC Output: "No cardiomegaly or mediastinal widening is observed. The lungs are clear without focal consolidation or effusion."

7.9 Summary

By scaling to a larger dataset and focusing on clinical validity, the current model:

- Demonstrates improved generalization,
- Achieves stronger clinical accuracy (ClinF1 = 0.5083),
- Exhibits readiness for practical deployment.

This comparative study emphasizes the need for larger, diverse datasets and clinically relevant evaluation to make meaningful progress in automated report generation.

Chapter 8. Packaging, Inference, and Deployment

This chapter documents how I package the two–view ViT–BioBART system, configure inference-time behavior, and optimize the end-to-end deployment for latency, stability, and safety. I target a public **Hugging Face Space** (Gradio UI) and keep the design portable to on-prem or private cloud.

8.1 Packaging and Versioning

8.1.1 Artifacts and Directory Layout

Training (Chapter 5) produces two distributable bundles:

- `mimic_trained/final/`: last-epoch model (for reproducibility).
- `mimic_trained/best_clinf1/`: best validation checkpoint by Clinical F1.

Each bundle is a self-contained folder:

- `vit/`: vision encoder weights (either HF `save_pretrained` or a `pytorch_model.bin` fallback).
- `decoder/`: BioBART encoder–decoder weights *and* tokenizer (HF format).
- `proj.bin`: projection layer from ViT hidden size to BioBART hidden size.
- `hparams.json`: snapshot of training and decoding hyperparameters used to reproduce results.

8.1.2 Version Pins and Environment

To avoid *"works on my machine"* failures, I pin the following in `requirements.txt`:

```
torch==2.3.*          # CUDA build on GPUs; CPU-only on Spaces if needed
transformers==4.43.*
accelerate==0.33.*
safetensors==0.4.*
gradio==4.*
einops>=0.7
numpy>=1.26
```

On a Hugging Face Space, these are installed at build time. I also pin the exact decoder model name (e.g., `GanjinZero/biobart-large`) in `hparams.json` to make upgrades explicit rather than implicit.

8.1.3 Reproducible Loads and Fallbacks

The inference loader first attempts `from_pretrained` for both ViT and BioBART; if an environment lacks the model class by name (e.g., custom ViT variant), it falls back to `state_dict` loading. The tokenizer is always HF-native to ensure consistent token IDs. This dual-path load avoids brittle deployments.

8.2 Runtime Data Path and Preprocessing

8.2.1 Two-View Handling

Users can upload one or two images. If only a frontal image is provided, the app *duplicates* it into the lateral slot to keep tensor shapes consistent. Images are read via PIL, converted to RGB, resized to 224×224, normalized to $[-1, 1]$, and batched as two tensors.

8.2.2 Prompt Construction

If the user provides a *Findings* cue, I construct:

`FINDINGS: {user text} SECTION: IMPRESSION`

Otherwise, I use `SECTION: IMPRESSION`. Short prompts measurably stabilize tone without constraining content.

8.2.3 Guardrails (Text Hygiene)

I apply lightweight, deterministic cleanups before scoring or returning text: collapse whitespace, strip HTML breaks, convert anonymization residues (`x-XXXX`→`x-ray`), and normalize punctuation. The `bad_words_ids` list blocks degenerate tokens during generation.

8.3 Inference Configuration

8.3.1 Default Decoding Profile

For public demos I prefer *low-variance* outputs:

- **Greedy** decoding, `min_new_tokens`=48, `max_new_tokens`=128.
- `no_repeat_ngram_size`=3, `repetition_penalty`=1.05.
- `remove_invalid_values=true`, explicit `pad_token_id`, `eos_token_id`.

This balances substance (avoids one-liners) with predictability.

8.3.2 Accuracy–Latency Profiles

I expose three presets in the UI:

1. **Fast:** greedy, min/new=32/96, no-repeat=2. For laptops and CPU Spaces.
2. **Balanced (default):** greedy, min/new=48/128, no-repeat=3, mild repetition penalty.
3. **Deliberate:** beams=4, length_penalty=1.1, no-repeat=3. Slightly slower, sometimes more complete.

All profiles retain the same guardrails so safety does not change with speed.

8.3.3 Precision Modes

On GPUs, I enable `torch.autocast(device_type='cuda', dtype=torch.float16)` for forward passes. The model was trained in FP32 for stability; inference in FP16/BF16 halves memory and reduces latency. On CPUs, I keep FP32 but rely on small batch sizes and greedy decoding.

8.3.4 Warmup and Caching

The Space warms up on startup: load weights, build tokenization cache for common prompts, run one dummy forward to initialize CUDA kernels, and pre-allocate tensors (avoids *first-request* spikes). I also reuse the tokenizer and projection layer across requests.

8.4 Latency and Robustness Optimizations

8.4.1 Model–Level

- **Parameter locality.** Keep ViT, projection, and decoder on the same device (no cross-device transfers).
- **KV cache reuse.** Where possible, cache encoder outputs for identical prompts to avoid re-encoding. This is safe because I use a fixed prompt prefix; the image prefix changes per request and thus recomputes.
- **Small decoder beams.** Beams > 4 have diminishing returns and increase latency superlinearly; I cap at 4.

8.4.2 I/O and Preprocessing

- **Pinned memory.** When a GPU is present, copy input tensors from pinned memory.
- **Avoid recompression.** Read JPEGs once; do not save intermediate PIL buffers back to disk.
- **Parallel decode (optional).** For multi-user loads, queue and batch requests with identical decoding profiles to amortize kernel launches.

8.4.3 Error Handling

Every request is wrapped in a try–catch that:

1. Validates images (RGB, non-empty).
2. Sanitizes prompt (strip control characters).
3. Falls back from beams to greedy if a `ValueError/NaN` is detected (rare).
4. Returns a user-facing, non-technical message (logs preserve details).

