UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ ROMÂNĂ

LUCRARE DE LICENŢĂ

CLIP-XRGen: Învățare contrastivă pentru generarea automată de rapoarte radiologice bazată pe alinierea semantică a conceptelor medicale

Conducător ştiinţific
prof. dr. Czibula Gabriela

Absolvent
Mihăiță Tudor-Octavian

2025

BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE

DIPLOMA THESIS

CLIP-XRGen: A Contrastive Learning approach for Automated Radiology Report Generation with Medical Concept Alignment

Supervisor
Professor, PhD. Czibula Gabriela

Author
Mihăiță Tudor-Octavian

2025

ABSTRACT

Automated Radiology Report Generation is a complex multimodal task designed to streamline the radiologist workflow by generating clinically coherent textual descriptions of medical images. Conventional approaches in this domain, such as retrieval-based methods and encoder-decoder models, have demonstrated meaningful progress in generating logical reports, but often struggle to establish strong alignment between visual and textual modalities, especially in medical environments with limited data availability. This limitation leads to poor semantic understanding and omissions of critical clinial information.

To address these challenges, this thesis proposes CLIP-XRGen, a weakly supervised hybrid multimodal approach that leverages contrastive vision-language pretraining to learn a semantically aligned representation space. This contrastive paradigm results in a pretrained backbone that can serve as a strong foundation for downstream tasks such as report generation, offering a data-efficient solution aligned with the constraints of the medical domain.

The originality in this work lies in the integration of medical concept-level supervision into the contrastive alignment process, ensuring that learned representations capture relevant clinical findings. Additionally, CLIP-XRGen extends this pretrained encoder functionality into a unified framework for both vision-language understanding and conditional report generation, adopting a prompt-guided decoding strategy. This approach enables the model to generate reports focused on the pathological patterns identified in the chest X-ray images.

To demonstrate the practical utility of the model, the inference pipeline is deployed via an API and integrated into VistaScan, a web-based platform enabling remote radiology consultations. This system offers a scalable and flexible solution for modern healthcare providers, with the potential to optimize the timely annotation process and improve patient access to expert evaluation.

CLIP-XRGen shows promising results in image-text retrieval and medical concept classification tasks due to strong multimodal alignment. While conditional generation remains a more challenging objective, the model provides a solid foundation for further research into advancing accurate and concept-aware medical report generation.

The presented work is the result of my own activity. I have not given or received any unauthorized assistance in its completion.

Mihăiță Tudor-Octavian

Contents

List of publications

[Mih25] **Mihăiță Tudor-Octavian**, *CLIP-XRad: Learning Multimodal Representations of Chest X-rays through Contrastive Pretraining with Medical Concept Alignment*, Studia Universitatis Babeş-Bolyai Informatica, 2025, submitted for publication

List of Figures

List of Tables

List of Algorithms

Introduction

Automated Radiology Report Generation (ARRG) is the task of generating textual reports from medical imaging studies, aiming to extract relevant clinical information for diagnosis, treatment planning, or patient monitoring. This task has emerged as a growing research focus due to its inherent complexity and ambiguity of interpreting medical images and describing them in natural language.

A core challenge of the report generation task is designing a solution that can replicate radiologist-level reasoning by learning to distinguish patterns from visual features and detect relevant medical concepts, while also generating coherent and clinically accurate descriptions using specialized vocabulary from the medical domain. This challenge is further amplified by the limited availability of large-scale, high-quality medical image-text datasets.

Given the context, this thesis introduces **CLIP-XRGen**, a weakly supervised hybrid vision-language model that leverages contrastive learning to align chest X-ray images with their corresponding textual reports. By integrating semantic concept similarity into the contrastive objective, the model enhances its understanding of shared visual and textual representations. These aligned representations are then used to guide the generation of reports focused on key pathological findings. Experimental results demonstrate promising performance in both multimodal understanding and domain-specific text generation. To further support its practical relevance, a web-based application is developed to showcase the model's potential in supporting remote radiology consultations.

This thesis is structured as follows. Chapter 1 presents the theoretical foundations and related work in the field that supported and motivated this research. Chapter 2 outlines the proposed methodology, detailing the unified framework for contrastive pretraining and report generation. Then, Chapter 3 introduces the supporting application and its relevance in real-world clinical scenarios. Finally, Section 3.5 concludes the work by summarizing the contributions and discusses future research directions.

Usage of Generative AI

During the preparation of this thesis, the author used ChatGPT and Claude Sonnet to assist with LaTeX formatting of visual elements and support debugging during application development, without employing them to generate any factual or scientific content. After using the mentioned tools, the author has carefully reviewed and edited the generated content and takes full responsibility for the content of this work.

Chapter 1

Background

Radiology report generation has emerged as a highly relevant research area, driven by its potential to address critical challenges in modern healthcare. Although recent studies have explored a wide range of architectures and learning paradigms, the clinical nature of the task leaves considerable room for improvement, especially given the high standard of reliability and precision required in real-world medical settings.

This chapter provides a comprehensive overview of the task, emphasizing its clinical and practical relevance, while introducing key technical concepts and methodologies relevant to this field. The chapter begins with an in-depth review of radiology reporting in Section 1.1, detailing its structure, purpose, and motivation to automate the process. We continue by highlighting in Section 1.2 an auxiliary process essential for tasks involving radiology reports, specifically automated label extraction. Subsequently, an introduction to essential theoretical foundations is provided, including Vision-Language Models in Section 1.3, which serve as the architectural backbone of ARRG systems. In Section 1.4, the Contrastive Learning paradigm is presented, as the central focus of the approach proposed in this thesis, highlighting its potential in radiology report generation through an analysis of foundational contrastive pretraining models such as CLIP and BLIP. The chapter then explores domain adaptation strategies designed for medical imaging tasks, addressing the challenges posed by clinical data, and their applicability to the ARRG setting. Finally, in Section 1.5, the chapter concludes with a review of state-of-the-art (SOTA) models, offering insights into the most effective approaches leading this research domain.

1.1 Automated Radiology Report Generation

Radiology reports serve as the primary means by which radiologists communicate the findings of medical imaging studies. These reports provide clinically relevant interpretations of observed abnormalities, regardless of whether they are expected or incidental, playing a critical role in guiding diagnostic decisions. Beyond their immediate diagnostic value, these reports also contribute to maintaining a historical record of the patient's medical condition, supporting continuity of care across multiple encounters. Although no universally accepted format exists, the European Society of Radiology (ESR) [(ES11] has proposed a standardized template for structuring radiological reports:

- **Clinical referral/Indication** - A brief summary of the clinical context and the reasons for the referral.
- **Technique** - A concise description of the imaging procedure performed.
- **Findings** - A comprehensive description of clinical observations made within the study, highlighting the appearance of certain conditions or the evolution of prior problems identified at the patient.
- **Conclusion/Impression** - A summarized diagnosis of the examination based on the findings.

In recent years, the growing demand for radiological interpretation, coupled with the global shortage of radiologists, has created a pressing need for scalable, automated solutions. This challenge has given rise to the task of **Automated Radiology Report Generation** (ARRG), which aims to generate coherent and clinically accurate text descriptions directly from medical images, using AI-driven methods. These generated reports serve as preliminary assessments that can assist clinicians in validating diagnoses and improving workflow efficiency, eliminating the timely process of manual annotation, and potentially reducing patient backlog in overcrowded healthcare facilities.

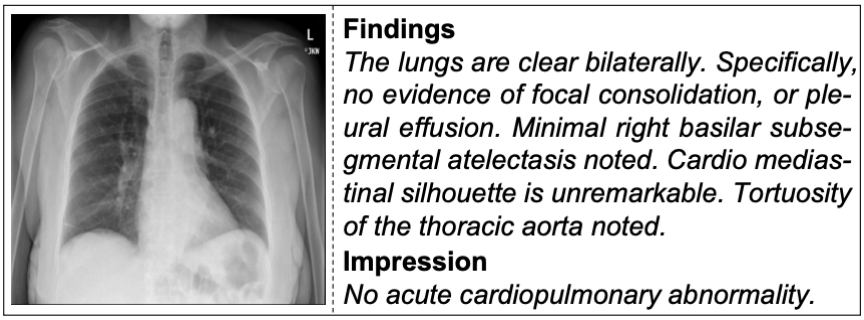


Figure 1.1: An example of a chest X-Ray with its corresponding radiology report, showing both Findings and Impression sections. The report describes normal lungs with minimal atelectasis and no acute cardiopulmonary abnormality.

A common design in ARRG systems is the **encoder-decoder** paradigm, where a visual encoder first processes the input medical image to extract semantically rich features, which are then used by a text decoder to generate the corresponding radiology report. This structure reflects the multimodal nature of the task and supports learning complex image-to-text relationships. While early approaches relied on CNN-RNN architectures, recent advances have shifted toward **Transformer-based architectures** for their stronger representation learning capabilities and ability to operate within a shared embedding space. This unified architecture simplifies the integration of visual and textual features during report generation and has led to improved performance across evaluation benchmarks.

1.2 Automated Label Extraction for Medical Imaging

While imaging studies are rich clinical information, their interpretation is typically documented through free-text radiology reports, which lack the structured format required for most supervised learning tasks. To address this challenge, automated labeling tools have been introduced to provide structured supervision in scenarios where no explicit annotated data is available. A representative example is CheXpert [IRK+19], a rule-based labeling system introduced alongside a large collection of chest X-rays, designed to convert unstructured radiology reports into multi-label annotations. CheXpert identifies the presence, absence or uncertainty of 14 common thoracic conditions, thereby enabling large-scale training of intelligent models in medical imaging tasks.

These automatically extracted labels play a crucial role in various healthcare-focused AI models: they enhance consistency in text generation, provide supervision for classification models, and enable concept-level alignment in representation learning. Such tools are particularly valuable in large-scale medical datasets, where manual annotation is often impractical or time-consuming.

Building upon CheXpert's foundation, the rule-based approach has been extended in **CheXbert** [SJR+20], a neural-enhanced labeling tool that leverages BERT-based models to improve labeling accuracy and better capture the semantic details present in clinical reports. The list of the extracted medical concepts is summarized in Figure 1.2.

CheXpert Label Set

- Atelectasis
- Cardiomegaly
- Consolidation
- Edema
- Enlarged Cardiomediastinum
- Fracture
- Lung Lesion
- Lung Opacity
- Pleural Effusion
- Pleural Other
- Pneumonia
- Pneumothorax
- Support Devices
- No Finding

Figure 1.2: The 14 diagnostic categories extracted by the CheXpert labeling tool [IRK+19], used for providing supervision in several medical-specific tasks.

1.3 Vision-Language Models

Vision-Language Models (VLMs) are multimodal architectures designed to jointly process and reason over visual and textual inputs. They learn aligned representations of images and text, enabling them to perform a wide range of cross-modal

tasks such as image captioning, visual question answering, image-text retrieval, and conditional text generation, often achieving strong performance in zero-shot scenarios. These models typically adopt an encoder-decoder or dual-encoder architecture, with each component dedicated to processing a specific modality, and are trained to produce meaningful interactions or predictions across modalities.

Although there is no specific paper that formally introduced the Vision Language Modeling paradigm, its development builds upon recent advances in both Natural Language Processing (NLP) and Computer Vision (CV), fields that increasingly converge through the shared foundation of the **Transformer** architecture. This section explores the theoretical foundations of the most prominent methods for processing each modality independently, and how these approaches are ultimately integrated to build systems capable of jointly understanding images and text. Afterward, a formal taxonomy of VLMs is presented, based on their architectural and functional characteristics.

1.3.1 Transformer-based sequence modeling

One of the most important breakthroughs that drove the development of VLMs is the Transformer architecture, proposed by Vaswani et al. [VSP+17] in the seminal paper ***Attention is All You Need***. This work marked a major paradigm shift from traditional sequence-modeling techniques such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory units (LSTMs), introducing an architecture that enables parallel computation and efficiently models long-range dependencies of input tokens.

The core innovation of the Transformer is the **Self-Attention** mechanism (Figure 1.3), which allows the model to dynamically weigh the importance of each token in relation to others within the encoded sequence. Rather than processing tokens sequentially, Self-Attention considers the entire sequence simultaneously and computes a new representation for each token by attending to all others. This allows the model to capture relationships between distant words and effectively understand the context.

Formally, this concept is defined mathematically as follows. Given an input sequence of token embeddings $X \in \mathbb{R}^{n \times d}$, each position is projected into three vectors: $Q = XW^Q, \quad K = XW^K, \quad V = XW^V$, where $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$ are learned weight matrices. These vectors are known as **Queries** (Q), **Keys** (K), and **Values** (V), respectively. The Self-Attention operation is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

This operation computes similarity scores between each query and all keys, scales them by the dimensionality d_k, applies a softmax to obtain attention weights, and uses these weights to compute a weighted sum over the values. This allows each token to encode information from the entire sequence based on learned relevance.

To allow the model to capture different types of relationships between input elements, the Transformer uses multiple attention heads in parallel, known as **Multi-Head Attention**. Each head operates in its own subspace of the input representation, therefore the operation can be defined as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \cdots, \text{head}_h)W^O$$

where each head is computed as $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, and $W^O \in \mathbb{R}^{hd_k \times d}$ projects the concatenated outputs back to the model dimension. By combining multiple attention heads, the model captures a richer set of dependencies between tokens.

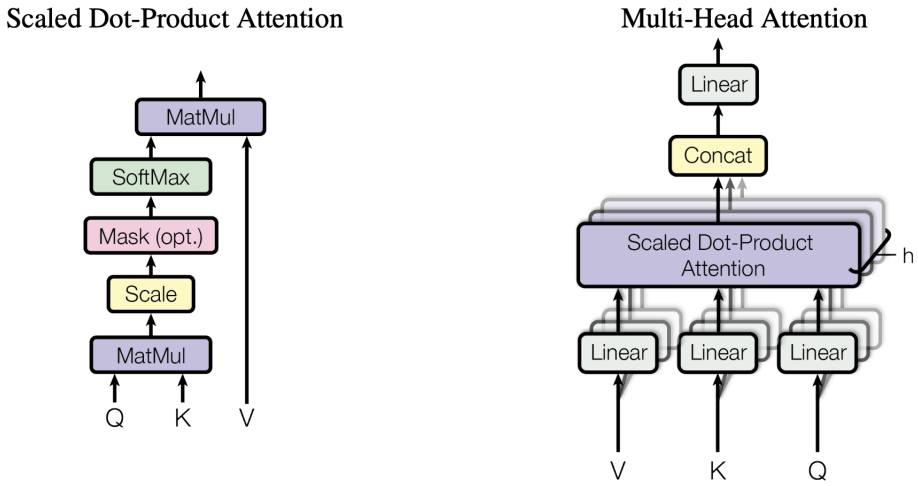


Figure 1.3: Scaled Dot-Product Attention and Multi-Head Attention mechanisms from [VSP+17]. The left diagram shows the computation of attention weights using dot products, scaling, optional masking and softmax. The right diagram illustrates how multiple attention heads with separate projections of Q, K, V.

As presented in Figure 1.4, the original Transformer architecture is composed of two main main components: a stack of **encoder** layers and a stack of **decoder** layers. Each encoder layer layer consists of a Multi-Head Self-Attention module followed by a position-wise Feed-Forward Network (FFN). To retain positional information, **Positional Encoding** is added to the input embeddings. Each sublayer is wrapped with a residual connection and is followed by layer normalization. The decoder has a similar structure, but includes a second attention block, known as **Cross-Attention**, that allows it to attend to the encoder's output representations, making the model suitable for sequence-to-sequence tasks like machine translations.

This architecture inspired several influental models with distinct configurations, adapted for various objectives. **BERT** [DCLT19] uses only the **encoder stack** for bidirectional language understanding via masked language modeling. **GPT** [RNSS18] employs only the **decoder stack** with causal self-attention for autoregressive generation. **BART** [LLG+20] combines both in a full **encoder-decoder** setup, pretrained with denoising objectives for conditional generation. These variants form the backbone of many modern Large Language Models (LLMs).

While these models have been originally designed for text, research advances have also naturally transitioned to applying similar principles for image processing tasks, by making use of the same attention mechanism in order to extract relevant information from the inputs.

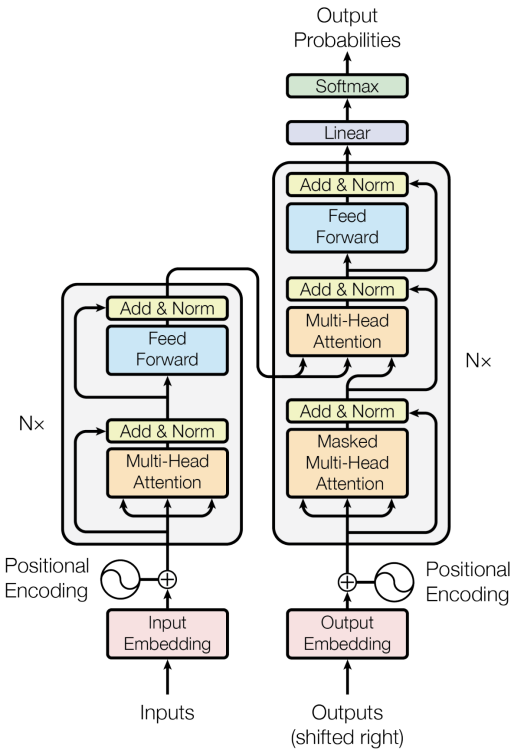


Figure 1.4: Conceptual overview of the Transformer architecture from [VSP+17]. The model consists of an encoder and a decoder, each composed of stacked layers with Multi-Head Self-Attention, Feed-Forward sublayers, and residual connections with layer normalization. Positional Encodings are added to preserve the order of input sequences.

1.3.2 Adapting Transformers for visual understanding

The **Vision Transformer** (ViT), introduced by Dosovitskiy et al. [DBK+21], adapts the standard Transformer architecture to image data by rethinking how visual inputs are represented. Instead of relying on convolutional filters and feature maps, ViT splits an image $x \in \mathbb{R}^{H \times W \times C}$ into a sequence of N non-overlapping patches of fixed-size $P \times P$. Each patch is then flattened as a vector and projected into a D-dimensional embedding space using a trainable linear projection $E \in \mathbb{R}^{(P^2 \cdot C) \times D}$, resulting in patch embeddings $x_p^i E \in \mathbb{R}^D$. The process is outlined in Figure 1.5.

To enable image-level tasks like classification, a learnable token x_{class}, also known as [CLS], is concatenated at the beginning of the patch sequence, representing the global image descriptor. To retain spatial information, learnable positional embeddings $E_{pos} \in \mathbb{R}^{(N+1) \times D}$ are added to the sequence, which can be mathematically expressed as:

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^N E] + E_{pos}$$

This resulting sequence is then passed through a stack of Transformer encoder layers, each layer consisting of a Multi-Head Self-Attention (MSA) block followed by a feed-forward MLP, both wrapped with residual connections and preceded by layer normalization:

$$z'_\ell = \text{MSA}(\text{LN}(z_{\ell-1})) + z_{\ell-1}, \quad z_\ell = \text{MLP}(\text{LN}(z'_\ell)) + z'_\ell \quad \text{for } \ell = 1 \ldots L$$

After L such layers, the final representation of the [CLS] token is extracted and passed to a classification head. Therefore, the model output is as follows:

$$y = \mathrm{LN}(x_L^0)$$

With this approach, ViT removes many of the architectural assumptions found in CNNs, such as an emphasis on local regions for extracting features and translation invariance. Although this means that it needs more data to learn spatial relationships effectively, it is at the same time an advantage from the perspective of generalization capabilities, making the model more flexible for several tasks that involve image processing. Furthermore, by processing image patches the same way that words are handled in text, ViT enables a unified architecture of VLMs for learning joint representations across both modalities.

Building on this idea, to address some of ViT's limitations in handling visual tasks that require fine-grained feature extraction on high-resolution images, the **Swin Transformer** [LLC+21] was introduced as a hierarchical and computationally efficient alternative. Unlike ViT's global attention mechanism, Swin partitions the image into local non-overlapping windows and applies Self-Attention within each window. These windows are shifted between layers to allow **cross-window interaction**. This structure mirrors the spatial hierarchy of CNNs while retaining the modeling power of Transformers. As a result, Swin Transformers have been widely adopted in tasks like object detection, semantic segmentation, and serve as a strong visual backbone for many modern multimodal systems.

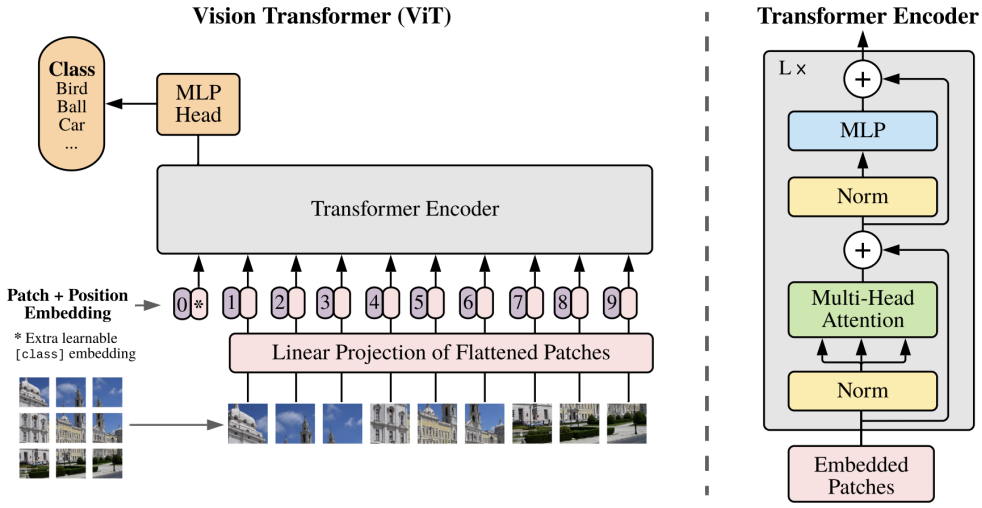


Figure 1.5: Overview of the Vision Transformer (ViT). The image is split into patches, linearly projected, and combined with a [CLS] token. The sequence is processed by the Transformer encoder, and the [CLS] token output is interpreted as the global descriptor. Source: [DBK+21]

1.3.3 Taxonomy of Vision-Language Models

While VLMs have been developed in diverse forms to address different downstream tasks, this thesis focuses on two primary categories that closely align with the proposed approach:

- **Contrastive-based VLMs**: These models learn aligned representations of different modalities through discriminative objectives. They usually leverage a dual-encoder architecture, each responsible for a specific data type, as it can be seen in Figure 1.6.
- **Generative VLMs**: These models are trained to generate a modality conditioned on the other one as input. Due to their autoregressive or nature or encoder-decoder architectures, generative VLMs tend to be more computationally intensive, but they can benefit significantly from pretrained language and vision backbones, usually designed as general-purpose and further adapted to a specific domain.

It is worth noting that these paradigms are not mutually exclusive and often overlap. Many recent VLMs adopt hybrid training strategies that integrate multiple types of VLMs to take advantage of the strengths of each approach and therefore enhance the overall model robustness and versatility. In line with this direction, this thesis builds upon combining contrastive and generative components to better align image-text representations and generate clinically relevant reports.

1.4 Contrastive Learning

Contrastive Learning is a self-supervised learning paradigm in which models are trained to learn representations by comparing samples, bringing similar pairs closer together in the embedding space while pushing dissimilar ones apart. Unlike traditional supervised learning that depends on discrete class labels, contrastive methods rely on implicit relationships derived from the data itself, making them especially effective in settings with limited annotated data.

In the context of cross-modal learning, contrastive methods are commonly used to align representations from different modalities, such as images and text, by training models to maximize similarity between semantically paired inputs (e.g., an image and its corresponding caption) while minimizing similarity between unrelated pairs (Figure 1.6). This is typically achieved using objectives like InfoNCE or Triplet Loss, which encourage the model to embed matching modalities close together in a shared latent space.

While contrastive methods are inherently self-supervised, recent work has shown that integrating supervision with **soft or structured labels** can further improve alignment in downstream task performance. This combination enhances the model's ability to capture fine-grained relationships and generalize effectively in **zero-shot** or **few-shot** scenarios.

In the following section, foundational contrastive-based Vision Language Models (VLMs) are presented, with an emphasis on their architectural innovations, training strategies, and key contributions to multimodal learning. We also highlight how these methods influenced the design choices behind the proposed approach for radiology report generation.

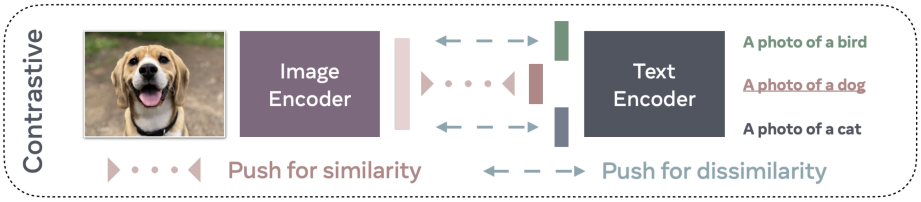


Figure 1.6: Illustration of the Contrastive Learning paradigm applied in a Vision Language Model. Image and text encoders are jointly trained to project matching image-text pairs closer in a shared embedding space while pushing apart non-matching pairs. Source: [BPA+24]

1.4.1 Contrastive Language-Image Pre-training

Developed by OpenAI as an open-source model and introduced by Radford et al. [RKH+21], **Contrastive Language-Image Pre-Training** (CLIP), set a new standard for zero-shot image classification tasks with strong performance across a wide variety of benchmark datasets. Based on the principle of Occam's razor, favoring simplicity and generality, CLIP is trained using a contrastive objective on 400 million uncurated image-text pairs collected from the web. Unlike traditional supervised models trained on fixed label sets, CLIP learns to align visual and textual modalities by jointly training an image encoder and a text encoder to embed their inputs into a shared semantic space (Figure 1.7). The model employs a contrastive learning framework based on the **InfoNCE** loss, which encourages matching image–text pairs to have high similarity, while non-matching pairs are pushed apart. Given a batch of N image–text pairs $\{(x_i, t_i)\}_{i=1}^{N}$, the image encoder $f(\cdot)$ and the text encoder $g(\cdot)$ are trained to maximize the cosine similarity between matching embeddings. The InfoNCE loss for an image–text pair (x_i, t_i) is defined as:

$$\mathcal{L}_{\text{InfoNCE}}^{(i)} = -\log \frac{\exp\left(\text{sim}(f(x_i), g(t_i))/\tau\right)}{\sum_{j=1}^{N} \exp\left(\text{sim}(f(x_i), g(t_j))/\tau\right)} \tag{1.1}$$

where $\text{sim}(u, v) = \frac{u^\top v}{\|u\|\|v\|}$ is the cosine similarity between the embeddings and τ is a temperature hyperparameter that controls the sharpness of the distribution. A symmetric version of this loss is also computed in the text-to-image direction, and both components are averaged during training. This simple yet effective training paradigm allows CLIP to effectively align visual and textual semantics without task-specific supervision, resulting in generalizable representations applicable to a wide range of downstream vision-language tasks, including image captioning, classification, and retrieval.

Due to its ability to learn rich and transferable visual representations from loosely paired image-text data, CLIP presents a compelling foundation for the proposed approach of this thesis. In the medical domain, where annotations are often sparse or inconsistent, CLIP's reliance on weak supervision makes is particularly effective. Additionally, its zero-shot capabilities enable the model to capture high-level clinical concepts, an essential aspect of radiology reporting, which aims to describe the key pathological findings within a study. In this context, a contrastive pretraining phase is employed to prepare a robust visual backbone for the downstream task.

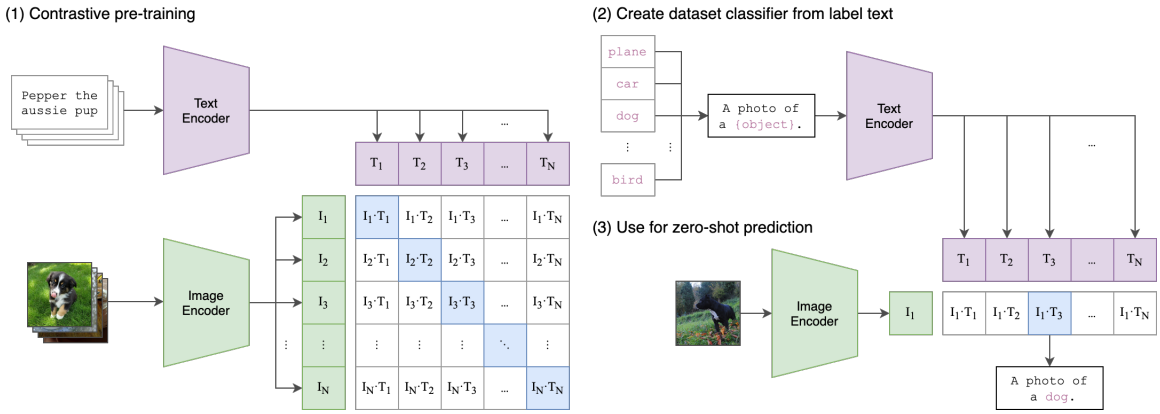


Figure 1.7: The architecture of the CLIP model from [RKH+21]. (1) During pretraining, image and text encoders are jointly optimized to match positive pairs and push mismatched pairs apart. (2) At inference, a zero-shot classifier is constructed by encoding textual prompts corresponding to class labels. (3) A test image is embedded and matched to the closes text embedding to perform zero-shot classification.

1.4.2 Unified Vision-Language Understanding and Generation

Bootstrapped Language Image Pretraining (BLIP), proposed by Li et al. [LLXH22], is a vision–language model that introduces a novel architecture of **Multimodal mixture of Encoder-Decoder** (MED), designed for effective multi-task pretraining and transfer learning. As seen in Figure 1.8, this **MED** framework supports multiple configurations, operating as a unimodal encoder, an image-grounded text encoder, or an image-grounded text decoder. By leveraging both noisy web image-text pairs and high-quality image-caption data, BLIP is jointly pretrained with three vision-language objectives:

- **Image–Text Contrastive Loss (ITC):** Encourages the alignment of global image and text representations in a shared embedding space, using a similar contrastive loss as CLIP's InfoNCE 1.1. It computes the same $sim(\cdot)$ cosine similarity between pairs in a batch and scales it by the τ temperature parameter, resulting in the loss expression $\mathcal{L}_{\text{ITM}}$.

- **Image–Text Matching Loss (ITM):** A binary cross-entropy loss used to predict whether an image–text pair is semantically aligned. This objective fine-tunes cross-modal fusion representations and helps filter false positives from noisy data. The loss is defined as:

$$\mathcal{L}_{\text{ITM}} = -\left[y \cdot \log(\sigma(h(x,t))) + (1-y) \cdot \log(1-\sigma(h(x,t)))\right]$$

  where $y \in \{0,1\}$ is the label telling if the pair is aligned (positive) or misaligned (negative), and $h(x,t)$ is the matching score predicted from the fused multimodal representation.

- **Language Modeling Loss (LM):** A standard next-token prediction objective that enables conditionally generating textual descriptions in an autoregressive manner. Given an input image x and partially generated caption tokens $t_{<i}$,

the loss is expressed as:

$$\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{n} \log P(t_i \mid t_{<i}, x)$$

While BLIP was originally developed for general-domain language-vision tasks, its architectural design and training strategy offer strong potential for specialized applications such as ARRG. Notably, the image-grounded text decoder demonstrates that contrastively pretrained models can be effectively fine-tuned for conditional text generation. However, the joint pretraining of both encoder and decoder components introduces significant training complexity, which may be suboptimal in domain-specific settings where high-quality annotated data is limited. This motivates the more modular, phased approach explored in this thesis, where contrastive pretraining is first used to build a robust image encoder, followed by a separate fine-tuning stage that adapts a text decoder for radiology report generation.

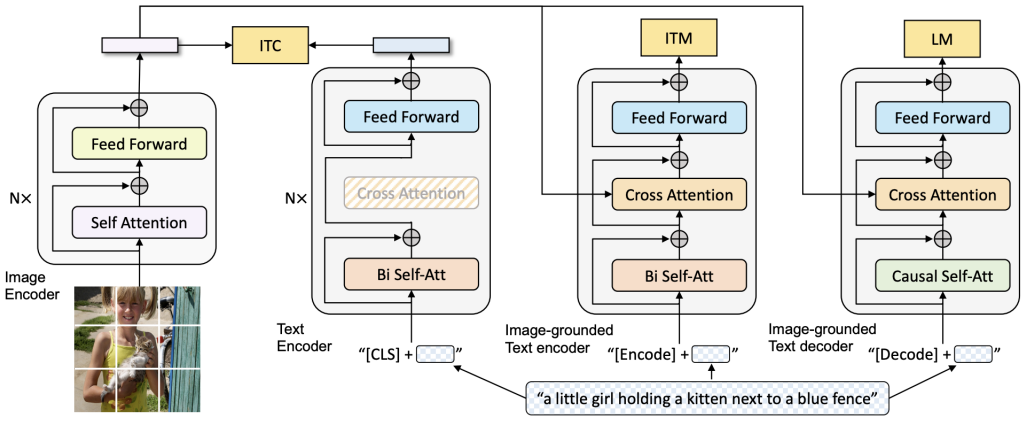


Figure 1.8: Overview of the BLIP pretraining framework. The architecture supports three vision-language pretraining objectives: Image-Text Contrastive (ITC) leaning, Image-Text Matching (ITM), and Language Modeling (LM). Source: [LLXH22].

1.4.3 Adaptations of CLIP for the medical domain

As demonstrated by the presented foundational contrastive-based VLMs, the CLIP framework can be effectively adapted to the medical domain with minimal supervision. However, several extensions of the original architecture have been proposed to address the unique challenges associated with clinical data [ZLW+23]. Firstly, the complexity and specificity of medical language demand more robust methods for learning shared image-text representations, often incorporating additional supervision to better optimize the contrastive objective. Furthermore, the limited availability of large-scale annotated medical datasets necessitates the use of advanced data augmentation strategies to improve model generalization. The following section explores these adaptations in greater detail, evaluates their performance, and discusses their relevance for further radiology-focused models.

1.4.3.1 Concept-Aware Semantic Matching Loss

The MedCLIP model, proposed by Wang et al. [WWAS22], is a vision-language contrastive pretraining approach specifically designed for the medical domain, which demonstrates strong performance in zero-shot prediction, supervised classification, and image-text retrieval tasks, with only 10% of the data required by prior state-of-the-art methods. Rather than relying on the traditional InfoNCE loss, MedCLIP introduces a novel contrastive objective called the **Semantic Matching Loss**, which leverages structured medical knowledge to redefine similarity between image and text samples.

To generate these improved alignments, MedCLIP uses a **medical knowledge extraction** module, that assigns semantic label vectors to unpaired images and texts. These label vectors, denoted as $\mathbf{l}_{img}$ and $\mathbf{l}_{text}$, represent medical conditions identified in the respective studies and are used to compute cosine similarities that reflect their semantic relatedness:

$$s = \frac{\mathbf{l}_{img}^{\top} \cdot \mathbf{l}_{text}}{\|\mathbf{l}_{img}\| \cdot \|\mathbf{l}_{text}\|}$$

These similarity scores are then normalized across the batch using a softmax function to produce **soft targets**, and the final Semantic Matching loss can be formulated as the average cross-entropy between the predicted similarities $\hat{y}$ of encoded image-text pairs and the soft targets y derived from their semantic labels:

$$\mathcal{L}_{\text{semantic}} = -\frac{1}{2N_{batch}} \sum_{i=1}^{N_{batch}} \sum_{j=1}^{N_{batch}} \left[y_{ij}^{v \to t} \log \hat{y}_{ij} + y_{ji}^{t \to v} \log \hat{y}_{ji} \right]$$

This loss not only introduces a more robust way to employ contrastive pretraining in medical settings, but also addresses one fundamental design issue of such models working with clinical data. Due to the complexity of radiology reports and the subtle differences between imaging studies, it is common for unrelated studies to share overlapping findings (Figure 1.9). MedCLIP's use of soft semantic targets helps reduce false negatives in such cases and enables effective training on unpaired data from heterogeneous sources. Achieving a Precision@1 (P@1) score of 45% on image-text retrieval and an average accuracy (ACC) of 54% in zero-shot classification on the MIMIC-CXR dataset, the proposed methodology demonstrates strong potential, but still may be limited by inaccuracies in the extracted semantic tags, particularly in cases where clinical negations or uncertainty expressions are missed, which can introduce noise into the semantic similarity supervision.

Building upon MedCLIP's ideas, this thesis explores the use of a semantic contrastive objective, which can significantly enhance alignment quality, especially in supervised settings.

1.4.3.2 Multi-View Supervision for radiographic understanding

CXR-CLIP [YGH+23] is a Vision-Language Pretraining (VLP) method specifically designed for chest X-ray interpretation. It enhances the discriminative capacity of both image and text encoders by leveraging a combination of image-text and image-label datasets. To address the scarcity of high-quality radiology reports, CXR-CLIP generates synthetic text prompts using radiologist-crafted templates applied

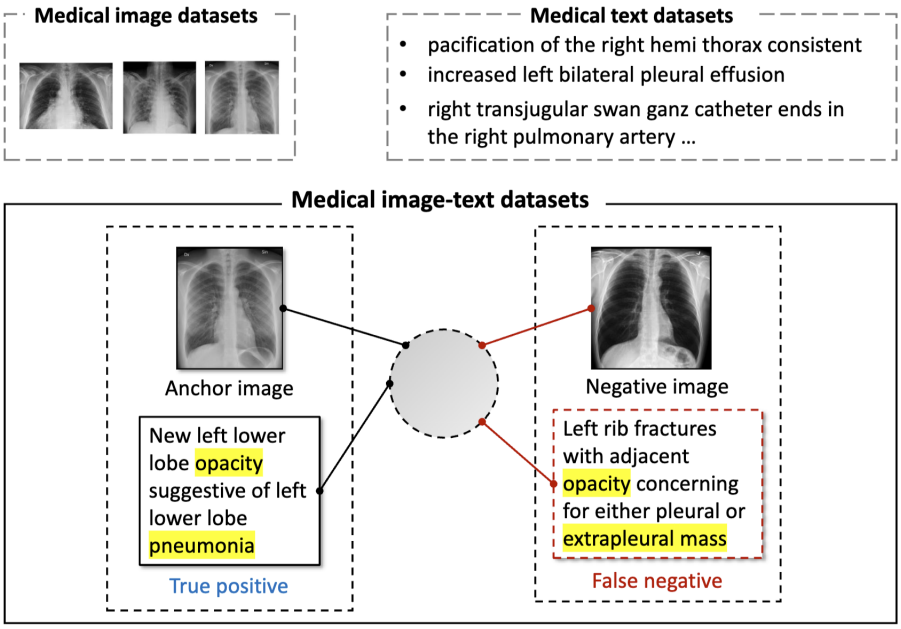


Figure 1.9: An example highlighting how semantically similar reports can appear in both positive and negative pairs. While the anchor image and its corresponding report form a true positive, the negative image-text pair contains overlapping findings (e.g., "opacity") that may cause false negatives if treated as unaligned. Source: [WWAS22].

to structured labels, producing natural language inputs that closely resemble clinical reports.