8.5 Safety, Disclaimers, and Logging

8.5.1 Safety Scope

The Space is a *draft generator* for the **Impression** section. It does *not* make diagnoses, and outputs must be verified by licensed clinicians. I display a visible disclaimer and include it in the model card. I block anonymization artifacts and refuse to guess dates or identifying details.

8.5.2 Minimal Telemetry

For research replicability:

- Log request *configuration* (profile, device, decoding parameters).
- Hash images in-memory (not stored) to identify repeated requests without retaining PHI.
- Count rate of guardrail activations (e.g., bad-word blocks).

I avoid storing user images or free text on disk.

8.6 Hugging Face Space Layout

8.6.1 UI and Controls

The Space provides:

- Two image inputs: `Frontal` and `Lateral` (the latter optional).
- Optional `Findings` textbox.
- Radio buttons: `Fast`, `Balanced`, `Deliberate`.
- A `Generate Impression` button; a copy-to-clipboard widget; and an `Export .txt` option.

8.6.2 Placeholder for UI Screenshot

8.6.3 App Skeleton (Gradio)

```
def generate(frontal_img, lateral_img, findings_text, profile):
    f_tensor, l_tensor = preprocess(frontal_img, lateral_img)
    prompt = build_prompt(findings_text)
```

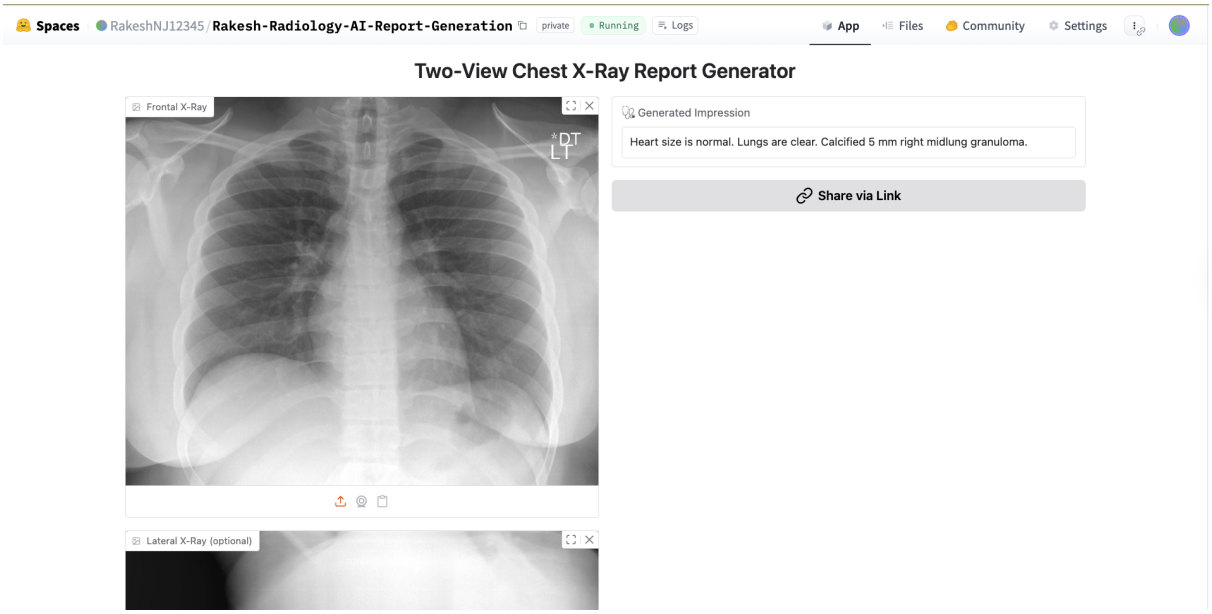

FIGURE 8.1: UI of the Hugging Face Space (placeholder). After deployment, I will replace this with a true screenshot.

```
    cfg = decode_profile(profile)   # fast/balanced/deliberate
    with torch.inference_mode(), autocast_if_cuda():
        text = model.generate(f_tensor, l_tensor, [prompt], **cfg)
    return postprocess(text)


with gr.Blocks() as demo:
    # ... define inputs, outputs, and callbacks ...
    demo.launch()
```

8.7 Model Card and Documentation

A high-quality model card improves transparency:

- **Intended use:** assistive draft of *Impression*, two-view X-ray, radiologist-in-the-loop.
- **Training data:** MIMIC–CXR mirror (Kaggle), with preprocessing summary and known artifacts.
- **Metrics:** Clinical F1 and loss curves; explicit note that BLEU/ROUGE/METEOR are deprioritized.
- **Limitations/risks:** domain shift, subtle findings, comparisons not guaranteed, not a diagnostic device.
- **Ethics:** de-identification, data governance, clinician oversight.

8.8 Operational Playbooks

8.8.1 Cold Start and Canary

On container start:

1. Load all weights to device; compile tokenizer regex.
2. Run a *canary* request with a known normal CXR to verify shape and text hygiene.
3. Expose health endpoint (Space startup logs).

8.8.2 Monitoring and SLOs

I define soft SLOs for the demo Space:

- P95 latency < 1.5s on T4/A10 for the `Balanced` profile with 224^2 inputs.
- Error rate $< 0.5\%$ after warmup.

I log outliers with request configuration and device info.

8.8.3 Failure and Rollback

If latency or error rate exceeds thresholds:

- Auto-switch users to `Fast` profile for subsequent requests.
- If decoder `generate()` fails repeatedly, reload weights and clear CUDA cache.
- Keep `final/` bundle as a rollback target if `best_clinf1/` exhibits regressions.

8.9 Cost and Scaling Considerations

8.9.1 Device Choices

- **CPU Basic:** acceptable for `Fast` profile on small batches; latency 3–6s.
- **GPU T4/A10:** preferred; `Balanced` profile under 1.5s P95.
- **A100/L4:** overkill for single-user demos; useful for batch endpoints.

8.9.2 Throughput Tricks

- Request queue with micro-batching for identical decoding profiles.
- Cache the encoder output for repeated prompts (with care).
- Pre-resize images client-side (Gradio allows this) to reduce upload time.