In addition to this contribution, CXR-CLIP expands on the research around **data-efficient learning** strategies, by suggesting the use of **Multi-View Supervision** (MVS). Beyond using original image-text pairs, the model leverages augmented or alternative versions of each modality and jointly trains on these variations to encourage consistent alignment, as outlined in Figure 1.10. Therefore, the InfoNCE-based contrastive loss, denoted as $\mathcal{L}_{CLIP}$ and defined in Formula 1.1, is applied across four combinations of image-text views and averaged, as illustrated in Formula 1.2. This improves the model's robustness and generalization in downstream tasks.

$$\mathcal{L}_{\text{MVS}} = \frac{1}{4}\left(\mathcal{L}_{\text{CLIP}}(U_1, V_1) + \mathcal{L}_{\text{CLIP}}(U_2, V_1) + \mathcal{L}_{\text{CLIP}}(U_1, V_2) + \mathcal{L}_{\text{CLIP}}(U_2, V_2)\right) \tag{1.2}$$

The framework further incorporates **self-supervision** within a CXR study. The Image Contrastive Loss (ICL) pulls image embeddings from the same study closer and pushes image embeddings from different studies apart, encouraging the encoder capture study-level diversity. Similarly, the Text Contrastive Loss (TCL) aligns multiple textual views from the same report (e.g., "Findings" and "Impression") while distancing them from unrelated texts. These two objectives are expressed as:

$$\mathcal{L}_{ICL} = \mathcal{L}_{CLIP}(V^1, V^2) \qquad \mathcal{L}_{TCL} = \mathcal{L}_{CLIP}(U^1, U^2)$$

The overall loss function combines these objectives with the multi-view align-

ment loss $\mathcal{L}_MVS$, weighted by hyperparameters λ_I and λ_T:

$$\mathcal{L} = \mathcal{L}_{MVS} + \lambda_I \cdot \mathcal{L}_{ICL} + \lambda_T \cdot \mathcal{L}_{TCL}$$

CXR-CLIP achieves strong results across multiple radiology benchmarks, particularly in low-resource settings. It reaches 21.6% Recall@1 on MIMIC-CXR and 62.8% accuracy on CheXpert5x200 in zero-shot classification. Additionally, fine-tuning a lightweight classifier on top of the pretrained image encoder further improves classification performance.

The methodology proposed by CXR-CLIP is highly relevant to this research, as it highlights the benefits of Multi-View Supervision for robust image–text alignment. Its improved retrieval and classification performance suggests strong potential for adaption in a radiology report generation system.

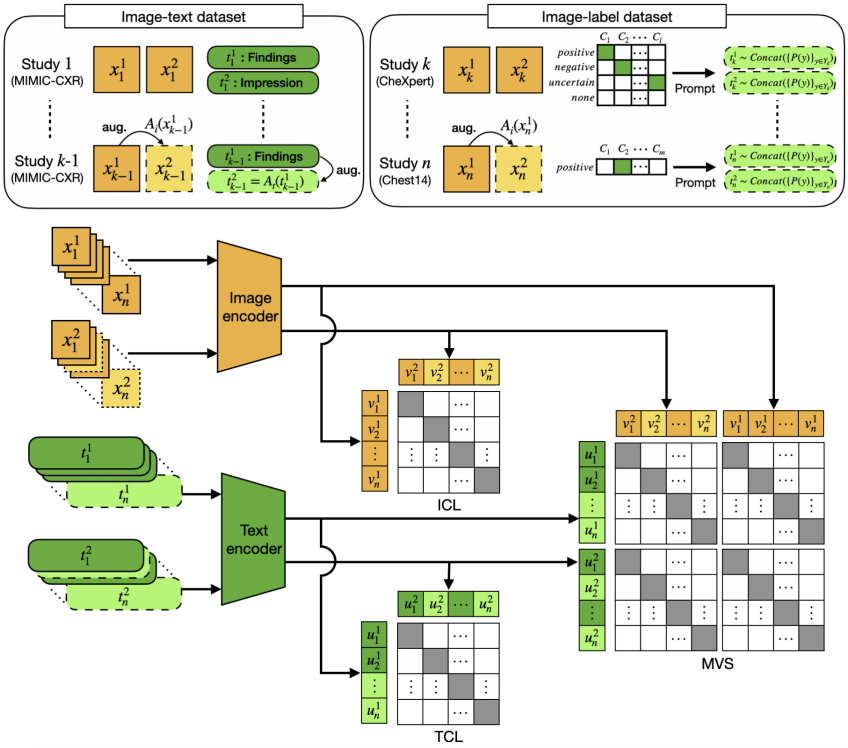


Figure 1.10: Overview on the CXR-CLIP training framework from [YGH+23]. The model leverages both image-text and image-label datasets and incorporates three contrastive objectives-ICL, TCL, and MVS-using paired and augmented views of images and texts across studies.

1.5 Related work on radiology report generation with encoder-decoder models

This section surveys leading methods in the field of radiology report generation that have achieved competitive results and are most representative of the design choices explored in this thesis. While unified by the shared encoder-decoder architectural paradigm, these methods differ in how they model the image and text

modalities and in how they conceptualize the generation task, ranging from extensions of image captioning, to retrieval-augmented methods or knowledge-infused generation frameworks [SCSM24]. Each approach provides valuable insights into learning cross-modal representations and highlights the ongoing evolution of report generation strategies in medical imaging.

1.5.1 R2Gen: report generation via Memory-Driven Transformers

Proposed by Chen et al. [CSCW20], **R2Gen** (Figure 1.11) presents a CNN–Transformer architecture specifically designed for automated radiology report generation. The model integrates a ResNet-101 CNN backbone for image encoding and a Transformer-based decoder to generate corresponding free-text reports. The extracted visual features are first embedded as hidden states and then passed to the decoder, which uses Multi-Head Cross-Attention to attend to these image-derived representations during the report generation process.

Two core innovations distinguish R2Gen from traditional encoder–decoder models: the **Relational Memory (RM)** module and the **Memory-driven Conditional Layer Normalization (MCLN)**. The Relational Memory module is designed to capture long-range dependencies between generated tokens by maintaining a matrix M_t that encodes contextual memory across decoding steps. At each time step t, the memory matrix M_{t-1} is updated using multi-head attention over itself and the embedding of the previously generated token y_{t-1}:

$$Q = M_{t-1}W_q, \quad K = [M_{t-1}; y_{t-1}]W_k, \quad V = [M_{t-1}; y_{t-1}]W_v$$

The updated memory representation is refined through residual connections and a gating mechanism to mitigate vanishing gradients:

$$\tilde{M}_t = f_{\text{MLP}}(Z + M_{t-1}) + Z + M_{t-1}, \quad M_t = \sigma(G_t^f) \odot M_{t-1} + \sigma(G_t^i) \odot \tanh(\tilde{M}_t)$$

Here, Z is the output of the attention block, and G_t^f, G_t^i represent the forget and input gates, respectively.

In parallel, the **MCLN** mechanism introduces dynamic modulation of layer normalization using memory-guided scaling and shifting parameters. At each decoding layer, the memory output M_t is aggregated into a vector m_t and used to compute:

$$\hat{\gamma}_t = \gamma + f_{\text{MLP}}(m_t), \quad \hat{\beta}_t = \beta + f_{\text{MLP}}(m_t)$$

These parameters are then used to normalize the intermediate activations in each Transformer layer, ensuring better conditioning of the generative process on the memory state.

R2Gen was evaluated on two widely used benchmark datasets, IU X-Ray [DKR+16] and MIMIC-CXR [JPB+19]. It achieved BLEU-4 scores of 0.165 and 0.103, respectively, outperforming baseline Transformer models with relative improvements of 17.6% and 12.1%. This performance positions R2Gen as one of the foundational models in the ARRG landscape.

However, despite its strengths, R2Gen has several limitations. It suffers from repetitive text generation and struggles with subtle clinical distinctions in images. Furthermore, its reliance on CNNs for feature extraction introduces architectural

complexity and requires an additional feature mapping step before decoding. These drawbacks have motivated newer approaches that replace the CNN encoder with **Vision Transformers (ViT)**, which model image patches as sequences and natively output token embeddings that can be directly consumed by a Transformer decoder. This shift toward fully Transformer-based architectures simplifies the pipeline and improves semantic alignment between visual features and generated text.

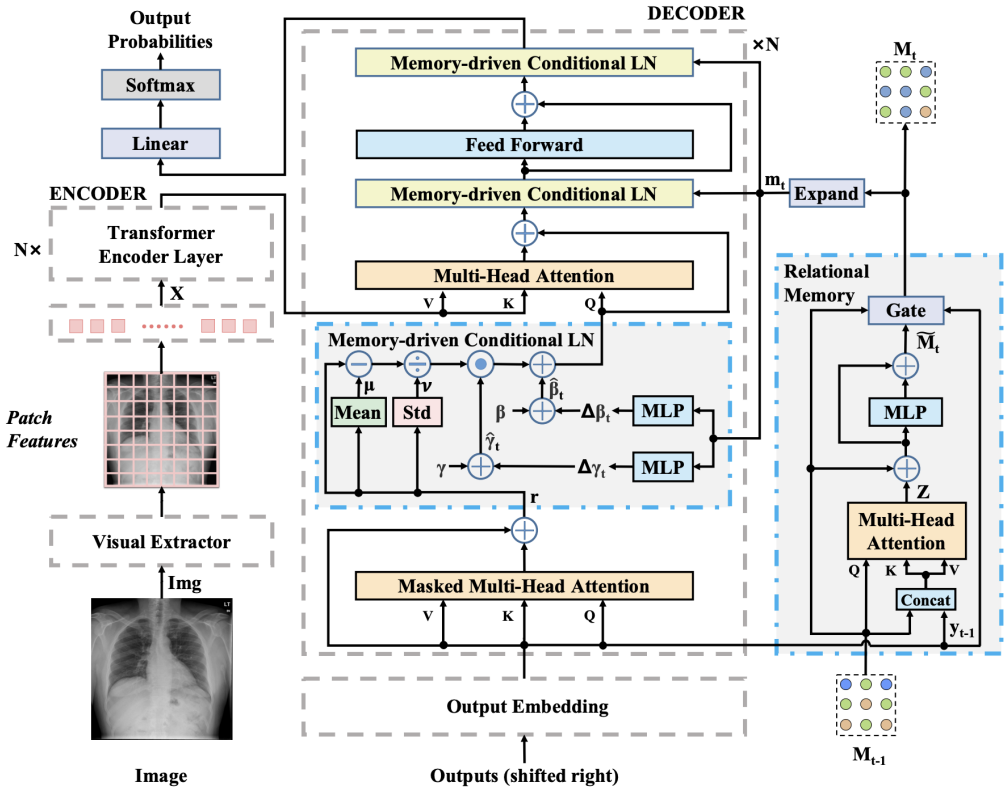


Figure 1.11: Architecture of the R2Gen model [CSCW20]. The framework consists of three main components: a visual feature extractor, a Transformer encoder, and a relational memory-augmented Transformer decoder. Gray dashed boxes indicate the encoder-decoder structure, while blue highlights show the memory-driven conditional layer normalization modules integrated into the decoder.

1.5.2 Retrieval-based report generation with CXR-RePaiR

CXR-RePaiR (Contrastive X-ray Report Pair Retrieval) [EKK+21] proposes a retrieval-based alternative to radiology report generation by reframing the task as one of identifying the most relevant report or report fragments from a large corpus, rather than synthesizing text from scratch. As seen in Figure 1.12, the approach leverages a CLIP-style contrastive pretraining objective to jointly embed chest X-rays and corresponding reports, enabling effective matching through vector similarity.

Given a corpus of radiology reports $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ and a test chest X-ray image x, the model encodes the image into an embedding $I = h(x)$ and each report $r \in \mathcal{R}$ into a text embedding $T = g(r)$ using pretrained image and text encoders $h(\cdot)$ and $g(\cdot)$. The similarity between a report and the input image is computed as the dot product of the embeddings, as CLIP suggested:

$$f(r, x) = g(r) \cdot h(x) = T \cdot I$$

The final generated report $\hat{p}$ is selected as the report $r \in \mathcal{R}$ that maximizes the similarity:

$$\hat{p} = \arg\max_{r \in \mathcal{R}} f(r, x)$$

To provide more control over the granularity of the output, the CXR-RePaiR framework includes several retrieval variants. The first variant, **CXR-RePaiR-R**, returns the full report $\hat{p}$ from the corpus that is most semantically aligned with the input image. The **CXR-RePaiR-k** variant extends this by retrieving $\hat{p}$ as the top-k sentences from the sentence set $\mathcal{S}(\mathcal{R})$ that maximize the similarity function $f(r, x)$. Lastly, **CXR-RePaiR-Select** introduces a dynamic sentence selection mechanism based on diagnostic relevance. It uses the **CheXbert labeler** [SJR+20] to adaptively select the sentences that compose the final report, depending on the identified clinical findings.

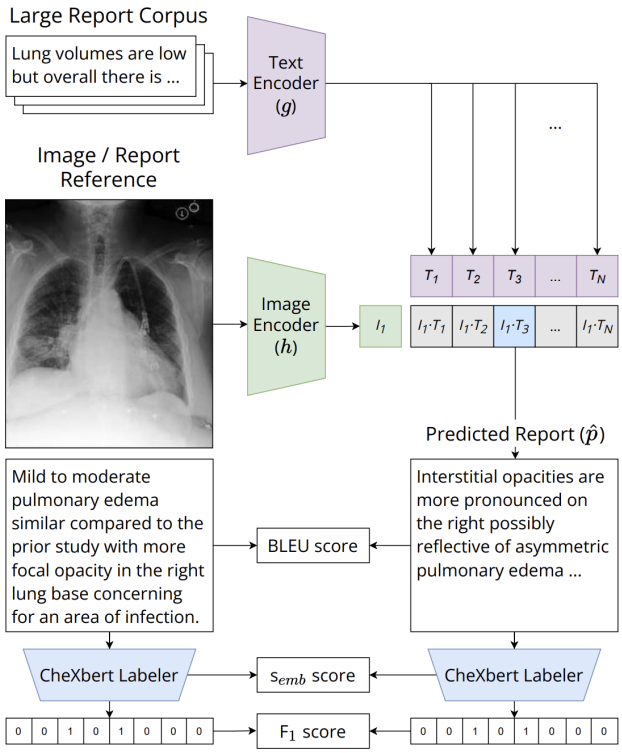


Figure 1.12: Retrieval-based report generation from [EKK+21]. Chest X-rays and reports are encoded using pretrained CLIP encoders, and the most similar report $\hat{p}$ is retrieved from a large corpus. Evaluation uses BLEU, $F1$ and s_{emb} scores based on CheXbert-extracted labels.

CXR-RePaiR achieves strong results across both internal and external benchmarks. It outperforms or matches prior state-of-the-art models on clinical accuracy metrics, with an average F1 score of 0.352 (+7.98%) on the external CheXpert dataset. Moreover, by using a compressed report corpus, the model generates reports up to 70% faster than leading generative models, while maintaining similar clinical fidelity.

By decoupling report generation from autoregressive text modeling, CXR-RePaiR offers a robust and efficient solution better suited to the bounded nature of radiological findings and enables more scalable integration into real-world clinical workflows. However, it still suffers from similar limitations as the majority of the models in this field, lacking any information diversity awareness in the process of retrieving relevant sentences, or struggling to generalize to rare or unseen pathologies.

1.5.3 RepsNet: a contrastive approach for VQA-based report generation

Building upon the capabilities of contrastive learning for aligning visual and textual modalities, Tanwani et al. [TBF22] proposed **RepsNet**, a unified multi-task encoder-decoder framework for radiology report generation. Unlike conventional sequence-to-sequence models, RepsNet formulates the task as a **Visual Question Answering** (VQA) problem, where reports are decomposed into question-asnwer pairs that the model learns to answer based on input images. This formulation allows RepsNet to support both *close-ended* classification-style answers (e.g., modality for obtaining the imaging study or view plane) and *open-ended* descriptive answers (e.g., clinical findings).

Formally, given an input image $x \in (X)$ and a set of natural language questions $\mathbf{q} = \{q_1, \ldots, q_s\} \in \mathcal{Q}$, the goal is to generate a set of corresponding answers $\mathbf{y} = \{y_1, \ldots, y_s\} \in \mathcal{Y}$, where each answer may be either categorical or open-ended. The model learns parameters Θ to maximize the conditional likelihood:

$$\Theta^* = \arg\max_{\Theta} \sum_{i=1}^{s} \log P_{\Theta}(y_i \mid x, q_i)$$

This objective is realized through an encoder-decoder setup, where the encoder $f_{\theta_{\text{enc}}} : \mathcal{X}, \mathcal{Q} \rightarrow \bar{X}, \bar{Q}$ projects images and questions into joint cross-modal representation space, and the decoder $h_{\theta_{\text{dec}}} : \bar{X}, \bar{Q}, \bar{C} \rightarrow \mathcal{Y}$ generates answer sequences conditioned on encoded representations and a prior context $\bar{C}$.

To enable strong cross-modal alignment, RepsNet adopts a CLIP-inspired encoder setup that integrates a ResNeXt-101 image encoder with a BERT-based text encoder. These components are trained using a bidirectional contrastive loss that pulls matching image-question-answer triplets closer in a shared embedding space while pushing apart mismatched pairs based on cosine similarity. A key component of this alignment process is the **Bilinear Attention Network** (BAN), which fuses image and question features before computing their alignment with the corresponding textual answers.

During generation, RepsNet employs a **retrieval-augmented decoding** strategy. Specifically, given the encoded representation of the current image-question pair, it retrieves the top-K most similar answer embeddings from the training set as prior context $\bar{C}$. These fragments serve as clinically relevant examples that guide the decoder toward fluent and medically coherent outputs. The decoder is built on GPT-2, modified to include attention over both the image-question encoding and the retrieved prior context, enabling it to produce more fluent, informative reports.

For closed-ended questions, the fused image-question embeddings are passed to a classification head to predict from a fixed set of answer categories. For open-ended

questions, the GPT-2 decoder generates the answer autoregressively. The two flows are illustrated in Figure 1.13.

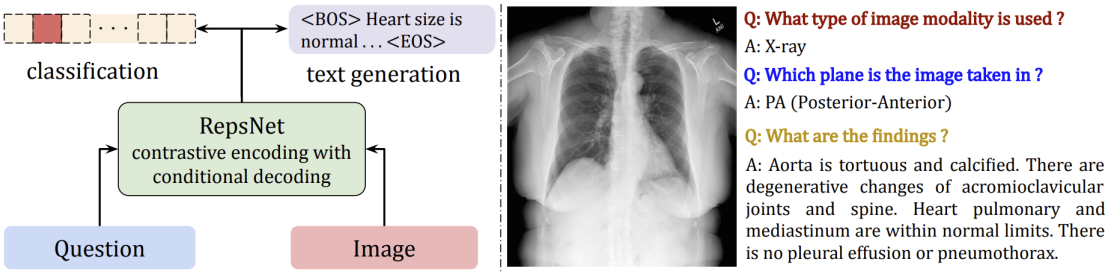


Figure 1.13: VQA-based report generation from [TBF22]. Given an image and diagnostic questions, RepsNet answers close-ended queries via classification and generated open-ended responses using a contrastive encoder and retrieved context to guide its GPT-2 decoder.

RepsNet was evaluated on VQA-Rad [LGBADF18] and IU X-Ray [DKR+16] benchmark datasets, achieving strong performance for both classification and report generation tasks. On IU X-Ray, it achieved a BLEU-4 score of 0.27, outperforming prior methods. Moreover, ablation studies confirmed that both contrastive encoding and prior context retrieval substantially contributed to its performance.

Although evaluated on different datasets than the ones leveraged in this work, RepsNet remains a relevant reference due to the conditional generation setup and its integration of contrastive learning for radiology report generation. However, the model exhibits several limitations, primarily originating from its reliance on VQA-style supervision. This task reformulation requires each training report to be parsed into structured question-answer pairs, which restricts applicability to datasets with templated reports. Furthermore, this format may limit the model's ability to capture important descriptive details typically described in free-form findings sections.

Chapter 2

`CLIP-XRGen`: A prompt-guided approach for Automated Radiology Report Generation with semantic concept alignment

This chapter presents the methodological framework of our proposed model for Automated Radiology Report Generation, titled **CLIP-XRGen**. The model approaches the task by leveraging contrastive vision-language pretraining within a conditional text generation paradigm. Building upon the strengths of the discussed established baseline models and specifically their adaptations of CLIP for related medical tasks such as image-text retrieval, image classification, and captioning, CLIP-XRGen further refines these domain-specific insights in order to achieve accurate and clinically meaningful text generation capabilities, with the potential to support real-world expert workflows.

The research and experiments detailed in this chapter aim to introduce a new approach to image-text alignment of chest X-rays and radiology reports [Mih25], alongside a method for adapting and applying it to text generation tasks. This approach employs a contrastive objective that leverages multi-view supervision and medical concept labels to ensure semantic understanding and accurate identification of clinical conditions, formulated as the **Multi-view Concept Similarity Loss** (MCSL). The rich representations learned by the encoders during the pretraining phase are subsequently used both to construct guidance prompts and to provide context for a decoder trained under teacher-forcing, enabling the generation of a descriptive text aligned with the imaging study. The performance of each component is evaluated individually and in the full end-to-end pipeline, with comparisons to related work to demonstrate its effectiveness and potential impact.

The central focus of this chapter is ultimately structured around two key objectives. First, we evaluate the effectiveness of the proposed contrastive learning method with semantic concept alignment in the context of medical datasets, with the goal of determining whether it improves the retrieval and classification capabilities of such a model. Second, we investigate the performance of image-grounded report generation with prompt guidance, with particular attention to the overall consistency and clinical utility of the generated texts, ensuring that critical findings requiring further expert analysis are explicitly mentioned.

In summary, these topics are addressed through the following research questions:

RQ1. To what extent does multi-view supervision combined with semantic concept alignment enhance the performance of contrastive vision-language models in learning rich, domain-adapted representations for medical data?

RQ2. Can a contrastive-based vision-language encoder be effectively integrated into a unified framework for both image understanding and text generation, while maintaining accuracy in the produced radiology reports?

2.1 Proposed Methodology

This section outlines the proposed methodology and presents the technical details for implementing CLIP-XRGen. We begin with an overview of model's workflow, highlighting how the individual modules interact to form a unified vision-language framework. Afterward, we introduce the original contrastive learning objective used during the pretraining phase, followed by an exploration of the classification capabilities of the resulting image encoder, in both zero-shot and few-shot settings. Finally, we describe the fine-tuning process for the downstream task of radiology report generation, detailing how the text decoder leverages prompt-based guidance and image-derived contextual features via Cross-Attention to generate reports in an autoregressive manner.

For clarity, although our prior conducted research focused exclusively on the encoder-side implementation, specialized on retrieval and classification tasks under the name **CLIP-XRad** [Mih25], in this work, we adopt the name **CLIP-XRGen** to denote the complete end-to-end framework. This includes not only the vision-language alignment stage but also the integrated report generation component. Therefore, throughout this chapter, both experimental configurations are discussed within this unified framework.

2.1.1 Designing a Vision-Language framework for Joint Understanding and Generation

As established in prior literature, and in accordance with the multimodal nature of the task, CLIP-XRGen is formulated as a Vision-Language Model with a hybrid architecture, comprising both a contrastive-based dual-encoder module, and a generative encoder-decoder module. This design follows a two-phase training paradigm: the first phase employs a discriminative objective to align image and text representations, while the second leverages these aligned representations to guide the text generation. Accordingly, the model consists of three main components, all built upon the Transformer architecture:

- **Image Encoder:** a **Swin Transformer Tiny** backbone is used, with a patch size of 4x4 and a shifting window size of 7x7. The input chest X-ray images are resized to 224x224, augmented using medical-specific transformations, and passed through the encoder to produce hierarchical visual representations.

The **[CLS]** token embedding from the final stage is extracted as the global image feature descriptor used for vision-language alignment and downstream tasks.

- **Text Encoder:** we employ a **ClinicalBERT** model, pretrained on medical vocabulary, to encode the radiology reports. The resulting textual embeddings are used during pretraining for contrastive alignment with visual embeddings, and later support zero-shot classification by enabling similarity-based matching with encoded label representations.
- **Text Decoder:** we use a **custom BERT-based** model with architectural adaptations for conditional report generation. This module leverages frozen encoders and predicted medical prompts to guide the generation process during fine-tuning.

As illustrated in Figure 2.1, the workflow of CLIP-XRGen is structured as follows. First, in the **contrastive pretraining phase**, the Image Encoder and Text Encoder are trained jointly to align their representations using a custom contrastive loss tailored to the medical domain. As a result, The Image Encoder can be effectively used for both image-text retrieval and image classification. Specifically, it supports **zero-shot classification** by comparing the similarity between the embeddings of each modality, or alternatively, can be extended with a **fine-tuned multi-label classification head** to predict the presence probabilities of CheXpert medical concepts [IRK+19] in the imaging study.

In the subsequent **generation phase**, the Image Encoder is frozen, and its output embeddings, together with prompt constructed from the predicted findings, serve as contextual inputs to guide the Text Decoder in generating free-text radiology reports.

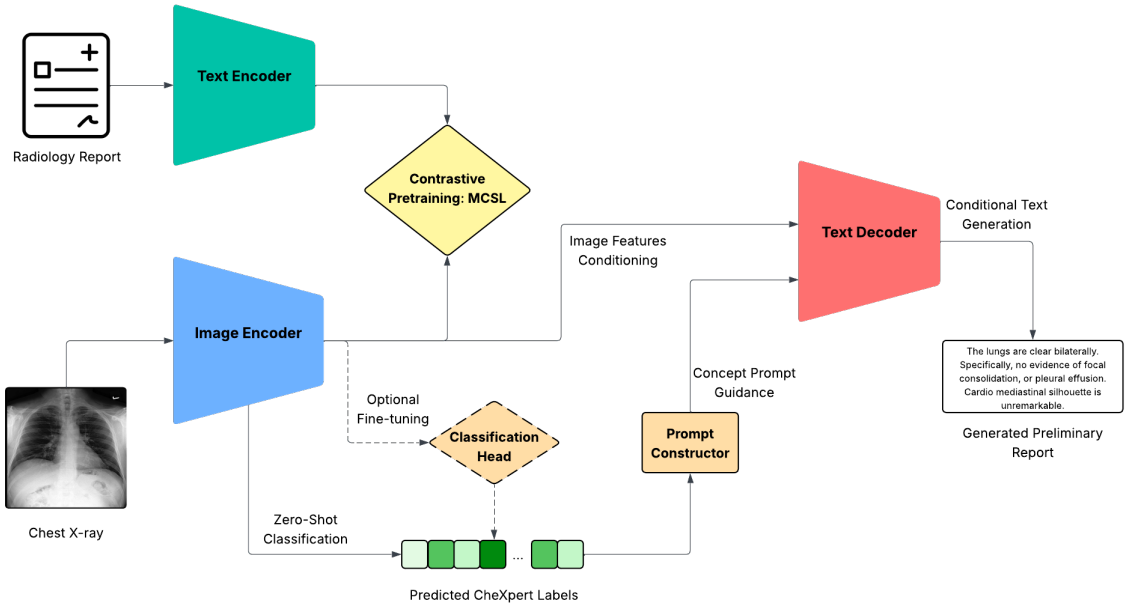


Figure 2.1: The CLIP-XRGen model workflow, highlighting its modular structure for vision-language pretraining and guided report generation.

2.1.2 Enhancing Language-Vision Pretraining with Medical Concept Alignment

This section provides an in-depth analysis of the contrastive pretraining strategy employed in the dual-encoder configuration of the hybrid CLIP-XRGen model, with a focus on its adaptation to the medical domain. The approach specifically addresses key challenges such as data scarcity and the semantic ambiguity inherent in medical image-text associations.

The goal of the pretraining phase is to learn a shared embedding space between image and text modalities, where semantically aligned pairs are drawn closer together, while unrelated or less relevant pairs are pushed further apart. This is accomplished by optimizing a contrastive loss function over multimodal representations. To formulate the goal, two fundamental aspects must be clearly defined:

1. Defining pairs: What constitutes a positive or negative pair in the context of the medical data.

2. Measuring similarity: How to quantify similarity between such pairs.

To address the difficulty of determining valid positive and negative pairs, particularly given that one-to-one mappings between images and reports are often imperfect, we incorporate an additional level of **semantic supervision** through the use of structured concept labels. Specifically, we utilize multi-label binary vectors extracted from medical datasets using the CheXpert labeler [IRK+19], which encodes the presence or absence of 14 common radiological findings. These label vectors serve as the foundation for defining similarity between samples, enabling a more accurate and nuanced estimation.

We adopt a **Multi-view Concept Similarity Loss** (MCSL), inspired by the contrastive supervision strategy proposed in MedCLIP [WWAS22], where instead of relying on a direct dot product between label vectors to estimate sample similarity, we employ the **Jaccard Index** to measure overlap between the concept label sets associated with each image-text pair. For a given batch size of N, let $l_i \in \{0,1\}^C$ represent the binary concept label vector for sample i, where C is the number of medical findings, The Jaccard similarity function between two samples i and j is given by Formula 2.1.

$$\text{sim}_{Jaccard}(i,j) = \frac{|l_i \cap l_j|}{|l_i \cup l_j|} = \frac{l_i \cdot l_j}{\|l_i\|_1 + \|l_j\|_1 - l_i \cdot l_j} \tag{2.1}$$

This pairwise similarity is computed for all sample pairs in a batch, resulting in a similarity matrix $J \in \mathbb{R}^{N \times N}$. To prevent the model from trivially matching samples to themselves, we zero out the diagonal of this matrix. Therefore, the distribution over potential matches for each anchor is obtained by applying a temperature-scaled softmax across each row of this matrix, as originally defined in Formula 1.1. This results in a **soft target matrix** $\hat{\mathbf{J}}$, where higher values indicate stronger semantic alignment.

$$\hat{\mathbf{J}}_{i,j} = \frac{\exp(\text{sim}_{Jaccard}(i,j)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}_{Jaccard}(i,j)/\tau)} \tag{2.2}$$

In Formula 2.2, i refers to the anchor sample, j is a candidate sample in the same batch from the other modality, and k indexes all possible candidates used to normalize the row.

To incorporate both hard and soft supervision, we construct the final **semantic target matrix** $\hat{\mathbf{S}}$, as shown in Figure 2.2, by blending the soft target values with the identity matrix, which enforces one-to-one alignment for the original image-text pairs. The scalar λ controls the relative weight of the semantic supervision, and the result is row-normalized to maintain a valid probability distribution. The elements of the matrix $\hat{\mathbf{S}}$ are computed as shown in Formula 2.3.

$$\hat{\mathbf{S}}_{i,j} = \frac{\mathbf{I}_{i,j} + \lambda \cdot \hat{\mathbf{J}}_{i,j}}{1 + \lambda} \tag{2.3}$$

In parallel, we compute a dot-product similarity between normalized image and text embeddings. Let e_i^{img} and e_j^{txt} denote the normalized embeddings of image i and text j. The dot-product similarity is given in Formula 2.4.

$$\text{sim}_{dot}(i,j) = e_i^{\text{img}} \cdot (e_j^{\text{text}})^\top \tag{2.4}$$

The resulting dot products are scaled by a learnable temperature parameter and normalized via softmax across each row, yielding the **logits matrix** $\hat{\mathbf{L}}$ (Formula 2.5). This matrix reflects the model's predicted alignment probabilities between image and text embeddings.

$$\hat{\mathbf{L}}_{i,j} = \frac{\exp(\text{sim}_{dot}(i,j)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}_{dot}(i,k)/\tau)} \tag{2.5}$$

To train the model, we compute the semantic alignment loss by comparing the predicted logits $\hat{\mathbf{L}}$ to the semantic targets $\hat{\mathbf{S}}$, using a **soft cross-entropy** formulation. The loss is computed bidirectionally, from image to text, and from text to image, with the latter involving transposed indexing, as seen in Formula 2.6.

$$\mathcal{L}_{semantic}^{i \to t} = -\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{\mathbf{S}}_{i,j} \cdot \log \hat{\mathbf{L}}_{i,k}, \quad \mathcal{L}_{semantic}^{t \to i} = -\sum_{j=1}^{N}\sum_{i=1}^{N} \hat{\mathbf{S}}_{j,i} \cdot \log \hat{\mathbf{L}}_{k,j} \tag{2.6}$$

The final semantic loss is obtained by averaging the bidirectional losses, ensuring balanced alignment from both modalities.

$$\mathcal{L}_{semantic} = \frac{1}{2}(\mathcal{L}_{semantic}^{i \to t} + \mathcal{L}_{semantic}^{t \to i}) \tag{2.7}$$

To further increase robustness and generalization, we apply this semantic loss (Formula 2.7) under a **multi-view supervision** (MVS) setup, as proposed in CXR-CLIP [YGH+23] and formalized earlier in Formula 1.2. Each input sample is augmented to produce additional views I' and T', resulting in four distinct image-text pairings, as illustrated in Figure 2.3. The semantic loss is computed for each combination and averaged to obtain the final **Multi-view Concept Similarity Loss**, from Formula 2.8.

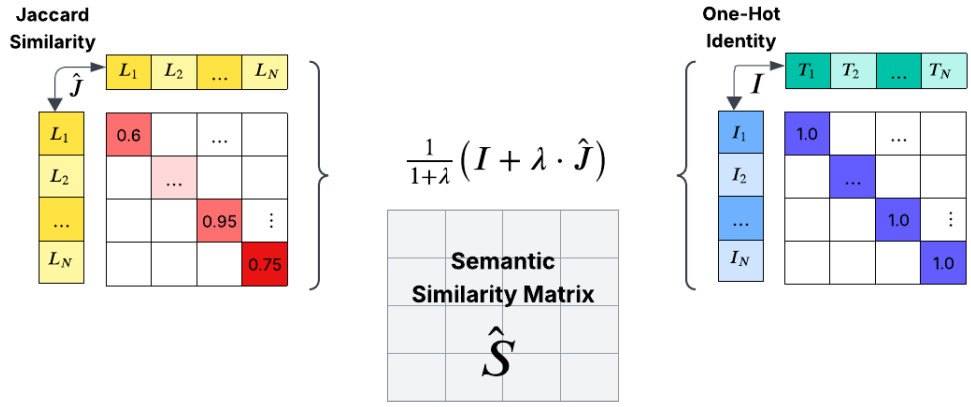


Figure 2.2: Semantic Similarity Matrix computation. The targets are computed by blending the Jaccard matrix $\hat{\mathbf{J}}$ with the identity matrix $\mathbf{I}$, weighted by λ.

$$\mathcal{L}_{\text{MCSL}} = \frac{1}{4}\left(\mathcal{L}_{\text{semantic}}(I,T) + \mathcal{L}_{\text{semantic}}(I',T) + \mathcal{L}_{\text{semantic}}(I,T') + \mathcal{L}_{\text{semantic}}(I',T')\right) \quad (2.8)$$

The aim of our proposed contrastive objective is not just to match image-text pairs, but to train the model to recognize when two samples are **semantically similar** based on their medical content. By supervising the model with concept-level labels, it learns that image-text pairs sharing more medical findings should be placed closer together in the embedding space. This enables the system to go beyond surface-level representation learning and instead develop an understanding of clinical data, even when samples are not directly paired in the dataset.

In addition, this approach does not rely exclusively on manually annotated datasets. The CheXpert labeler can automatically extract findings from any radiology text, which makes it possible to apply this training strategy to unpaired image and text datasets, or even image-only datasets with generated labels. This flexibility allows scaling the method to longer, more robust pretraining phases.

2.1.3 Leveraging Pretrained Encoders for Image Classification

Given the strengths of the learned multimodal embeddings with medical concept alignment, we further investigate the potential of the CLIP-based configuration of our approach to perform **multi-label classification** on chest X-ray images. The task involves predicting the presence or absence of the 14 clinical conditions defined by the CheXpert label set, which represent common radiological findings. Due to the inherent complexity and ambiguity of medical imaging, some labels are often marked as uncertain (with a value of -1), reflecting cases that typically require re-examinations. In this study, we exclude these uncertain labels by masking them as 0, focusing our efforts on confidently annotated positive and negative cases. While various strategies for handling uncertainty have been proposed in the literature [IRK+19], our choice aligns with the capabilities of our pretrained encoders, which are primarily optimized for clear semantic distinctions.

We evaluate two complementary strategies for the auxiliary classification task:

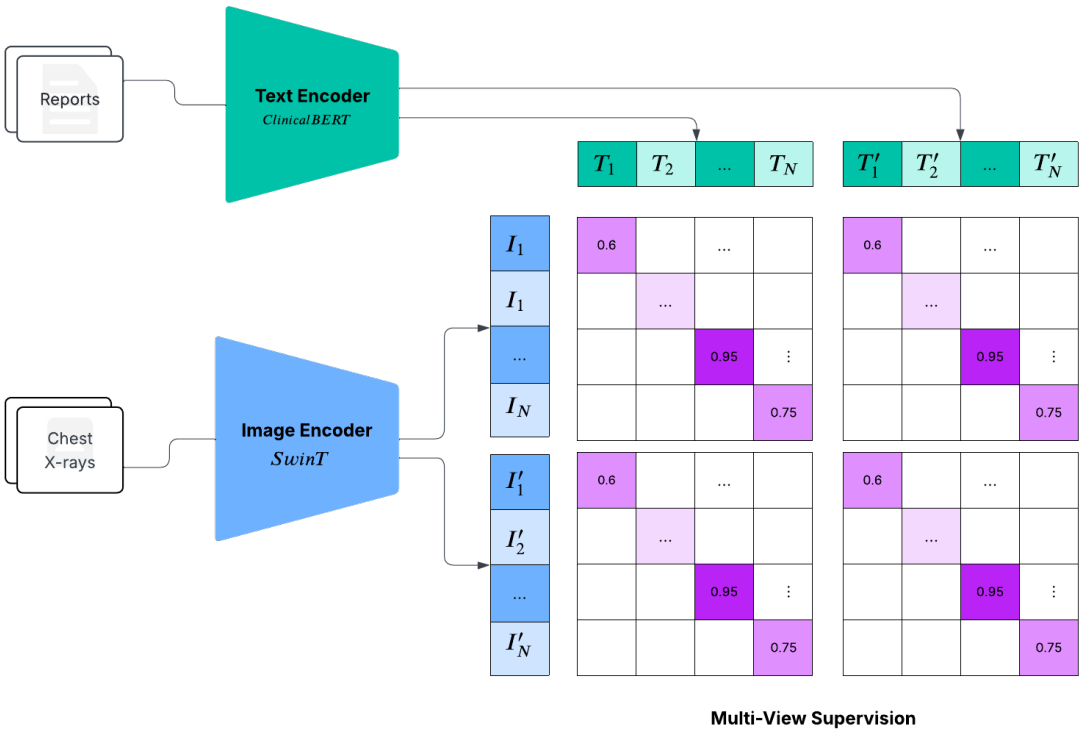


Figure 2.3: Multi-View Supervision Strategy adopted in the pretraining phase. Each image and text sample is augmented to create four cross-modal pairings.

a **zero-shot classification** approach leveraging image-text similarity, and a **supervised fine-tuning** approach where a classification head is trained on top of the image encoder. The two variants are compared in order to determine whether introducing a supervised training step provides a meaningful performance gain.

2.1.3.1 Zero-Shot Classification

Zero-shot classification refers to the ability of a model to assign labels to previously unseen examples by relying solely on learned semantic relations, rather than explicit training on those labels. In our case, this is achieved by computing similarity scores between the image embeddings and textual representations of class-specific label prompts, allowing the model to predict the presence of each medical condition directly from the shared multimodal embedding space learned during contrastive pretraining.

This formulation naturally decomposes the original multi-label classification task into 14 independent **binary-classification** sub-tasks, each corresponding to a specific CheXpert finding. Let $e_{img} \in \mathbb{R}^d$ denote the image embeddings produced by the vision encoder. For each medical concept $c_i \in \{c_1, c_2, \ldots, c_{14}\}$, we define two text embeddings: one representing the positive form of the concept (e.g., "Cardiomegaly present") and one representing the negative form (e.g., "No Cardiomegaly"). These are denoted as $e_{c_i}^{pos} \in \mathbb{R}^d$ and $e_{c_i}^{neg} \in \mathbb{R}^d$, respectively. As presented in Table 2.1, a diverse set of prompt templates is used to construct the textual embeddings, ensuring the model can reliably identify medical concepts expressed in different forms. An overview of the prompt construction pipeline is illustrated in Figure 2.4.

To evaluate the model's confidence, we compute the cosine similarity between the image embedding and each of the two corresponding text embedding vectors (Formula 2.9).

$$s_i^{pos} = \frac{e_{img} \cdot e_{c_i}^{pos}}{\|e_{img}\| \cdot \|e_{c_i}^{pos}\|}, \quad s_i^{neg} = \frac{e_{img} \cdot e_{c_i}^{neg}}{\|e_{img}\| \cdot \|e_{c_i}^{neg}\|} \tag{2.9}$$

As shown in Formula 2.10, these similarity scores are then scaled by the learned temperature parameter τ and passed through a softmax function to produce a probability for estimating the presence of the condition.

$$P(c_i = 1 \mid e_{img}) = \frac{\exp(s_i^{pos}/\tau)}{\exp(s_i^{pos}/\tau) + \exp(s_i^{neg}/\tau)} \tag{2.10}$$

The resulting probability expresses the model's confidence that a given medical condition is present in the imaging study. However, in order to obtain a binary decision indicating the presence or absence of the condition, the continuous probability must be converted using a thresholding strategy, addressed in the evaluation phase of this thesis.