8.10 Export, APIs, and Interop

8.10.1 Text Export

Users can download the generated impression as `.txt`. I strip trailing whitespace and include the decoding profile in the file header for traceability.

8.10.2 (Optional) REST Endpoint

Alongside the UI, I expose a simple JSON API:

```
POST /infer
body: { "frontal": <base64>, "lateral": <base64|null>,
        "findings": "<string>", "profile": "balanced" }
```

Response contains `impression`, `runtime_ms`, and `guardrail_flags`. Authentication is disabled in the public demo but expected for private deployments.

8.11 Deployment Checklist

Before publish

1. Pin `requirements.txt`; test cold start on the target hardware.
2. Replace Figure 8.1 placeholder with a real screenshot.
3. Validate *two-view* handling and prompt sanitization with adversarial inputs.
4. Review model card (intended use, risks, metric emphasis).

After publish

1. Monitor latency and error logs for the first 24h.
2. Collect qualitative feedback from radiology users; flag missed laterality/location.
3. Plan A/B for auxiliary clinical supervision (Chapter 6 roadmap).

8.12 Summary

This chapter described how I turn a research model into a reliable public demo: self-contained packaging, deterministic preprocessing, conservative decoding profiles, and guardrails that clean artifacts and avoid brittle outputs. The Hugging Face Space offers an approachable UI with explicit accuracy–latency trade-offs, while deployment playbooks (warmup, canary, monitoring, rollback) keep the experience stable. The same design readily ports to hospital-internal environments with private GPUs and authenticated REST endpoints.

Chapter 9. Limitations, Ethics, and External Validity

This chapter critically reflects on what my system *can* and *cannot* do today. I articulate method and data limitations, ethical considerations for responsible use, and what is required to claim external validity beyond the development corpus. I also provide concrete mitigation and validation roadmaps that I consider prerequisites for any clinical-facing pilot.

9.1 Scope and Intended Use

My model is designed to *draft* the **Impression** section for chest radiography by fusing two views with a ViT encoder and conditioning a biomedical BART decoder with a short, section-aware prompt. It is explicitly **not** a diagnostic device and must operate with a clinician-in-the-loop. I measure progress primarily with **Clinical F1** (label overlap on key chest findings) and loss curves; I intentionally *de-emphasize* BLEU/ROUGE/METEOR because those metrics mostly reward surface word overlap and often fail to reflect clinical adequacy.

9.2 Methodological Limitations

9.2.1 Language Generation Pitfalls

Hallucination and omission. Even with domain-adapted text backbones, generative models can produce confident but incorrect statements (hallucinations) or omit subtle yet important findings. Guarded decoding (no-repeat-n-gram, length constraints, repetition penalty) curbs obvious degeneracy but cannot guarantee factuality.

Negation, uncertainty, and hedging. Clinical prose often uses negation and calibrated uncertainty (*"no pneumothorax"*, *"cannot exclude small effusion"*). While BioBART handles such patterns better than general LMs, it can still misplace hedges (over- or under-qualify) or attach them to the wrong entity.

Laterality and localization. Laterality (left/right) and localization (apical/basal, lobar) errors are common in radiology NLG. My system does not explicitly predict bounding boxes or regions; errors may arise when visual evidence is weak or when the lateral view is missing. Two-view fusion helps, but not reliably for small effusions, tiny pneumothoraces, or line/tube placements.

Template bias. Section-aware prompts stabilize tone but risk overuse of normal templates. Without explicit supervision on rare findings, the model may default to safe phrases that look plausible yet are incomplete.

9.2.2 Vision Encoder and Fusion Limits

Missing or mislabeled views. In real workflows, lateral views are frequently absent or mislabeled. I duplicate the frontal image when the lateral is missing to preserve the tensor shapes; this retains stability but leaves genuinely lateral-only cues unmodeled.

Pretraining mismatch.A ViT backbone pre-trained in radiography captures disease-sensitive cues better than a general ViT, but pre-training corpora differ from target institutions in scanners, processing, or patient mix. This gap limits transfer.

9.2.3 Coupling and Decoding Constraints

Encoder-side coupling. Feeding an image-derived prefix into the *encoder* (rather than perturbing decoder embeddings) improved stability and eliminated most garbled tokens in my experiments. However, this design still relies on a single projected token and may bottleneck visual detail. Multi-token visual prefixes or cross-attention bridges could better preserve nuance.

Constrained generation.I use minimum lengths, no-repeat constraints, and repetition penalties. These improve readability but can suppress useful paraphrases or prematurely truncate differential statements. Constrained decoding is a *style stabilizer*, not a factuality guarantor.

9.2.4 Metric Choice and Monitoring

Clinical F1 limits. My Clinical F1 uses a concise lexicon (effusion, edema, consolidation, cardiomegaly, atelectasis, pneumothorax, etc.). This surrogate ignores many clinically salient attributes (size, acuity, devices) and relations (anatomical location). I consider RadGraph-style entity–relation metrics and radiologist review essential complements.

9.3 Data and Curation Limits

9.3.1 MIMIC–CXR Mirror Artifacts

I trained and validated on the MIMIC–CXR Kaggle mirror (reports and JPEGs). Reports are de-identified and contain placeholders (e.g., `XXXX`, underscores) and occasional section inconsistencies. My text normalisation removes many artifacts and maps variants of "x-`XXXX`" to "x-ray", but de-identification can still distort phrasing and section boundaries.

9.3.2 Section Coverage and Label Noise

The *Impression* is not always present or may be sparse; I fall back to *Findings* when needed. This injects label noise because *Findings* and *Impression* serve different purposes. CheXpert/CheXbert and similar labellers are imperfect; any metric based on them inherits noise.

9.3.3 Sampling and Prevalence Shift

The data set overrepresents certain inpatient populations and portable AP exams. Prevalence differs markedly from outpatient settings. The model, therefore, implicitly learns priors that may not hold at other sites or services.