Concept Label	**Positive Prompt**	**Negative Prompt**
Edema	Pulmonary edema is present.	No convincing signs of pulmonary edema.
Cardiomegaly	Cardiac size appears enlarged.	Heart size is normal.
Enlarged Cardiomediastinum	Cardiomediastinal silhouette is widened.	Cardiomediastinal silhouette is unremarkable.
No Finding	Both lungs appear clear.	-

Table 2.1: Examples of positive and negative prompts for each concept label, used in zero-shot classification to improve robustness, following the formulation proposed in CXR-CLIP [YGH+23].

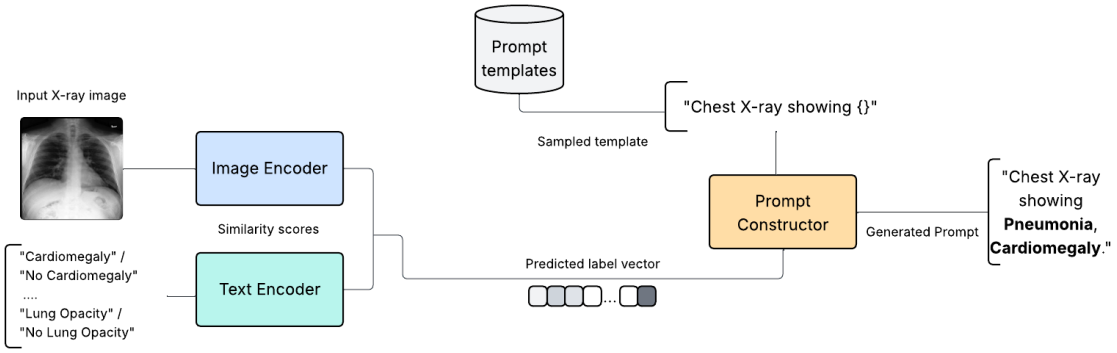


Figure 2.4: Prompt construction pipeline in the context of the zero-shot classifier. The similarity scores computed between the image and prompt pairs describing the absence or presence of each concept determine the predicted label vector values, which are then formatted in a sampled prompt template.

2.1.3.2 Feature Extraction Fine-Tuning

Feature extraction fine-tuning is a **Transfer Learning** strategy in which a pretrained model is used to extract high-level representations from input data, while a task specific head is trained on top of these frozen features. In the context of our task, we utilize the pretrained image encoder to extract visual embeddings from chest X-ray images, and train a classification head in a supervised manner to perform **multi-label prediction** over the 14 CheXpert findings.

To allow the model to capture possibly more complex dependencies between visual features and diagnostic labels, we employ a **Multi-Layer Perceptron** (MLP) as the classification head, as illustrated in Figure 2.6. This component takes the frozen image embeddings as input and outputs a probability for each class, which are subsequently used to construct a prompt, as shown in Figure 2.5.

Formally, let $e_{img} \in \mathbb{R}^d$ denote the image embeddings produced by the frozen image encoder. The MLP classifier is a function $f : \mathbb{R}^d \to \mathbb{R}^C$ where $C = 14$ is the number of medical conditions defined in the label set. The output of this classifier is passed through a **sigmoid activation** to obtain class-wise probabilities, as in Formula 2.11.

$$\hat{c} = \sigma(f(e_{img})) \in [0, 1]^C \tag{2.11}$$

Each dimension $\hat{c}_i$ represents the predicted probability that condition c_i is present in the image. To train the classifier, we use a **binary cross-entropy** loss, applied independently to each label.

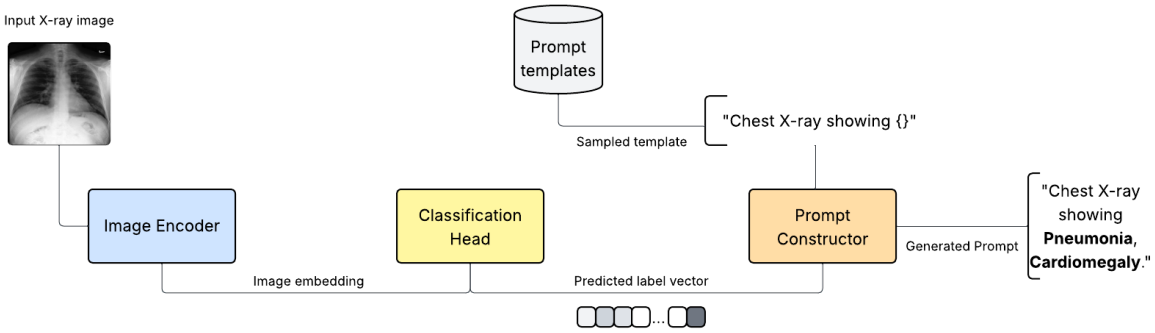


Figure 2.5: Prompt construction pipeline using a classification head. The image encoder extracts visual embeddings from the input X-ray, which is passed through a supervised classification to determine the present concepts. The class names are then introduced in the prompt template to compute the final generated guidance prompt.

2.1.4 Prompt-Guided Report Generation conditioned on Aligned Visual Representations

To enable Automated Radiology Report Generation (ARRG), we extend our CLIP-XRGen framework into an encoder-decoder configuration, where a modified language decoder generates free-text reports conditioned on both visual features and predicted medical prompts. The following section further provides an architectural

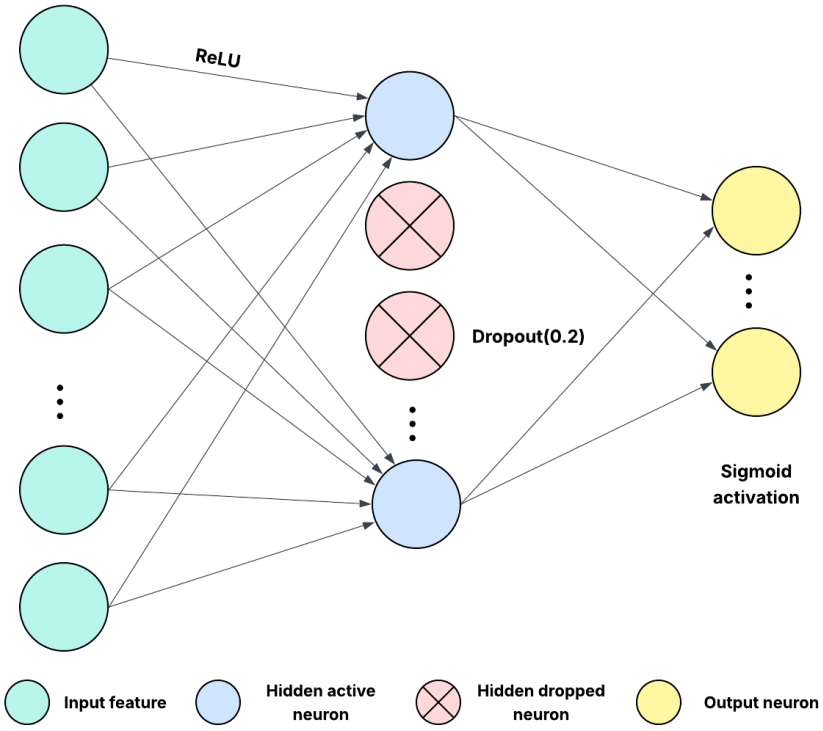


Figure 2.6: Structure of MLP classifier head one hidden layer, ReLU activation, Dropout regularization ($p = 0.2$), and sigmoid outputs for multi-label prediction.

overview of the decoder component and describes how it integrates with the pretrained modules in order to obtain coherent and clinically accurate descriptions of imaging studies.

The decoder builds upon the same **ClinicalBERT** backbone as used in the text encoder. Originally, BERT [DCLT19] is structured as a multi-layer Transformer model composed of stacked encoder blocks, each containing Multi-Head Self-Attention (MHSA) and Feedforward layers. It is primarily designed for bidirectional language understanding tasks, such as sentence classification, named entity recognition, or masked language modeling. Therefore, it does not support autoregressive generation out of the box, as it lacks causal masking and does not condition on external context.

Despite this, BERT offers the possibility to be converted in a decoder-only Transformer module. We make two primary architectural modifications, inspired by BLIP's unified design [LLXH22]. These changes are summarized in Figure 2.7, and are detailed as follows.

Cross-Attention Extension. We extend each Transformer block in BERT by inserting a Cross-Attention module after the original Multi-Head Self-Attention layer, allowing the decoder to attend to image embeddings extracted from the pretrained vision encoder. Formally, let $X \in \mathbb{R}^{n \times d}$ be the sequence of encoder hidden states, and $V \in \mathbb{R}^{m \times d}$ the image embeddings. Cross-Attention computes attention weights between each position in X and the image context V, conditioning token generation on visual semantics, aligned with text representations.

Language Modeling Head. We attach a Language Modeling (LM) head on top of the decoder. This consists of a linear projection layer that maps the final hidden states to vocabulary logits, enabling next-token prediction. The decoder generates tokens autoregressively, starting from a special token introduced into the tokenizer vocabulary to mark the start of the report generation, in our case being the `<report>` token.

In addition to visual context from image embeddings, the generation process is also supported by prompt guidance, based on the predicted medical findings from the chest X-ray image. While these prompts are excluded from the loss computation via attention masking, they still influence generation through Self-Attention, acting as soft conditioning signals that encourage the model to mention relevant conditions and assist experts in focusing their review.

As training objective, we apply the standard **cross-entropy loss** token-wise between predicted and ground-truth sequences, using a **Teacher Forcing** setup, relying on feeding ground-truth tokens at each decoding step. Let $\hat{y}_t$ be the model's probability distribution at timestep t, and y_t the corresponding target token. The loss is computed as seen in Formula 2.12.

$$\mathcal{L}_{CE} = -\frac{1}{T}\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, X) \tag{2.12}$$

At inference time (Algorithm 1), all encoder-side components, including the vision encoder, projection layer, and optionally the classification head, remain frozen. The input X-ray image is encoded into dense visual embeddings, projected to the decoder's hidden size, and reshaped as a token sequence. Predicted medical findings are formatted into a structured prompt, tokenized, and passed to the decoder alongside an attention mask over the image features. The decoder generates the report auto-regressively using Beam Search to ensure coherence and diversity of valid outputs.

Input: X-ray image X, Frozen image encoder f_{img}, Image projection f_{proj}, Classification module f_{cls}, Prompt constructor $\mathcal{P}$, Text decoder f_{dec}, Tokenizer $\mathcal{T}$
Output: Generated radiology report $\hat{R}$

$v \leftarrow f_{\text{img}}(X)$
$v \leftarrow f_{\text{proj}}(v)$; `/* Project image embeddings to decoder hidden size */`
$v \in \mathbb{R}^{B\times D} \rightarrow \tilde{v} \in \mathbb{R}^{B\times 1\times D}$; `/* Add sequence dimension */`
$\hat{y} \leftarrow f_{\text{cls}}(v)$
$p \leftarrow \mathcal{P}(\hat{y})$
$t_{\text{input}} \leftarrow \mathcal{T}(p$ + `"<report>"`$)$; `/* '+' represents concatenation */`
for $i = 1$ **to** B **do**
  for $j = 1$ **to** L_v **do**
    $a_{\text{enc}}[i][j] \leftarrow 1$; `/* Set attention mask to 1 for all positions */`
  end
end
$\hat{R} \leftarrow f_{\text{dec}}(t_{\text{input}},\ v,\ a_{\text{enc}})$
return $\hat{R}$

Algorithm 1: Inference pass of the CLIP-XRGen model. The model decodes a report conditioned on both extracted image features and constructed concept prompts for guidance.

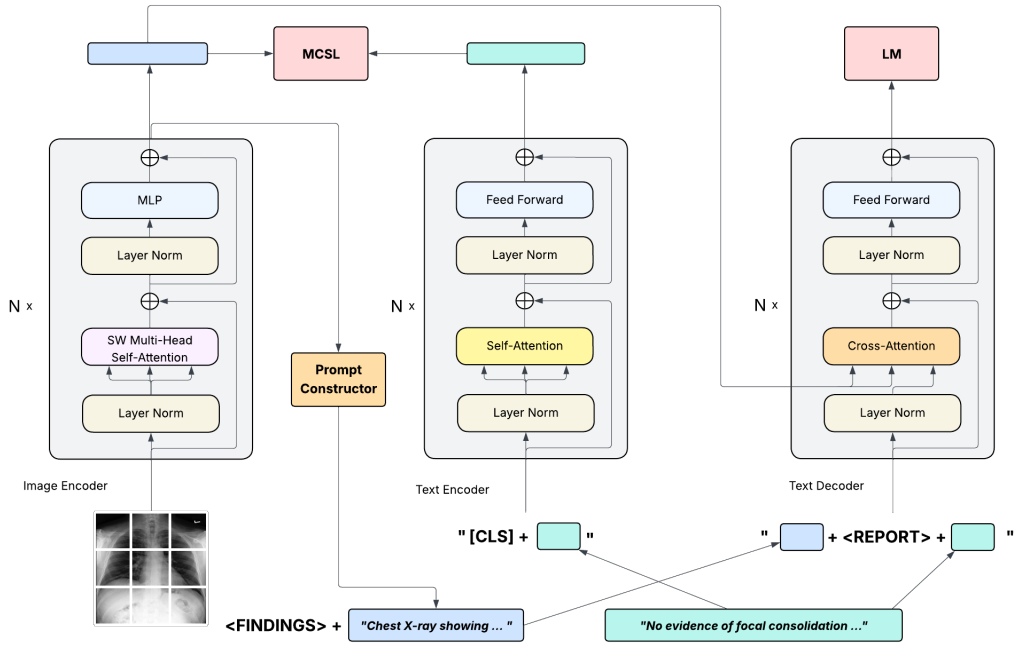


Figure 2.7: Illustration of the proposed hybrid CLIP-XRGen architecture. The model integrates within a unified framework a vision encoder, a text encoder, and a text decoder extended for prompt-guided conditional generation.

2.1.5 Performance evaluation

In order to comprehensively assess the effectiveness of the proposed CLIP-XRGen model, we use a modular evaluation strategy, that reflects the hybrid nature of its architecture. Given that the model undergoes distinct phases of training (contrastive pretraining, auxiliary classification fine-tuning and end-to-end text generation), each phase requires specific evaluation criteria aligned with its intended objectives.

This staged evaluation approach enables us to gain a more granular understanding of how well each component contributes to the overall system performance, while also highlighting potential limitations and areas of improvement in both representation learning and generation. Moreover, this separation allows for fair comparisons with prior work that addresses individual subtasks, rather than the full pipeline of radiology report generation.

The subsequent sections detail the evaluation methodologies used at each stage, including the motivation behind choosing the specific metrics, based on the medical context and the characteristics of the used datasets, presented further during Experimental Evaluation.

2.1.5.1 Contrastive Pretraining evaluation: Image-Text Alignment and Retrieval

To evaluate the performance of the contrastive vision-language encoders, we focus on assessing whether the learned image and text representations are aligned within the shared semantic space. In this context, alignment implies that a representation from one modality (e.g., chest X-ray image) should be the closest to its corresponding counterpart from the other modality (e.g., associated radiology report) in

the latent space. The degree of alignment is measured using cosine similarity, which quantifies the angular distance between embedding vectors. Therefore, in line with prior work in literature, we adopt a two-fold retrieval evaluation strategy, reflecting both the exact matching capabilities and the semantic richness of the learned embeddings.

First, we use the **Recall@K** metric to evaluate **exact image-to-text alignment**. This metric quantifies the proportion of image queries for which the correct textual report appears among the top-K most similar candidates retrieved based on cosine similarity.Formally, let N be the number of query images, r_i the ground truth report associated with the i-th image, and $R_{i,K}$ the set of top-K retrieved reports. The Recall@K score is computed as shown in Formula 2.13, where the indicator function $\mathbf{I}[\cdot]$ returns 1 if the correct report r_i is present in the top-K set, and 0 otherwise.

$$\text{Recall@K} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}[r_i \in R_{i,K}] \tag{2.13}$$

A higher Recall@K indicates stronger alignment between visual and textual modalities in terms of exact pair retrieval, measurable in the context of annotated datasets.

In addition to exact retrieval, we also evaluate the model's ability to perform **class-based semantic retrieval**, inspired by MedCLIP's methodology [WWAS22]. In our case however, we adapt this evaluation method to a **multi-label setting**, reflecting the nature of the datasets used in our experiments, where both images and reports can be associated with multiple diagnostic categories, closely mirroring real-world clinical scenarios. Therefore, retrieval is considered successful if any of the top-K retrieved reports share at least one overlapping label with the query image. This setup better captures the semantic alignment of embeddings, especially when evaluated on unpaired image-text collections.

To quantify this, we employ the **Precision@K** metric, which measures the proportion of retrieved reports that are semantically consistent with the image query, meaning that they have at least one shared medical concept. It is formally defined in Formula 2.14, where x_i is the i-th image, $label(x_i)$ the set of its diagnostic labels, and $R_{i,K}$ the top-K retrieved reports relevant in terms of the searched class.

$$\text{Precision@K} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\{r \in R_{i,K} \,|\, \text{label}(r) \cap \text{label}(x_i) \neq \emptyset\}|}{K} \tag{2.14}$$

However, it is important to note that, in a multi-label setup, top-1 Precision can appear higher than top-5 or top-10. This is because the first retrieved report may already match a label, while additional results might introduce noise. In contrast, Recall increases with larger retrieval windows, as more opportunities arise to capture the ground-truth associated pair with the query.

2.1.5.2 Auxiliary Multi-Label Classification evaluation: diagnostic label prediction

The model's ability to capture clinically meaningful patterns is also evaluated from the perspective of a multi-label classification task on CheXpert-derived medical labels. This evaluation is conducted in both configurations, for zero-shot classification and additional supervised fine-tuning. In both cases, the model outputs a

probability for each diagnostic label, indicating the likelihood of its presence.

Given the inherent class imbalance typical of medical datasets and the multi-label nature of the problem, we employ a combination of threshold-independent and threshold-dependent metrics to obtain a robust performance assessment.

We report **Accuracy** as the primary evaluation metric, defined in Formula 2.15. It offers a direct measure of overall correctness by evaluating the proportion of correctly classified samples after thresholding. While it can be sensitive to class imbalance, it remains one of the widely used metrics in related literature and offers an intuitive measure of overall correctness for predicting the present medical findings.

To complement accuracy and provide additional insight into the model's decision behavior, we also use **Area Under the ROC Curve** (AUC). This threshold-independent metric reflects the model's ability to distinguish between positive and negative cases, and can be interpreted as the probability that a randomly chosen positive instance will be ranked higher than a randomly chosen negative one.

Moreover, the additional standard classification metrics **Precision** and **Recall** are considered to better contextualize the trade-offs associated with using Accuracy as the primary evaluation metrics. While Accuracy provides a single aggregate score, it may be misleading in the presence of a skewed label distribution. Precision and recall help disentangle this by quantifying the model's tendency toward false positives or false negatives, respectively. These metrics offer a more nuanced perspective on model behavior and are formally defined in Formulas 2.16 and 2.17, where TP, FP, and FN denote the number of true positives, false positives, and false negatives.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.15}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.16}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.17}$$

To convert the model's probability outputs into binary predictions, a threshold value must be applied in order to distinguish when an example is marked as positive or negative. Since using a default fixed value (e.g., 0.5) would not reflect the actual capabilities of the model to achieve notable performance on clinical data, we implement a **per-label threshold optimization strategy** based on **Youden's J Index** (Formula 2.18.

We determine the optimal threshold for each label using the validation split by selecting the value that maximizes Youden's Index. This enables a balanced trade-off between sensitivity and specificity for each diagnostic label. While certain clinical applications may prioritize sensitivity to avoid missing critical findings, our use of Youden's Index serves as a principled, label-specific strategy to optimize overall classification performance across diverse conditions. Therefore, this thresholding approach helps align the decision boundary to the relevant clinical patterns identified in the learned embedding space.

$$\begin{gathered} J = \text{Sensitivity} + \text{Specificity} - 1 \\ \text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \end{gathered} \tag{2.18}$$

2.1.5.3 End-to-End Text Generation evaluation: clinical report quality

To evaluate the quality of the generated radiology reports following downstream fine-tuning, we adopt standard Natural Language Generation (NLG) metrics that are commonly used in both general and domain-specific text generation tasks. In particular, we report **BLEU** and **ROUGE-L**, each capturing different dimensions of similarity between the generated and ground-truth reports.

BLEU (Bilingual Evaluation Understudy) computes the geometric mean of modified n-gram precisions, penalized by a brevity penalty to discourage overly short generations. The BLEU-n score is defined as in Formula 2.19, where p_i is the modified precision for i-grams, w_i are uniform weights (e.g., $w_i = 1/n$), and BP is the brevity penalty.

$$\text{BLEU-n} = BP \times \exp\left(\sum_{i=1}^{n} w_i \log p_i\right) \tag{2.19}$$

ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) evaluates the longest common subsequence (LCS) between generated and reference sequences. The ROUGE-L score is computed as in Formula 2.20, where P_{LCS} and R_{LCS} denote LCS-based precision and recall, respectively, and β is a factor typically set to emphasize recall ($\beta = 1.2$ is common).

$$\text{ROUGE-L} = \frac{(1+\beta^2) \cdot P_{LCS} \cdot R_{LCS}}{R_{LCS} + \beta^2 \cdot P_{LCS}} \tag{2.20}$$

These metrics provide a well-rounded view of generation quality, covering both n-gram precision and information coverage to ensure coherent and accurate reports.

2.2 Experimental evaluation

Based on the evaluation strategy outlined previously, this section presents the experimental environment and the results of our proposed approach for radiology report generation, along with the associated auxiliary tasks. We begin by analyzing the datasets used for training and evaluation, including the applied preprocessing steps, followed by an analysis of the technical setup and implementation details. Finally, we report and interpret the results of our evaluation metrics across retrieval, classification, and report generation tasks, highlighting performance and limitations compared to relevant state-of-the-art methods.

2.2.1 Dataset

For thorough benchmarking and comprehensive training of our proposed approach, we utilize the **MIMIC-CXR-JPG** dataset [JLP+24], a widely-adopted, large-scale collection of chest X-ray images paired with expert-annotated radiology reports. Publicly available via PhysioNet [GAG+13], the dataset is part of the broader MIMIC database initiative, which provides de-identified, multimodal clinical data collected from diverse real-world healthcare environments, to support the development of intelligent algorithms for medical applications.

2.2.1.1 Structure and Partitioning

MIMIC-CXR-JPG is derived from the original **MIMIC-CXR** dataset [JPB+19], by converting high-resolution DICOM-format chest radiographs into compressed JPG images. The transformation significantly reduces storage requirements and computational overhead, therefore facilitating more accessible and efficient use in machine learning pipelines. In addition, a substantial part of the dataset is enriched with structured diagnostic labels automatically extracted from the corresponding radiology reports using the CheXpert labeling tool [IRK+19], offering an extra layer of supervision for training and evaluation in clinical tasks. To further enhance the relevance and clarity of textual data, we adopt the **CXR-PRO** variant of the reports [RCR22], in which references to prior studies have been removed. This refinement ensures that the model learns associations grounded solely in the current study, improving both interpretability and alignment between visual input and diagnostic descriptions.

In alignment with our methodology, which requires image-text-label triplets for contrastive pretraining with additional supervision, we extract a curated subset of the MIMIC dataset. Specifically, out of the total 377,110 imaging studies with associated radiology reports, we retain **253,993** examples. This filtered subset includes only those studies that have structured diagnostic CheXpert labels and include improved textual reports from the CXR-PRO collection.

We organize this subset into **training**, **validation**, and **test** splits to support both extensive training and evaluation. While the original dataset provides official train and test splits, with the test set manually curated to include a balanced amount of each medical findings, we introduce an additional **validation split** by partitioning the training set. Specifically, we allocate **15%** of the training samples to validation using a **multi-label stratified shuffle split** strategy, ensuring that the distribution of medical concepts is preserved, which is critical in multi-label clinical classification settings.

Split	No. samples
Train	213,429
Validation	37,664
Test	2,900

Table 2.2: Number of samples in each dataset split used for training, validation and evaluation.

Table 2.2 summarizes the number of samples per split. In addition, to highlight the inherent **class imbalance** in the dataset, we visualize the overall label distribution across all samples in Figure 2.9, and the per-plit distribution in Figure 2.8. This imbalance is especially visible for rarer conditions such as Fracture, which require robust training methodologies in order to be correctly captured.

2.2.1.2 Data Preprocessing

To prepare the dataset for model training, we apply several preprocessing and augmentation steps. First, uncertain labels (value -1) in the CheXpert annotations are excluded by masking them as 0, effectively treating them as negative in line

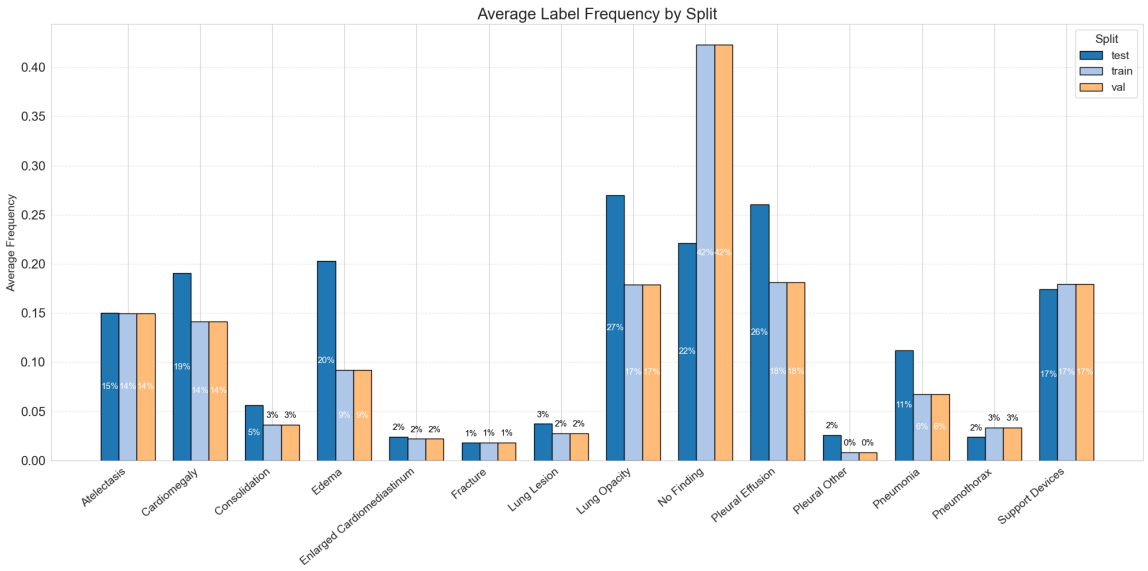


Figure 2.8: Average CheXpert label frequency across datasets splits, reflecting the adopted stratified multi-label splitting strategy.

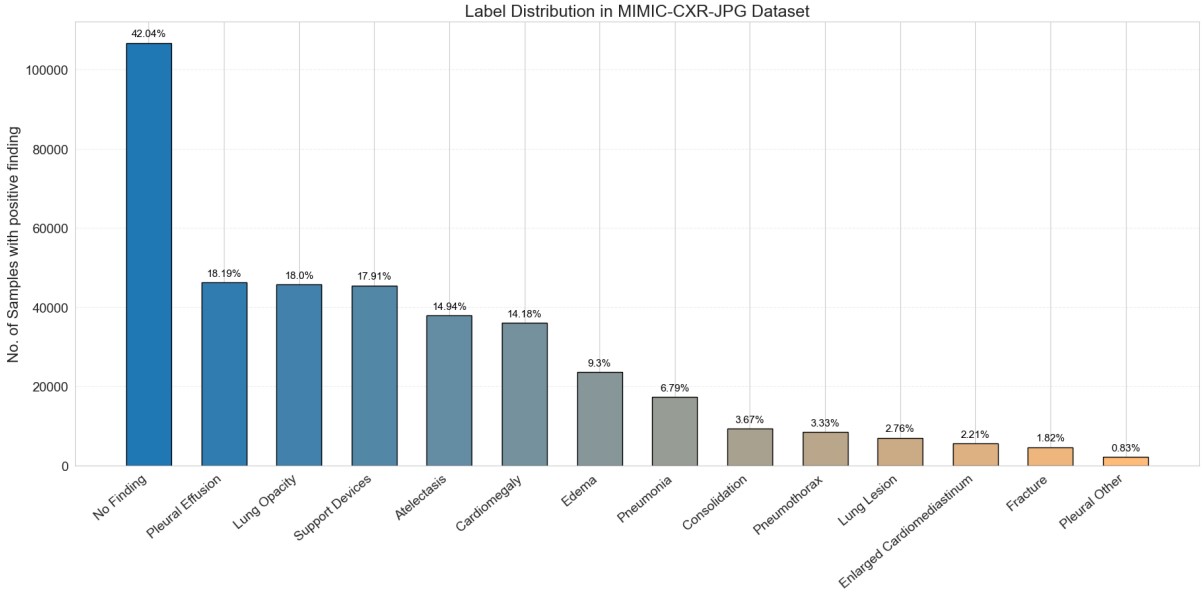


Figure 2.9: Class distribution of positive samples per CheXpert label in the MIMIC-CXR-JPG dataset. Common conditions such as Pleural Effusion appear more frequently, while critical findings like Fracture are less represented.

with prior work. All images are resized to a fixed resolution of **224x224** to match the input requirements of our Vision Transformer image encoder, while also optimizing memory usage.

Beyond these standard steps, additional augmentations are applied in context of our multi-view supervision strategy. More specifically, besides leveraging both frontal and lateral chest X-ray views from the same imaging study to improve diversity of visual inputs during pretraining, we also apply **CLAHE** (Contrast Limited Adaptive Histogram Equalization) to enhance local image contract, helping the model extract finer-grained visual features from radiographs. This is particularly beneficial for detecting subtle pathological patterns, which usually require experienced radiologists to confidently confirm their presence. The full image transformation pipeline is illustrated in Figure 2.10.

In parallel, we also apply a form of augmentation to textual data, known as **sentence swapping**. This technique involves randomly reordering or replacing sentences within the same report, so that the model would be able to capture relevant medical content instead of learning fixed sentence structures.

While these augmentation strategies are applied exclusively on the training split to ensure robustness and generalization, the validation and test splits only use minimal preprocessing, such as image resizing and normalization. This approach preserves real-world evaluation conditions by avoiding artificial alterations.

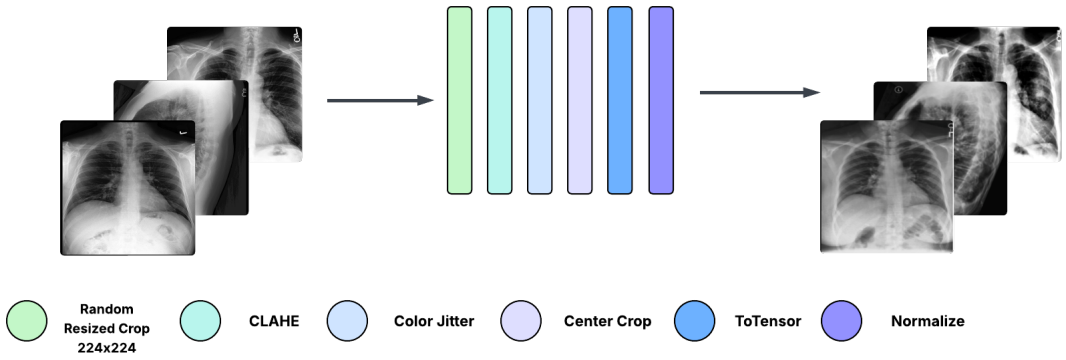


Figure 2.10: Image transformation sequence used for preprocessing training samples to improve generalization.

2.2.2 Experimental setup

All experiments were conducted within a modular project structure that enables independent loading of components for training or integration into a unified inference framework. The project is implemented as a Python package built on the **PyTorch** framework [PyT24], with core functionality extended through Hugging Face's `transformers` and `datasets` libraries. These libraries enable flexible loading of configurable architecture components, supporting our hybrid approach through the independent initialization of Transformer modules via dedicated factory functions.

Training was performed on a single NVIDIA Tesla A100 GPU with 40GB VRAM, using mixed-precision training through PyTorch's automatic gradient scaler to maximize computational efficiency and minimize memory consumption. Memory opti-

mization was further supported by efficient data loading routines that ensure only the active batch resides in memory during forward passes.

As outlined in the methodology, the model is trained in three distinct phases, each aligned with a specific objective and configuration: *Contrastive Pretraining* (CPT) for aligning multimodal representations, *Classification Fine-Tuning* (CLFT) for supervised label prediction, and *Report Generation* (RGEN) for synthesizing radiology reports. The following sections provide an overview of the training setup for each phase, along with the corresponding hyperparameter settings, compared and summarized in Table 2.3.

Hyperparameter	CPT	CLFT	RGEN	**Description**
Optimizer	Adam	Adam	Adam	Adaptive optimizer with weight decay for stable training.
Learning Rate	3×10^{-5}	1×10^{-4}	1×10^{-4}	Controls the step size during optimization.
Scheduler	Cosine	Cosine	Linear	Gradually reduces the learning rate to aid convergence.
Batch Size	64	32	16	Number of samples per batch. Lowered for report generation due to memory load.
Epochs	10	5	15	Number of training epochs. Varies by phase complexity.

Table 2.3: Comparison of hyperparameter settings for each training phase. Each configuration is adapted to the specific goals and constraints of the phase.

Contrastive Pretraining

The contrastive pretraining phase implies learning aligned representations of image and text modalities, which are projected in a **shared embedding space**. The dimensionality of this space is set to 768, a common choice inherited from standard transformer architectures which offer sufficient granularity for capturing multimodal features. The obtained aligned embeddings are scaled using a **temperature parameter** initialized with 0.07 and learned during training, following the CLIP baseline [RKH+21]. This value provides a sharp similarity distribution at initialization, improving discrimination among hard negatives.

Classification Fine-Tuning

An additional auxiliary experiment is conducted to comprehensively evaluate the classification capabilities of the pretrained CLIP-based model, by fine-tuning a lightweight classification head on top of the image encoder to predict the 14 CheXpert findings. Given the nature of this task and the limited complexity of the clas-

sifier, a reduced number of training epochs is employed to prevent overfitting and ensure efficient convergence.

Downstream Report Generation

The downstream task adaptation of the pretrained model is performed through the fine-tuning of a BERT-based text decoder, extended with generation-enabling modifications. The image encoder remains frozen and is integrated into the architecture, together with a custom prompt constructor module, dynamically configured based on the chosen prompt conditioning strategy. When zero-shot classification is employed for prompt generation, the constructor utilizes both the pretrained image encoder and the associated text encoder. In contrast, for supervised prompt conditioning, only the pretrained classification head is additionally loaded to provide CheXpert label predictions.

The constructed prompt, derived from the predicted findings, is prepended to the decoder's input sequence to guide generation toward medically relevant content during training. The input is complemented by projected image embeddings, adapted to match the dimensionality of the decoder's Cross-Attention layers, therefore enriching the generation context. For decoding, **Beam Search** is used with a beam width of 3, meaning that at each decoding step, the model keeps the top-3 most likely partial sequences, expanding only the most promising candidates to improve fluency and relevance compared to greedy decoding.

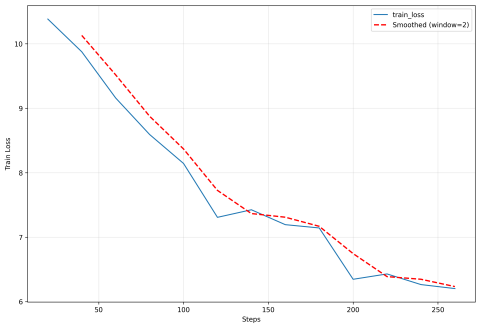
2.2.3 Results and discussion

The following section outlines the experimental results of the proposed method, with a focus on analyzing the performance of each auxiliary task that contributes to the overall report generation pipeline. A comprehensive comparison with related work is conducted, supported by both quantitative metrics and visualizations, which are further analyzed in subsequent discussions. All reported results are computed on the test split of the dataset using the selected evaluation metrics. In addition, to better understand the learning behavior of the model, Figure 2.11 illustrates the progression of loss and key performance metrics over epochs, demonstrating stable convergence and consistent performance improvement over time.

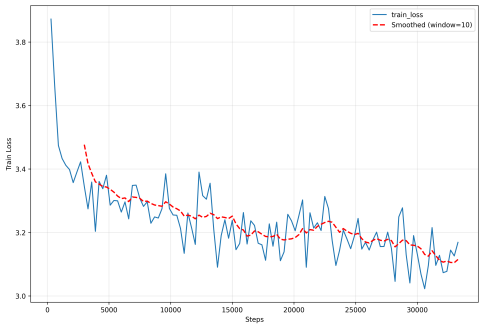
2.2.3.1 Image-Text Retrieval

The Image-Text retrieval task evaluates the alignment between visual and textual representations learned during the contrastive pretraining phase. We compare our CLIP-based configuration with domain-adapted contrastive methods, and baseline general-purpose architectures fine-tuned on the same medical dataset. The goal is to assess whether our proposed **Multi-view Concept Similarity Loss** (MCSL) leads to improved retrieval performance compared to alternative objectives.

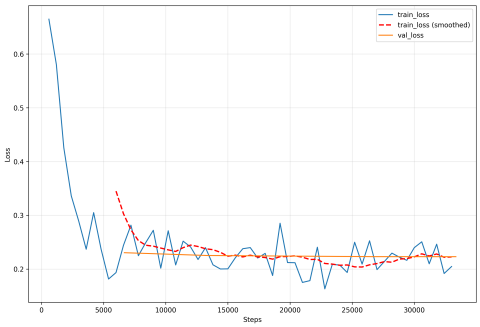
As shown in Table 2.4, CLIP-XRGen achieves a top-1 Recall of 3.8% for exact image-text retrieval, where the goal is to identify the correct ground-truth report as the first most similar sample for a given image. While this performance is significantly lower than that of CXR-CLIP [YGH+23], it still outperforms MedCLIP [WWAS22], which achieves only 1.1%. In the class-based retrieval setting where



(a) Generation loss curve.



(b) Pretraining loss curve.



(c) Classification loss curve.

Figure 2.11: Training loss curves for the CLIP-XRGen model.

Precision@K is measured, our model reaches a top-1 score of 48.3%, a competitive result approaching the values reported by MedCLIP. However, it is important to note that MedCLIP's evaluation is conducted in a simplified single-label context using the CheXpert5x200 subset, focusing only on five medical conditions. In that setup, each query image and a target text correspond to a single concept. In contrast, our evaluation uses a **multi-label setup** based on MIMIC-CXR data and includes all 14 CheXpert labels, considering one retrieval as successful if label overlap exists between the found sample and the query chest X-ray, i.e., they have at least one shared finding. Although this configuration is more realistic and clinically relevant, the results are less directly comparable and should be interpreted accordingly.

Nonetheless, these results highlight the trade-offs inherent in different contrastive learning strategies applied to the medical domain. MedCLIP's lower Recall@k can be attributed to its fully decoupled learning objective, which relies solely on label correlation without requiring explicit alignment between image and text pairs. This lack of pairwise supervision appears to limit its ability to retrieve exact ground-truth matches. CLIP-XRGen, by contrast, employs **MCSL**, which balances semantic similarity with pairwise matching. While this leads to lower performance than CXR-CLIP on exact retrieval, it enables notable performance in class-based retrieval and demonstrates **generalization to semantically related samples**.

Overall, CLIP-XRGen shows promise in overcoming the limitations of existing contrastive learning approaches for learning joint image-text representation adapted to medical data, by effectively balancing semantic alignment and pairwise consistency. However, further improvements are needed to enhance its applicability in real-world scenarios. Future work may explore training on larger, high-quality datasets in fully decoupled setups with automated label extraction, as well as in-

corporating structured medical knowledge to strengthen semantic understanding and support broader vision-language applications.