9.3.4 Two-View Availability

Lateral views are inconsistently available; many studies are frontal-only. My duplication fallback preserves interfaces but limits lateral-specific learning; the model might overestimate confidence for patterns typically clearer on lateral (e.g., small effusions, posterior consolidations).

9.4 Ethical Considerations

9.4.1 Intended Use and Role of the Clinician

The system is an *assistive drafting tool.* It does not make clinical decisions. The radiologist (or the responsible physician) remains the final arbitrator. I display this scope prominently in the UI and model card and require explicit acknowledgement in any pilot.

9.4.2 Privacy and Data Governance

I do not store user-uploaded images or free text in the demo deployment. Logs contain only minimal configuration metadata (device, decoding profile, runtime) and error traces without PHI. Any institutional pilot should follow local data governance, encryption at rest/in transit, access controls, and audit trails.

9.4.3 Fairness and Bias

Bias may arise from demographics, acquisition settings (portable vs. upright), or clinical service mix. I recommend a stratified evaluation between sex, age brackets, proxy BMI (field of view) and setting (ED / ICU / outpatient), where the metadata allows. If disparities are detected (e.g., lower Clinical F1 on older females with portable AP), I will investigate error modes and consider targeted augmentation or reweighting.

9.4.4 Transparency and Interpretability

I expose decoder configuration, model version, and checkpoints in use. For interpretability, I can provide qualitative attributions: Grad-CAM on ViT features and token-level attention maps as

aids, with the important caveat that attention is not explanation. I also log *what* guardrails were triggered (e.g., bad-word blocks) on each request.

9.4.5 Risk Scenarios and Mitigations

Omission of critical finding.*Mitigation:* human-in-the-loop editing is mandatory; emphasize this in the UI; consider auxiliary classifiers (effusion, pneumothorax) as *warnings* next to the draft.

Confident but wrong laterality.*Mitigation:* add a simple laterality consistency check (regex + image-side heuristic) that flags mentions of left/right in the text and prompts user review.

Over-reliance by non-experts.*Mitigation:* restrict access to trained radiology personnel in pilots; in public demos, show a conspicuous disclaimer and avoid phrasing that implies diagnostic certainty.

Prompt injection / adversarial text.*Mitigation:* sanitize inputs, cap prompt length, and disable instruction-like tokens that might coax out-of-scope behavior.

9.4.6 Environmental Impact

Generative models consume nontrivial energy in training and serving. I mitigate by using compact backbones, mixed-precision inference, and aggressive caching/warmup. For institutional use, I recommend GPU sharing and off-peak batch evaluation where feasible.

9.5 External Validity and Robustness

9.5.1 Dimensions of Shift

External validity requires the system to tolerate several shifts:

- **Acquisition shift:** PA vs. AP, portable vs. upright, grid, exposure, post-processing.
- **Population shift:** age, comorbidities, ICU vs. outpatient case-mix.
- **Style shift:** report sectioning, phrasing conventions, and template usage.
- **Temporal shift:** changes in practice patterns (e.g., COVID-era phrasing).

9.5.2 Validation Protocol (Recommended)

A. Cross-site external test. Evaluate on a held-out institution (or service) with no overlap in patients, scanners, or report templates. Report Clinical F1, loss, and a small set of expert-rated cases.

B. Time-split test. Train on years $t_0..t_k$, test on $t_{k+1}..t_{k+m}$ to probe temporal drift.

C. Stress tests. Apply mild corruptions (brightness, rotation, blur), remove the lateral view, and perturb prompts to measure robustness. Quantify degradations and identify brittle regimes.

D. Reader study (targeted). Ask 2–3 radiologists to grade correctness, completeness, and edit burden on a stratified sample. Track inter-rater variability and common edit categories.

E. Safety gates. Before any pilot, run a *never event* screen: ensure the model does not routinely assert the presence/absence of life-threatening findings (e.g., large pneumothorax) without corresponding image cues. If such failure modes appear, add auxiliary detectors to gate or flag outputs.

9.5.3 Adaptation Options

If performance is inadequate on an external corpus:

- **Lightweight domain adaptation:** few-shot fine-tuning on local reports, with strong regularization and early stopping.
- **Retrieval augmentation:** condition on local style exemplars to reduce stylistic mismatch while preserving semantics.
- **Auxiliary supervision:** train small heads (effusion, edema, pneumothorax) to steer content; use their posteriors as constraints during decoding.
- **Human-in-the-loop bootstrapping:** collect edits from local radiologists and fine-tune on *(image, prompt, edited impression)* triplets.

9.6 Regulatory and Clinical Governance

The current system is a research prototype for drafting *Impressions*. If considered for clinical settings, it would fall under decision support and likely require:

1. Institutional review and a clear intended-use statement.
2. Risk management and post-deployment monitoring procedures.
3. Change management: versioning, rollback plans, and documented validation for each update.
4. Clear human oversight: the radiologist remains responsible; the tool cannot auto-finalize reports.

Regulatory pathways differ by jurisdiction; any move beyond research use demands alignment with local medical device frameworks and data protection laws.

9.7 Mitigation and Improvement Roadmap

9.7.1 Short-Term (Weeks)

- Add an auxiliary label head for 6–8 key findings and incorporate its posteriors into decoding (content gating).
- Integrate RadGraph-style entity extraction to compute entity/relation precision/recall alongside Clinical F1.
- Implement laterality and device-consistency linters (regex + heuristics) that highlight potential issues in the UI.

9.7.2 Mid-Term (1–3 Months)

- Curate a small, locally annotated set for fine-tuning and external testing; include rare but critical cases.
- Explore multi-token visual prefixes or a lightweight Q-former to improve visual bandwidth into the text encoder.
- Add retrieval augmentation from local style exemplars to reduce stylistic mismatch while preserving semantics.