Model	Recall@K			Precision@K		
	@1	@5	@10	@1	@5	@10
CXR-CLIP	21.6	48.9	60.2	-	-	-
MedCLIP	1.1	1.4	5.5	45.0	48.0	50.0
CLIP-XRGen (ours)	**3.8**	**16.2**	**25.9**	**48.3**	**47.4**	**46.5**

Table 2.4: Retrieval performance comparison with related work in both exact image-text retrieval using Recall@K and class-based retrieval using Precision@K.

Comparison to baseline architectures

For a more in-depth analysis of our proposed contrastive strategy, we additionally compare CLIP-XRGen with baseline Vision-Language Models (VLMs) originally designed for general-purpose tasks, specifically CLIP [RKH+21] and BLIP [LLXH22], both of which we fine-tune on the MIMIC-CXR dataset. These models serve as reference points for evaluating the impact of our domain-adapted contrastive objective.

Model (MIMIC-CXR)	Recall@1	Recall@5	Recall@10
CLIP	0.1	0.3	1.2
BLIP	0.8	4.1	6.8
Ours (CLIP-XRGen)	**3.8**	**16.2**	**25.9**

Table 2.5: Comparison of image-text retrieval performance with baseline contrastive models. All models are fine-tuned on MIMIC-CXR.

Therefore, as presented in Table 2.5, our model demonstrates a significant performance improvement in exact image-text retrieval compared to both CLIP and BLIP under the same evaluation conditions. This performance gap further highlights the limitations of applying general-purpose VLMs that are trained for direct multimodal alignment, to the medical domain, and confirms the effectiveness of a concept-level alignment objective such as MCSL, which leverages more specialized supervision for ensuring better discrimination of medical semantics within the learned multimodal embeddings.

To visually support this, Figure 2.12 presents a t-SNE projection of the multimodal embedding space in reduced dimensionality, revealing how medical concept alignment guides the clustering of semantically related samples to a given extent, noting that the variability of medical cases and the frequent co-occurrence of multiple findings limit a clear separation between clusters.

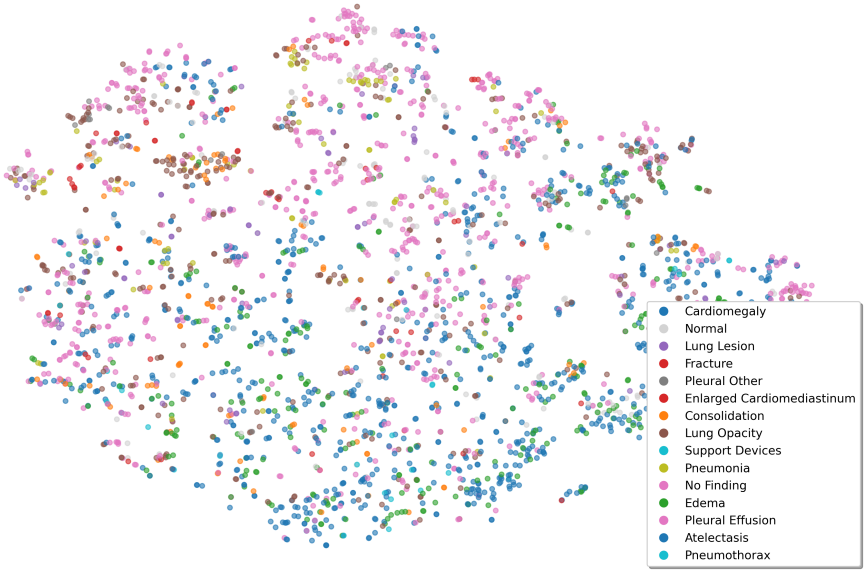


Figure 2.12: t-SNE visualization plot of image embeddings from CLIP-XRGen, colored by CheXpert pathology labels.

2.2.3.2 Medical Image Classification

Medical image classification serves as the backbone of the report generation pipeline by enabling the construction of **condition-aware contextual queries** from predicted findings. As shown in Table 2.6, we evaluate both prompt-based scoring and supervised fine-tuning on full label space, using the same test split as in image-text retrieval. Class-specific optimal thresholds are selected via Youden's index for robust evaluation. Unlike related work that relies on a restricted CheXpert5x200 subset, our setup enables a more comprehensive assessment, though at the cost of direct comparability.

Model	Zero-Shot ACC	Supervised ACC
MedCLIP	50.2	56.5
CXR-CLIP	62.8	65.7
Ours (CLIP-XRGen)	**66.7**	**72.7**

Table 2.6: Comparison of concept classification accuracy under zero-shot and supervised fine-tuning settings.

Zero-Shot Classification. In the zero-shot setting, the pretrained encoders achieve an average classification accuracy of 66.7%, outperforming both MedCLIP (50.2%) and CXR-CLIP (62.8%). This highlights the **strong generalization capability** of our model in identifying clinical findings **without any task-specific supervision**.

Supervised Fine-Tuned Classification. When fine-tuned on labeled examples using a classification head, CLIP-XRGen demonstrates a notable performance gain, reaching an average accuracy of 72.7%. This improvement confirms that the model

benefits from **task-specific supervision** in distinguishing medical findings. In this setting, MedCLIP reaches 56.5% with a similar approach of freezing the image encoder and training a classification head, while CXR-CLIP reports 65.7% accuracy score in the context of providing supervision to the pretrained encoders and using prompt-based learning for boosting the initial zero-shot performance. To gain deeper insight into class-wise performance, we also provide ROC curves for each CheXpert label in Figure 2.13. These visualizations confirm strong performance on well-defined findings such as Cardiomegaly, Edema, and Pleural Effusion (AUC $>$ 85.0%), while highlighting lower performance on more ambiguous or rare findings like Fracture. The determined optimal thresholds reflect varying difficulty and decision boundaries across labels.

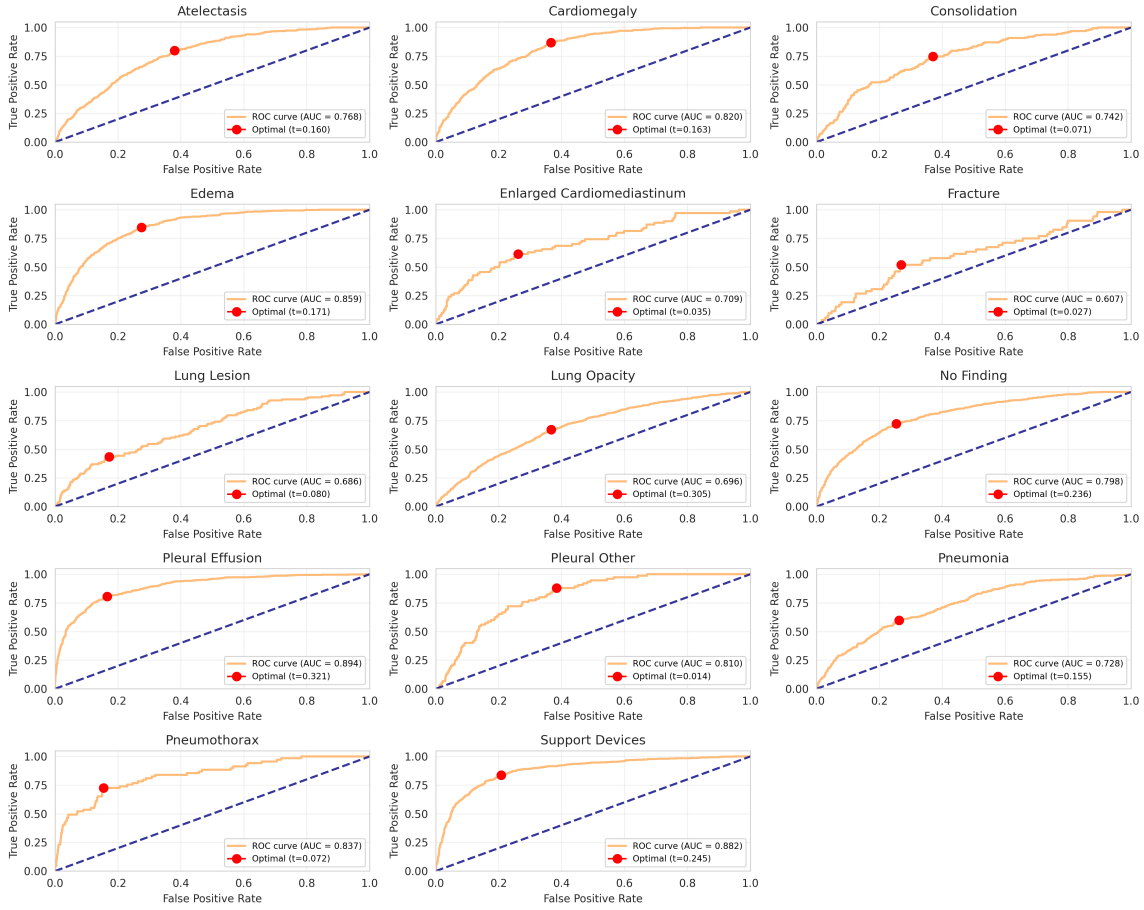


Figure 2.13: ROC curves for CheXpert label classification with CLIP-XRGen. Each subplot shows the AUC and optimal threshold (red dot) for a specific pathology.

The classification results reveal several key trends. First, despite the differences in evaluation protocols, CLIP-XRGen consistently outperforms prior models in both zero-shot and fine-tuned settings, confirming the effectiveness of the used concept similarity for obtaining a **more semantically rich shared embedding space**.

Second, the substantial improvement achieved through fine-tuning (from 66.7% to 72.7% accuracy) underscores the importance of supervision in ensuring medical domain performance. Unlike general-purpose datasets, chest X-rays often contain subtle pathologies, which benefit from explicit guidance during training. The ROC curves further illustrate that model performance varies significantly across labels,

reflecting the **inherent difficulty of certain findings** and the impact of class imbalance.

Overall, these results validate the design of the classification module within our pipeline and support the usage of a contrastive method in multimodal clinical settings. The enhanced obtained discriminative capability of the supervised configuration provides a promising foundation for accurate prompt generation for the downstream report generation task. Moreover, the effectiveness of this classification-guided prompting can be further influenced by **prompt engineering strategies**, as different template formulations may significantly affect the quality and specificity of the generated text.

2.2.3.3 Radiology Report Generation

As the central focus of this thesis, the radiology report generation task aims to assess the effectiveness of our proposed hybrid approach, which leverages the contrastive pretrained backbone for downstream adaptation on conditional language generation. We report NLG metrics to compare CLIP-XRGen against prior models specifically designed for clinical report generation.

Table 2.7 shows that CLIP-XRGen **underperforms on standard generation metrics** compared to R2Gen [CSCW20] and CXR-RePaiR [EKK+21]. For the latter, we reference both the CXR-RePaiR-2 variant, which retrieves the top-2 most similar sentences from the computed sentence set, and CXR-RePaiR-Select, which selects sentences based on clinical relevance to construct the report. Our approach achieves a BLEU-2 score of only 0.020 and a ROUGE-L score of 0.124, comparable to the retrieval-based scores of CXR-RePaiR (0.069 and 0.050 for BLEU-2, respectively), but substantially lower than state-of-the-art performance of R2Gen, which reports a BLEU-2 of 0.218 and a ROUGE-L of 0.277.

Model	**BLEU**				**ROUGE-L**
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	
R2Gen	0.353	0.218	0.145	0.103	0.277
CXR-RePaiR-2	–	0.069	–	–	–
CXR-RePaiR-Select	–	0.050	–	–	–
Ours (CLIP-XRGen)	**0.018**	**0.020**	**0.004**	**0.001**	**0.124**

Table 2.7: Comparison of radiology report generation performance against related work using BLEU@K and ROUGE-L. CXR-RePaiR [EKK+21] reports only BLEU-2.

To further analyze generation performance beyond numerical metrics, Figure 2.14 illustrates a sample prediction made by our model, compared with the ground-truth report and findings. While the output does not represent a fully coherent and accurate description, it highlights that **concept-guided prompting can enforce topic relevance**. The model successfully identifies the medical conditions present in the study, but fails to localize them accurately or provide further detailed clinical context, limitations largely originating from the restricted scope of the leveraged concept labels.

This performance gap reveals key limitations in the current architecture. Despite promising performance on auxiliary tasks such as classification and retrieval, the

decoder component struggles to effectively use pretrained encoders during generation. A likely cause is the use of BERT as a decoder backbone, which lacks autoregressive capabilities unless explicitly adapted, therefore limiting the possible fluency and structure. Furthermore, clinical report generation demands handling of complex medical vocabulary, uncertainty and severity, subtleties that are not well captured by the current encoder-decoder setup. As a result, generated reports may appear vague or diagnostically unreliable.

Overall, while our approach shows notable image-text alignment, the generation module lacks the expressivity and fluency needed for clinical-grade reporting. Future work should explore integrating more advanced generative modes, such as **instruction-tuned LLMs**, to better handle structured, semantically rich context extracted from visual inputs.

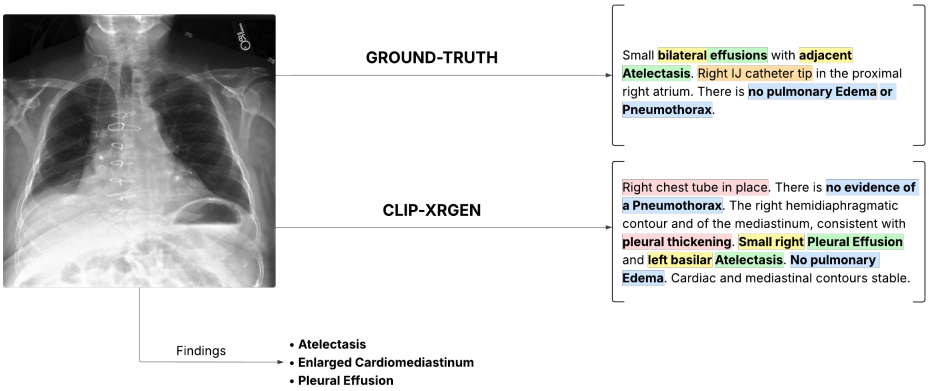


Figure 2.14: Comparison of model prediction and ground-truth report for a sample chest X-ray. Highlight colors indicate semantic categories: green for correctly identified findings, yellow for localization mismatches, blue for correctly negated conditions, red for hallucinated content, and orange for clinically relevant context omitted by the model.

2.3 Conclusions and future work

This chapter introduced **CLIP-XRGen**, a new approach for radiology report generation that extends contrastive-based VLMs for the medical domain. The proposed modular architecture demonstrated effectiveness in a multi-task setting, repurposing pretrained multimodal alignment capabilities for downstream classification and conditional text generation. Through the contributions presented in Section 2.1, including a custom contrastive learning objective (MCSL) and a prompt-guided decoding mechanism, our model shows promise toward intelligent clinical assistants capable of aiding expert radiologists, as further discussed in Section 2.2.3.

The conducted experiments addressed the research questions **RQ1** and **RQ2** defined at the beginning of this chapter. First, we have shown that the **MCSL objective**, which integrates multi-view supervision and medical concept alignment balanced with exact pairwise matching, effectively enhances the semantic structure of the learned shared embedding space. Although exact retrieval performance may be slightly inferior to that of traditional InfoNCE-based objectives, evaluation in multi-label learning environments with high variability in medical cases shows that our method successfully captures clinically relevant concepts. These embeddings,

when leveraged in classification tasks, validate the improved semantic alignment of the contrastive method, which is particularly beneficial for downstream modules such as prompt construction, where concept-level fidelity is crucial.

However, in addressing the second research question, while **prompt-guided decoding** does improve topic relevance by focusing generation around the predicted findings, the decoder struggles to capture the full depth of medical semantics, especially in expressing positional information and severity or uncertainty levels. These shortcomings originate from both the encoder's limited capacity to represent such fine-grained features and the architectural constraints of the adopted module itself. Consequently, the generated reports lack the coherence and precision required for deployment in clinical settings.

Several promising directions for future work can be explored to improve both core components of this research. Foe the *contrastive pretraining* stage, future efforts could investigate hybrid training strategies that leverage both paired datasets and decoupled image-text collections as seen in MedCLIP [WWAS22]. In this way, a better balance may be achieved between retrieval and concept understanding, exposing the model to a greater diversity of training scenarios. Another valuable extension would involve enriching the label supervision scheme with additional contextual indicators, such as positional information and uncertainty or severity levels, which could help the model capture more complex medical semantics. While this would increase the complexity of the classification task to to the expanded label space, it holds potential for improving downstream interpretability. Moreover, incorporating structured medical knowledge graphs [HLL+23] could further enhance semantic grounding and strengthen the model's ability to interpret medical vocabulary within the text modality.

In terms of *report generation*, integrating more powerful generative language models such as GPT-style architectures [RNSS18], particularly those designed to handle additional conditioning as seen in RepsNet [TBF22], could significantly enhance fluency and factuality of generated reports. In addition, more effectively bridging the gap between contrastive and generative modules remains a promising direction. One example being the use of a query-based intermediary, as in BLIP-2 [LLSH23], which enables visual-textual interaction while keeping both encoder and decoder largely frozen. This lightweight adaptation strategy could facilitate efficient training while maintaining flexibility for the development of more clinically reliable automated radiology reporting systems.

Another important direction involves addressing the limitations of standard NLG evaluation metrics that are unable to effectively offer insight into the clinical accuracy or diagnostic value of generated text sequences, as they primarily assess the surface-level structure and n-gram overlap. Therefore, future research should prioritize the development and broader adoption of specialized metrics that better reflect medical relevance and enable fair comparisons.

Chapter 3

`VistaScan`: A web application for remote radiology consultation

This chapter presents **VistaScan**, a web-based remote consultation platform designed to enable radiology experts to evaluate medical imaging studies online. The platform facilitates the delivery of diagnostic assessments and treatment plans without requiring in-person appointment. Based on the expert's interpretation of the submitted study, they can recommend further investigations or advise the patient to visit a medical facility promptly for intervention, if necessary.

In order to support the diagnostic process, experts can leverage the integration of the **CLIP-XRGen** assistant model, which can generate preliminary reports on demand, outlining initial medical findings. This AI-generated draft assists experts by highlighting potential areas of concern, helping them focus their investigation more effectively. As a result, diagnostic accuracy improves, the review process becomes more efficient, and practitioners gain greater flexibility in managing their workload. Patients can register an account, upload their imaging studies, and receive expert evaluations in a timely manner.

VistaScan is built using a modern technology stack that emphasizes scalability, maintainability and responsiveness. The backend is implemented using FastAPI, which powers the application's API and provides an interface for the model integration. The frontend is built using React, delivering a dynamic and intuitive user experience. Authentication and authorization are implemented using stateless JWT-based tokens for secure access control, while MongoDB handles structured data storage and an S3-compatible object storage service provides efficient large file management.

The development process followed the standard software engineering life cycle, thoroughly documented in the sections that follow. The chapter is structured as follows: Section 3.1 defines the functional and non-functional requirements of the application, followed by the architectural and design decisions outlined in Section 3.2. Section 3.3 details the implementation process, followed by the presentation of a user manual (Section 3.4) to guide the interaction with the system. Finally, potential future enhancements are discussed in Section 3.5, aimed at increasing the system's flexibility and clinical reliability.

3.1 Requirements Engineering

Building an effective software product involves more than just its technical implementation. It begins by understanding the goals of the project, identifying a target audience, and recognizing the specific problems the system is intended to address. Based on this, key functionalities are defined to shape the user experience and meet the expectations of all potential users. For VistaScan, this process of requirements engineering consisted in translating the intended behavior of a remote consultation platform, along with the workflows of radiology experts, into a coherent set of features that the platform could deliver.

Therefore, based on the identified actors, VistaScan employs a role-based access model that supports three distinct categories of users, each interacting with the system through distinct flows specific to their responsibilities and needs:

- **Patients:** Can create an account or be assigned one, submit consultation requests by uploading medical imaging studies (e.g., chest X-rays), and access diagnostic reports and recommendations once reviewed by a medical expert.
- **Experts:** Are assigned to accounts with access to a dashboard for reviewing pending consultations, optionally use the integrated AI model to generate preliminary reports, annotate submitted images, and provide final diagnostic assessments.
- **Admins:** Manage account creation, role permissions, and monitor consultations activity. They also have access to all functionalities available to other user roles.