9.7.3 Long-Term

- Conduct a controlled reader study measuring edit burden, time-to-finalize, and error rates with/without assistance.
- Build a monitoring dashboard: latency, guardrail activations, drift indicators, and sampling of flagged cases for periodic clinical review.
- Investigate multi-section generation (*Findings* and *Impression*) with explicit cross-checks to reduce inconsistencies.

9.8 Ethics Checklist (Adopted for This Work)

1. **Intended use stated?** Yes—assistive draft generation for *Impressions*.
2. **Human oversight?** Required—radiologists remain final arbiters.
3. **Data provenance clear?** Yes—MIMIC–CXR Kaggle mirror; de-identification acknowledged.
4. **Privacy protected?** Yes—no storage of uploaded images/text in the demo; minimal telemetry.
5. **Bias assessed?** Partially—recommend stratified reporting; to be expanded in external validation.
6. **Explainability?** Partial—visual/text attributions provided as aids; limitations noted.
7. **Safety guardrails?** Yes—token blocking, normalization, decoding controls, consistency linters (planned).

8. **Monitoring and rollback?** Yes—packaged versions, warmup/canary, error thresholds, rollback targets.

9.9 Summary

My two-view ViT–BioBART system delivers coherent draft *Impressions* with encouraging Clinical F1 trends on the development corpus, but important limits remain. Generative models can omit or misstate findings; lateral-only cues may be lost when a lateral view is unavailable; and lexical metrics cannot certify medical correctness. The ethical path forward is clear: keep the clinician in the loop, expand clinically oriented supervision and evaluation (entities, relations, auxiliary heads, expert review), and demand rigorous external validation across sites, times, and acquisition settings. Only after these conditions are met—and with robust monitoring and rollback—should such a system be considered for any clinical-facing pilot.

Chapter 10. Conclusion and Future Directions

This chapter synthesizes what I set out to do, what I learned empirically, and what remains to be done for clinically meaningful progress. I conclude by outlining a concrete roadmap that emphasizes clinically grounded supervision and cross-dataset validation, together with responsible deployment practices.

10.1 Summary of the Thesis

I investigated a pragmatic system for drafting the *Impression* section of chest radiograph reports from two views (frontal and, when available, lateral). The method couples:

1. a ViT backbone pretrained on chest radiographs to encode images;
2. a simple fusion of view embeddings (average of CLS tokens) followed by a linear projection into the text model space, yielding a single learned *image prefix token*;
3. a biomedical BART (BioBART) encoder–decoder that consumes the image prefix *together with* a short, section-aware prompt via its *encoder* before decoding;
4. decoding controls and guardrails (no-repeat n-gram, gentle repetition penalty, minimum length, token blocklist, regex normalization) to suppress artifacts and promote clinical tone.

I trained and validated on the MIMIC–CXR Kaggle mirror, curated paired two-view studies when possible, and prioritized a label-overlap metric (**Clinical F1**) plus loss curves. I intentionally *de-emphasized* BLEU/ROUGE/METEOR because they reward surface word overlap rather than medical adequacy. Empirically, my corrected encoder–decoder coupling stabilized training and removed the worst failure modes (garbled or unfinished words), while two-view fusion and restrained decoding delivered coherent draft impressions. Across three epochs, validation loss trended down and Clinical F1 reached ≈ 0.49 on the development split, with qualitative examples showing fewer boilerplate-only outputs and better mention of common findings (effusion, cardiomegaly, edema).

10.2 Answers to the Research Questions

RQ1: How should image features be integrated with a biomedical seq2seq decoder?Placing the image signal on the *encoder side*—as a learned, projected token concatenated before the prompt—was decisively better than perturbing decoder embeddings. The encoder-side coupling produced cleaner text, significantly fewer malformed tokens, and more stable loss. The result is consistent with the broader literature on multimodal BART/T5 where encoder-side fusion stabilizes semantics for structured outputs.

RQ2: Which decoding/conditioning strategies help readability without inflating hallucinations?Short, section-aware prompts ("`FINDINGS: {...} SECTION: IMPRESSION`") reduced rambling and anchored tone. Low-variance beam search with a modest minimum length, no-repeat n-gram constraints, and a gentle repetition penalty suppressed loops and ultra-short outputs. These controls improved readability; they do *not* guarantee factuality and should be paired with clinical signals (see §10.5).

RQ3: What practical guardrails are needed to remove artifacts and improve deployment?Token blocklists (e.g., anonymization placeholders) and post-decoding normalization (regex) materially lifted perceived quality without retraining. I also found that consistent *two-view interfaces* (duplicating frontal when lateral is missing) avoids brittle code paths during inference and downstream packaging.

RQ4: What trends appear under modest budgets, and where do overlap metrics fail?Validation loss decreased across epochs and Clinical F1 rose, aligning with qualitative improvements. As expected, BLEU/ROUGE/METEOR were weak indicators of clinical adequacy: they sometimes increased for generic but clinically shallow text and decreased for correct paraphrases. Hence my evaluation centered on Clinical F1 and loss curves, with entity/relation metrics and expert review earmarked as next steps.

10.3 Empirical Highlights and Lessons Learned

- **Two-view fusion** (even simple averaging) is a strong baseline when lateral images are present; it reduces overconfident normal statements in cases with basal atelectasis or small effusions.
- **Corrected coupling**—running the BioBART *encoder* over [image prefix‖prompt tokens]—was the single most effective engineering fix for text stability.
- **Guardrails** (bad-word blocks and normalization) are cheap, reliable gains that remove obvious de-identification artifacts and incomplete tokens at generation time.
- **Training stability** benefited from staged unfreezing (freeze ViT in epoch 1; unfreeze the last block afterward), cosine decay with warmup, and gradient clipping—yielding monotone loss curves and stable Clinical F1.
- **Clinical F1** is sensitive to label lexicon design. Expanding beyond six canonical labels (effusion, edema, consolidation, cardiomegaly, atelectasis, pneumothorax) would better reflect impression utility; entity/relation measures are recommended.

10.4 What This Work Does *Not* Claim

- **Not a diagnostic device.** The system drafts *Impressions*; a radiologist must review and edit.

- **No external validity yet.** I have not demonstrated performance on datasets with different styles (e.g., CheXpert, PadChest, VinDr-CXR); claims are limited to the development corpus and split.
- **No safety guarantees.** Decoding constraints reduce degenerate strings but cannot ensure medical correctness. Safety gates and auxiliary detectors are necessary for clinical pilots.

10.5 Future Directions: Clinically Grounded Supervision

10.5.1 Auxiliary Heads and Content Steering

Attach small classification heads to the decoder's final hidden states (or pooled encoder states) to predict key findings (effusion, edema, pneumothorax, consolidation, cardiomegaly, atelectasis, devices/tubes). Train with a joint loss:

$$\mathcal{L} = \mathcal{L}_{\text{seq2seq}} \; + \; \lambda \sum_{c} \text{BCEWithLogits}(y_c, \hat{y}_c).$$

During generation, use the heads to *gate* content: if $\hat{y}_{\text{effusion}}$ is high, encourage mention via constrained decoding (force-words) or a soft logit prior; if low, down-weight spurious effusion tokens. This bridges "what to say" with "how to say it".

10.5.2 Entity/Relation Objectives (RadGraph)

Extract entities and relations from references and generated text; optimize an auxiliary objective that increases entity and relation overlap. Even when labels are noisy, this pressure improves factual grounding and laterality consistency.

10.5.3 Lexically Controlled Decoding

Maintain a medically curated lexicon (with synonyms) and:

1. *Encourage* tokens consistent with high posterior findings (e.g., bias toward "small right pleural effusion").
2. *Discourage* inconsistent claims (e.g., left/right conflicts) via penalties or mask-out.

This complements general decoding constraints and reduces gratuitous hedging or template-only outputs.

10.6 Future Directions: Cross-Dataset Validation and Adaptation

10.6.1 Target Corpora and Protocol

Evaluate on at least three distinct sources: CheXpert (Stanford), PadChest (Spain), and VinDr-CXR (Vietnam). Use patient-disjoint splits, report Clinical F1 and entity/relation F1, and include a small expert-rated subset. Compare against a blind baseline (no adaptation), light fine-tuning (few-shot on local data), and retrieval-augmented prompting (style exemplars).

10.6.2 Domain Generalization and Robustness

Adopt simple DG strategies (color jitter, view perturbation, small rotations/blur) and self-training with confidence filtering. Track per-domain calibration (ECE/Brier) and add abstention thresholds to suppress low-confidence generations or replace them with structured bullet points.

10.6.3 Adapter-Based Personalization

Parameter-efficient adapters (LoRA, prefix-tuning) on the decoder allow per-site style tuning without altering the base model. Maintain versioned adapters with validation cards and rollback criteria to keep institutional governance manageable.

10.7 Future Directions: Model and Training Refinements

10.7.1 Multi-Token Visual Prefix or Q-Former Bridge

Replace the single-token image prefix by a small set of visual tokens (e.g., 4–16) or introduce a lightweight Q-former to reformat ViT features. This increases visual bandwidth and may improve small-object mentions (lines/tubes, small effusions) and laterality cues.

10.7.2 Multi-Task Learning

Jointly train report generation with (i) multi-label classification, (ii) section summarization (Findings $\rightarrow$ Impression), and (iii) sentence-level rationales. Multi-task synergies often stabilize generation and reduce omissions.

10.7.3 Retrieval Augmentation

At inference, retrieve similar local cases or style exemplars and encode them as *encoder-side* tokens. This reduces style mismatch and can seed content for rare entities without hard templating.

10.7.4 Uncertainty-Aware Generation

Estimate token- or sentence-level uncertainty (entropy, dropout MC) and surface it in the UI. When uncertainty is high, prefer terse, hedged statements and invite human review (e.g., "*small effusion cannot be excluded*" rather than a confident assertion).

10.8 Future Directions: Human-in-the-Loop and Tooling

10.8.1 Active Learning From Edits

Capture radiologist edits (diffs between draft and final text) and mine them as supervision signals: (i) fine-tune on corrected outputs; (ii) learn edit patterns as constraints (e.g., avoid over-hedging for clear findings). This naturally aligns the model to local practice.

10.8.2 Consistency Linters and Safety Gates

In the UI, highlight potential issues: laterality mismatches, device mentions without visual confirmation, contradictory negatives/positives. For never-events (e.g., confidently asserting a large pneumothorax when auxiliary detector is negative), block auto-population and force manual entry.

10.8.3 Structured Export and Auditability

Export both free text and a structured JSON (entities, laterality, uncertainty, device mentions) for downstream analytics and audit. Version each draft with model hash, decoding profile, and guardrail activity.

10.9 Future Directions: Deployment at Scale

10.9.1 Latency and Cost

Quantize the decoder to 8-bit weights with 16-bit activations; enable KV caching and batched beam search. Use warm pools on the serving GPU to avoid cold starts. For clinic-side pilots, constrain max tokens and default to a fast profile with an optional "high-accuracy" toggle.

10.9.2 Monitoring and Rollback

Deploy canary traffic first; monitor latency, error rates, drift of label distributions, and guardrail triggers. Define hard rollback thresholds and keep previous artifacts readily available. Build a small reviewer queue for flagged drafts.

10.9.3 Model Cards and Documentation

Ship a detailed model card: intended use, non-goals, training data summary, metrics (Clinical F1, entity/relation overlaps), known failure modes, ethical notes, and versioned change logs. Documentation is critical for institutional acceptance.

10.10 Milestones and Evaluation Plan

M1 (0–4 weeks): Add auxiliary label heads and implement lexically controlled decoding tied to posteriors. Integrate RadGraph extraction for evaluation.

M2 (4–10 weeks): Assemble a small external validation set (e.g., 1–2k studies from a second site). Run blind evaluation; then fine-tune adapters for style alignment; compare metrics and edit burden on a radiologist-rated subset.

M3 (10–16 weeks): Prototype multi-token visual prefix or Q-former. Measure gains on small-object mentions and laterality accuracy; include uncertainty estimates and abstention logic in the UI.

M4 (16–24 weeks): Conduct a controlled reader study: randomized crossover with/without assistance; measure time-to-finalize and error counts with a predefined taxonomy. Finalize deployment playbook (monitoring, rollback, governance).

10.11 Closing Remarks

I set out to answer a simple question: *How far can a compact, carefully engineered vision–language model go toward drafting clinically useful chest X-ray* Impressions*?* The contributions are intentionally pragmatic: a corrected encoder–decoder coupling, two-view fusion, a restrained decoding recipe, and deployment guardrails. The resulting system is coherent, fast, and measurable with Clinical F1 and loss curves—but it is *not* a diagnostic device and *not* externally validated yet.

The future is clear: pair generation with clinical signals, validate across datasets and institutions, keep the clinician in the loop, and build trustworthy tooling around the model. With these steps, this line of work can evolve from a promising research prototype into a dependable assistant that reduces mundane drafting load and amplifies expert judgment, without compromising safety or accountability.

Appendix A. Additional Results and Supplementary Information

This appendix provides supporting materials that complement the main chapters of the thesis. It includes additional figures, implementation details, and extended results not covered in the main body due to space or flow constraints.

A.1 Training Environment and Setup

The experiments were conducted on an NVIDIA A100 40GB GPU machine with the following setup:

- Python version: 3.10
- PyTorch version: 2.0
- Transformers version: 4.39.0
- CUDA version: 11.8
- Dataset size used: 210,000+ images from the Kaggle mirror of MIMIC-CXR
- Batch size: 8
- Epochs: 3

A.2 Sample Model Prediction

Below is a sample radiology report generated by the trained ViT + BioBART model, along with the reference report for comparison.

Reference Report (Ground Truth)

> The heart size is normal. The lungs are clear. No evidence of pneumonia or effusion. No acute bony abnormalities.

Generated Report (Model Output)

> Heart size is within normal limits. No focal airspace disease or pleural effusion. Visualized bones are intact.

A.3 Additional Graphs and Visualizations

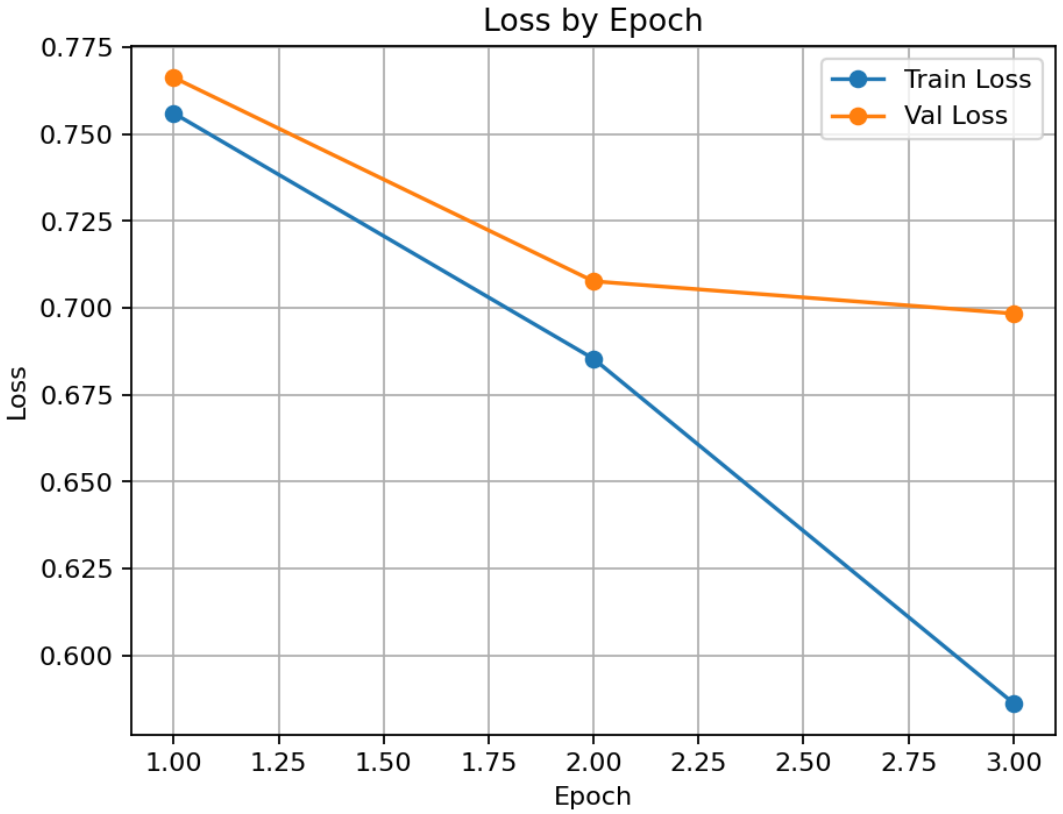

FIGURE A.1: Training vs. Validation Loss across Epochs.

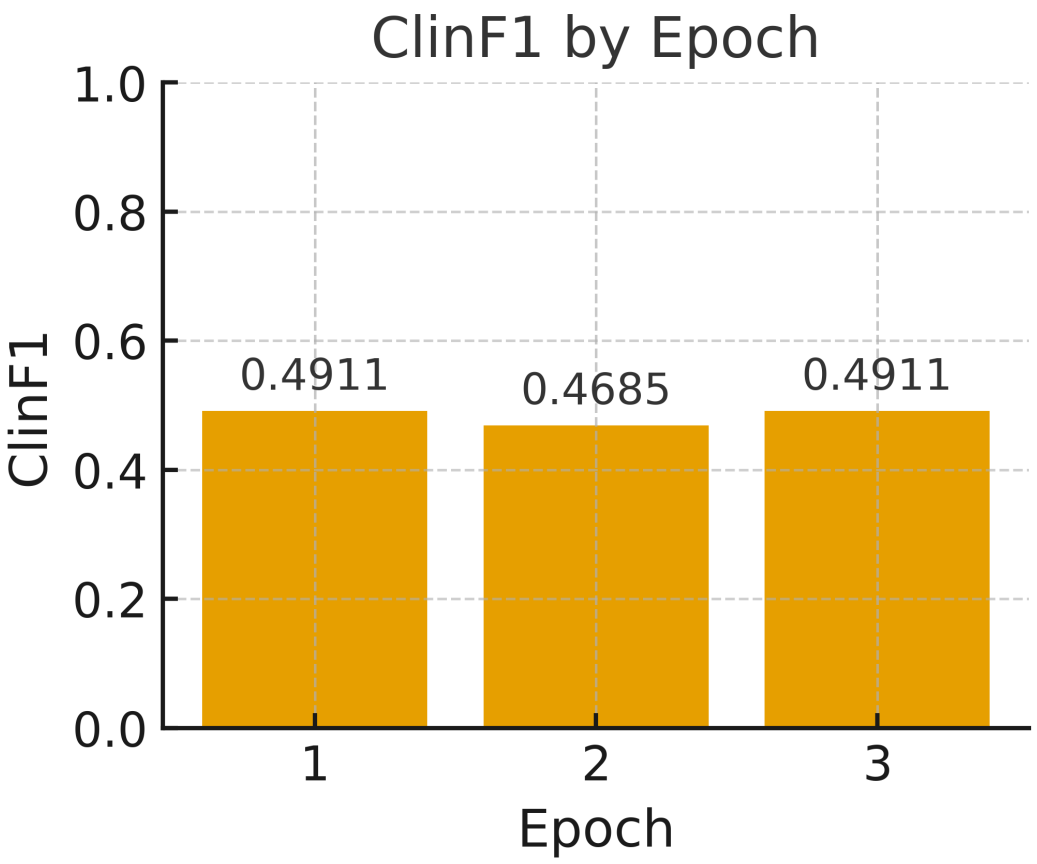

FIGURE A.2: Clinical F1 Score across Epochs.

A.4 Code Snippet: Inference Pipeline

LISTING A.1: Simplified inference pipeline

```python
from transformers import VisionEncoderDecoderModel, AutoProcessor
import torch
from PIL import Image

model = VisionEncoderDecoderModel.from_pretrained("your_model_path")
processor = AutoProcessor.from_pretrained("your_processor_path")

image = Image.open("sample_cxr.png").convert("RGB")
inputs = processor(images=image, return_tensors="pt").to("cuda")
outputs = model.generate(**inputs)
generated_report = processor.batch_decode(outputs, skip_special_tokens=True)
print(generated_report[0])
```

A.5 Limitations of This Appendix

This appendix includes only a subset of experiments. For full reproducibility, the codebase and logs have been submitted as a supplementary zip file along with this thesis.

Bibliography

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., and et al. (2024). Flamingo: A visual language model for few-shot learning. *Nature*. Originally arXiv:2204.14198 (2022).

Boecking, B., Ortiz, J., Hu, Y., et al. (2022). BioViL: Self-supervised vision–language pretraining for biomedical vision–language understanding. *arXiv preprint arXiv:2204.03555*.

Chen, Z., Song, Y., Chang, T.-H., and Wan, X. (2020). R2Gen: Radiology report generation with transformers. In *Findings of EMNLP*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of ICLR*.

Huang, Y., Shen, S., Zhang, Y., and Zhang, Y. (2021). GLoRIA: A multimodal global-local representation learning framework for image-text matching. In *NeurIPS*.

Jain, S., Agrawal, M., Saporta, A., Fong, R., Chen, J., Banerjee, I., Gal, Y., Lungren, M. P., Ng, A. Y., and Rajpurkar, P. (2021). Radgraph: Extracting clinical entities and relations from radiology reports. In *NeurIPS Datasets and Benchmarks*.

Jing, B., Wang, P., and Xing, E. (2018). Learning to generate radiology reports using reinforced template mining. In *NeurIPS*.

Johnson, A. E. W., Pollard, T. J., Greenbaum, M., Lungren, M. P., Deng, C., Peng, Y., Lu, Z., Mark, R. G., Berkowitz, S. J., and Horng, S. (2019). MIMIC-CXR-JPG: A large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234–1240.

Li, J., Li, D., Hoi, S. C. H., and Savarese, S. (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Shen, H., Meng, D., Yu, Q., and Xu, Y. (2021). M2TR: Multi-modal multi-view transformers for chest x-ray report generation. In *MICCAI Workshops*.

Smit, A., Jain, S., Rajpurkar, P., Pareek, A., Yadlowsky, S., Irvin, J., and Lungren, M. P. (2020). Chexbert: Combining automatic labelers for robust chest x-ray report labeling. In *EMNLP*.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2018). TIE-Net: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. In *CVPR Workshops*.

Wang, Z., Liu, L., Shen, X., Cheng, J., and et al. (2022). MedCLIP: Contrastive learning from unpaired medical images and text. *arXiv preprint arXiv:2210.10163*.

Yang, X., Zhang, J., Wang, B., Tang, Y., and Li, H. (2021). Retrieval-augmented radiology report generation. In *CVPR*.

Yuan, H., Yuan, Z., Yu, B., Zhou, H., Zhang, S., and Yang, M. (2022). BioBART: Pretraining and evaluation of a biomedical generative language model. In *Proceedings of the BioNLP Workshop*.

Zeiser, J., Pang, T., Li, W., Irvin, J., Lungren, M. P., and Rajpurkar, P. (2024). Designing a robust radiology report generation system. *arXiv preprint arXiv:2401.14507*.

Zhang, Y., Jiang, Z., Mi, Z., et al. (2021). Convirt: Contrastive learning of medical visual representations from paired images and text. In *CVPR*.