These interaction flows can be further formalized through a series of use cases, each describing a specific scenario derived from the functional requirements. A visual representation of these use cases is provided in the use case diagram shown in Figure 3.1.

User Registration and Authentication. Patients can register for an account, while experts are assigned accounts created by admin users. All users can securely log into the platform. This functionality enables personalized access and linking submitted studies and diagnostic reviews to user identities.

Patient Dashboard and Consultation Overview. Patients have access to a personal dashboard where they can view their consultation history, read diagnostic reports, and submit new cases. This feature empowers users to manage their own medical consultations independently and access evaluations remotely.

Consultation Submission. Patients can initiate a new consultation by uploading imaging studies. This submission triggers the diagnostic overflow, notifying experts of a new case awaiting review.

Expert Dashboard and Consultation Review. Radiology experts are presented with a dashboard displaying all pending and completed consultations. They can claim cases, view submitted imaging studies, and manage their workload efficiently.

Annotation and Findings Documentation. After claiming a consultation, experts can analyze the imaging study, document their clinical findings, and record a final diagnostic report. This step formalizes the outcome of the consultation for the patient.

AI-Assisted Diagnosis. Experts can optionally use the integrated CLIP-XRGen assistant model to generate a preliminary report. This functionality supports decision-making by highlighting potential findings and helping ensure comprehensive evaluation.

Admin Platform Management. Admin users manage platform-wise operations, including creating and modifying user accounts, assigning roles, and overseeing the consultation process. This provides essential administrative control and ensures the platform runs reliably and securely.

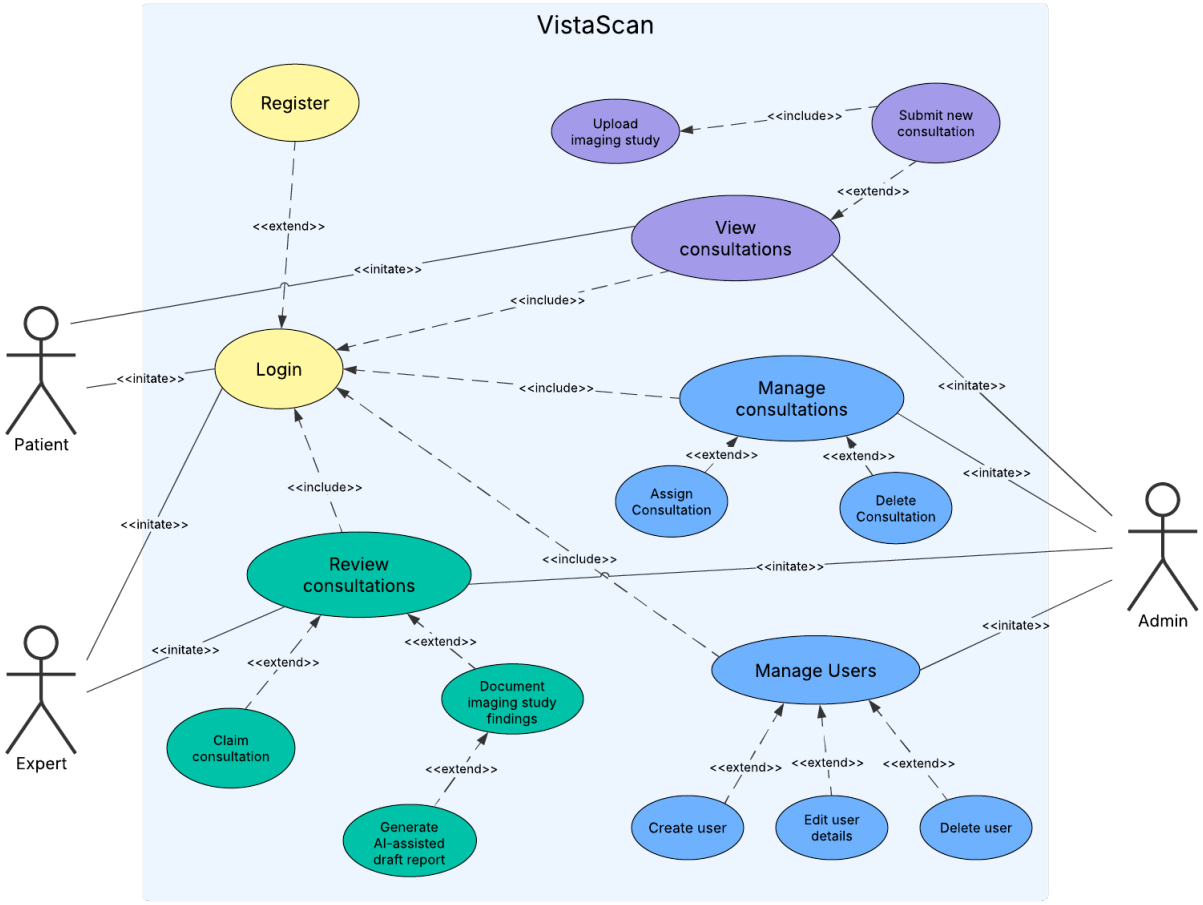


Figure 3.1: Use case diagram of VistaScan, highlighting user roles and system interactions.

3.2 System Design

In the system design phase, we transition from a conceptual understanding of system requirements to concrete architectural decisions. We focus on defining the

structure and interaction of the application's main components in order to satisfy the identified functionalities. A high-level architectural overview is illustrated in Figure 3.2, and each design choice is detailed in the following subsections.

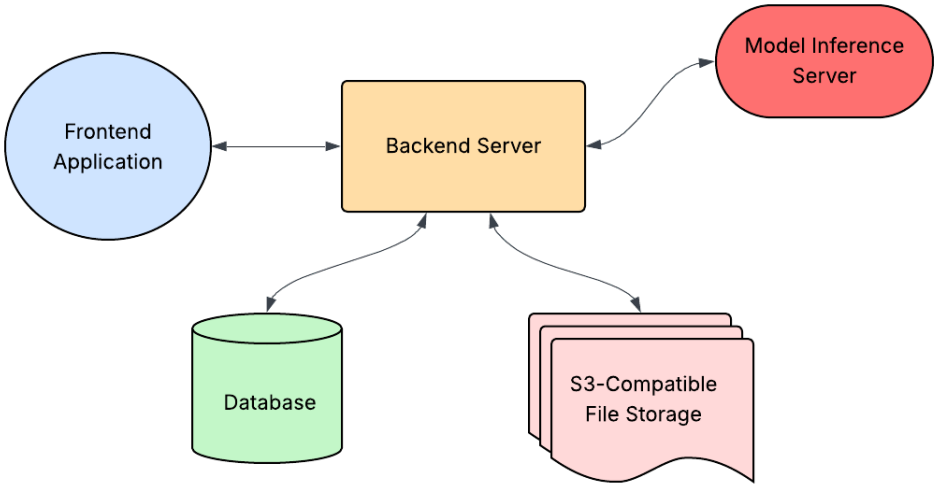


Figure 3.2: High-level overview of the application architecture.

3.2.1 Architectural Approach

Based on the expected scale of the platform and anticipated usage load, VistaScan adopts a **client-server** architectural model, organized around two primary components.

The **web-based frontend** is responsible for handling all user-facing interactions, rendering dynamic content, adapted to each user's role. On the other side, the **API-driven backend** encapsulates the core application logic, including user management, consultation workflow coordination, and integration with external services such as the report generation model.

The overall architecture promotes a loosely coupled interaction between components, allowing each module to operate independently while contributing to a cohesive application experience.

3.2.2 Data Management

An essential aspect of the system design phase is establishing an effective data management strategy. Due to the diverse data requirements of a remote radiology consultation platform, the application uses two specialized storage solutions, one for structured data and another for unstructured large medical records, both of which are detailed in this section.

3.2.2.1 Database

To store and manage core domain entities, VisaScan uses **MongoDB** [Mon24], a NoSQL database framework built around the concept of documented-oriented storage. Unlike traditional relational databases that rely on fixed schemas and tabular structures, MongoDB supports unstructured and semi-structured data formats,

making it ideal for applications that require adaptability in data modeling. Therefore, this flexibility aligns well with the clinical domain, where data types can vary in structure and may need additional fields over time.

In the context of our application, this flexible schema design allows the system to expand the support for more complex imaging formats used in healthcare, such as `DICOM` (.dcm) or `NIfTI` (.nii), which necessitate saving additional metadata alongside the file reference. Similarly, to improve explainability in AI-assisted workflows, the schema currently adopted for communicating with the integrated model can be extended to include prediction confidence scores of identified medical concepts or potential region-of-interest annotations, without requiring costly migrations.

The system defines two main document collections, as illustrated in Figure 3.3:

- **Users**: Contains authentication details, role assignments, and user profile information. Each user is uniquely identified by a UUID and can be linked to multiple consultation records via the `patient_id` or `expert_id` fields, based on the use case or his permissions.

- **Consultations**: Stores metadata for each submitted case, including status indicator fields, timestamps for tracking the consultation's progress, and references to the associated imaging studies and diagnostic reports. Similarly identified by a UUID field.

A logical association between users and consultations is maintained via the user identifier field, enabling SQL-like query operations such as retrieving all consultations submitted by a specific patient or under review by a given expert. In addition, the system leverages **embedded documents** within the Consultations collection to store the imaging study metadata and expert report. This hierarchical structure improves read performance by localizing all relevant consultation data in a single document, eliminating the need for complex joins or cross-collection queries.

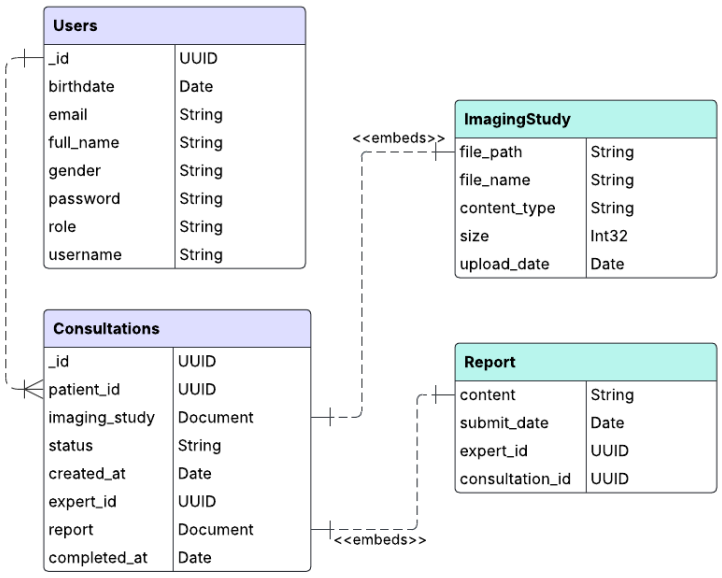


Figure 3.3: Diagram of the Database Schema.

3.2.2.2 Object Storage

To efficiently handle high-resolution medical images and optimize performance for read operations implied by the radiology expert workflow, VistaScan separates the storage of large binary files from operational data stored in the database. For this purpose, the application integrates **MinIO** [Min24], an open-source, S3-compatible object storage solution designed for high-performance and scalable file handling.

MinIO provides several advantages that are particularly aligned with the requirements of a remote radiology consultation platform. It is optimized for demanding storage workloads, it supports horizontal scalability, and integrates seamlessly with the backend through the use of the industry-standard **S3 API specification**, exposing CRUD operations through straightforward HTTP requests.

In terms of data organization, the imaging studies in currently supported PNG and JPEG formats, are stored as individual objects in a dedicated **storage bucket**, following a structured naming convention: `{user_id}/{uuid}-{original_filename}`. This format, meant to facilitate the traceability of objects by grouping them based on the owner user identifier, represents the file path that is persisted within an embedded document in the database consultation entity, alongside additional lightweight metadata (original file name, content type, size and upload timestamp).

When users or experts need access to imaging studies, the system generates **time-limited presigned URLs**, which provide direct, authenticated access to MinIO without exposing storage credentials to the frontend. This allows secure file handling with automatic URL expiration and enables direct browser-to-storage transfers, reducing backend load and improving overall system responsiveness.

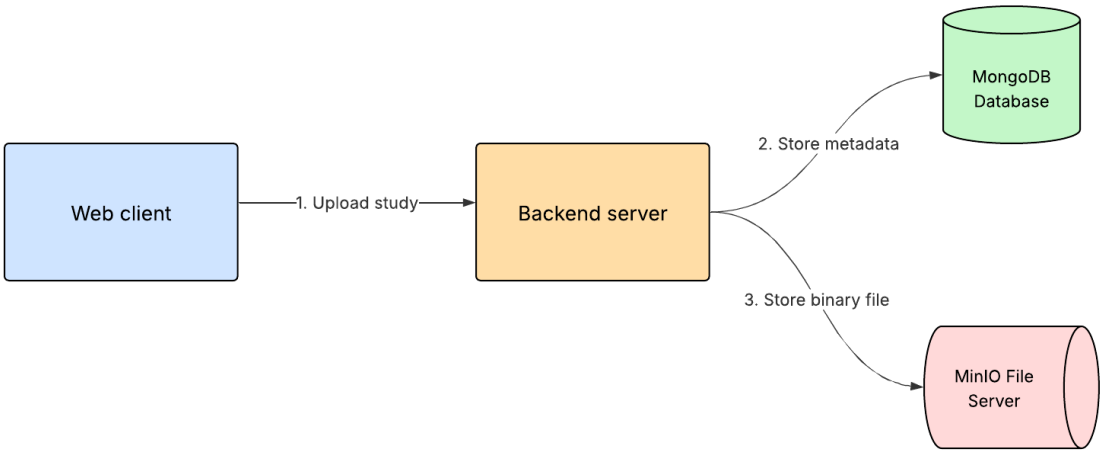


Figure 3.4: Data flow diagram for uploading an imaging study.

3.2.3 Deployment Strategy

The goal of the deployment process is to define how the application is packaged, configured and orchestrated in an environment accessible to end users. VistaScan uses a **containerized multi-service deployment architecture** built with **Docker** and Docker Compose. Each core system component illustrated in Figure 3.2 is encapsulated in its own container, with dependencies managed via technology-specific

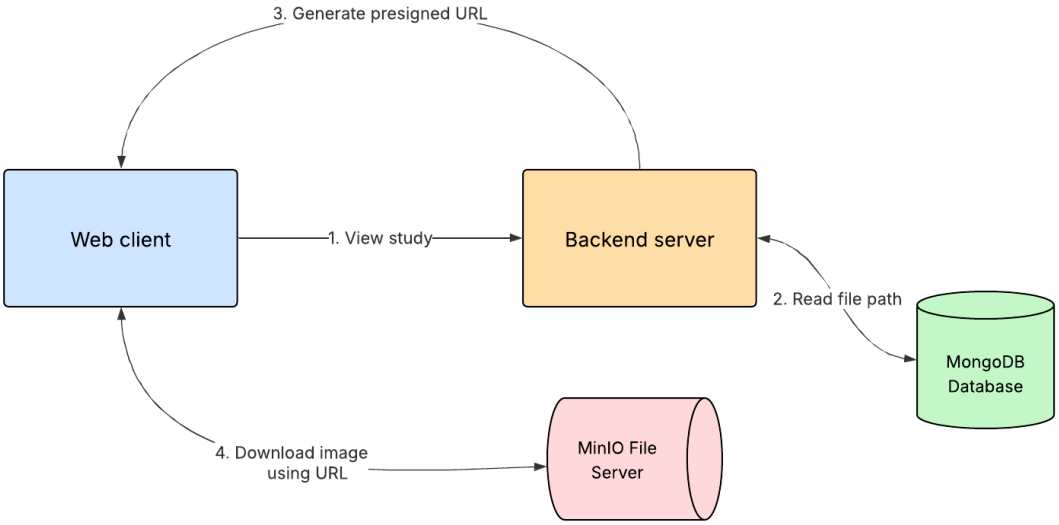


Figure 3.5: Data flow diagram for viewing an imaging study.

package managers. These containers communicate through a dedicated Docker bridge network, ensuring secure and optimized inter-service communication.

With this setup, the application can be deployed on any infrastructure that supports Docker, including cloud providers and local servers. For production-ready deployments, platforms such as **DigitalOcean** are well-suited, offering lightweight virtual machines known as **Droplets** that can host the entire container stack. When deployed to a cloud environment like DigitalOcean, the application can be easily exposed via a custom domain name, making it publicly accessible over the internet.

The entry point for the application is configured using **Nginx**, a high-performance HTTP server that acts as both a static file server and a reverse proxy. In this setup, Nginx serves the web client by delivering the SPA application index file and handling client-side navigation requests, while simultaneously forwarding all `/api/*` requests to the backend container on its exposed internal port. This **reverse proxy** configuration provides multiple benefits: it eliminates Cross-Origin Resource Sharing (CORS) issues by ensuring that frontend and backend traffic originates from the same domain, and enhances security by keeping the backend isolated from direct external access.

3.3 Implementation

Building upon the previously defined system architecture, the implementation phase focuses on converting the design into actual source code that defines the final accessible software product. This section details the development of each module using the adopted technology stack, highlights key code-level decisions, and demonstrates how the concept is realized through concrete technological solutions.

3.3.1 Backend

The backend of a web application represents the component responsible for managing the core logic, data persistence, and communication between internal and

external services. It acts as a bridge between the frontend user interface and the underlying infrastructure, handling business rules to ensure the system behaves as intended.

The implementation of VistaScan's backend module is facilitated by the **FastAPI** framework [Ram24], a modern, high-performance Python web framework. FastAPI was chosen for its asynchronous request handling, automatic data validation using Pydantic models, and built-in support for generating interactive API documentation. Therefore, these features significantly reduce boilerplate code while enhancing security, reliability, and developer productivity.

3.3.1.1 Clean Architecture

To further support the idea of modularity and long-term scalability, the backend is structured around the principles of **Clean Architecture**, a widely adopted software design approach introduced by Robert C. Martin in his 2012 article and later formalized in his book *Clean Architecture: A Craftsman's Guide to Software Structure and Design* [Mar17]. While Clean Architecture has become the most recognizable name, it is part of a broader family of architectural patterns that share the same core philosophy: isolating business logic from external implementation details.

These principles were first introduced in **Hexagonal Architecture** (also known as Ports & Adapters) by Alistair Cockburn in 2005 [Coc05], later reinterpreted in **Onion Architecture** by Jeffrey Palermo in 2008 [Pal08], and finally consolidated under the Clean Architecture model. Despite their differences in structure and terminology, all three emphasize **separation of concerns** and **inversion of dependencies** in order to keep the core business logic completely agnostic of frameworks, databases, or external tools necessary for implementation. This system design choice allows applications to be more maintainable, testable and resilient to changes in technology over time.

In designing VistaScan, the layering strategy proposed by Onion Architecture was selected for its ability to maintain balance between structure and simplicity, given the medium-scale nature of the application and the need to iterate quickly during development.

The backend is divided into the following four key layers, illustrated in Figure 3.6:

- **Domain Layer:** Encapsulates the core entities and business rules of the application. In the context of our app, this includes abstractions such as users, imaging studies, reports, and consultations. This layer has no dependencies on external libraries or frameworks, making it highly reusable.

- **Application Layer:** Defines the use cases of the system, orchestrating interaction between the domain and the outside world. It expresses system behavior in terms of domain abstractions and interacts only with interfaces instead of concrete implementations. This decoupling allows technologies to be swapped or updated without touching the inner workings of the backend. Typical operations here include user authentication workflows, initiating a consultation, or model-assisted imaging study review.

- **Infrastructure Layer:** Provides concrete implementations for the interfaces defined in the application layer. This includes communicating with databases

(e.g., MongoDB), file storage (e.g., MinIO), and external services (e.g., the model server). By strictly adhering to the business logic contract, this layer maintains independence and flexibility in how technologies are integrated.

- **Presentation Layer:** Serves as the delivery mechanism of the application, typically framework-specific route definitions, request validation and response formatting, This is the entry point of the backend system, which interprets incoming client actions, delegates them to the appropriate application services, and returns structured responses, following conventions of the adopted communication model.

This architecture ensures that the application logic can evolve independently of the technology stack, reducing the risk of vendor lock-in and simplifying quality assurance processes. For example, the system can be adapted to use a different database or file storage provider without rewriting any of the domain logic.

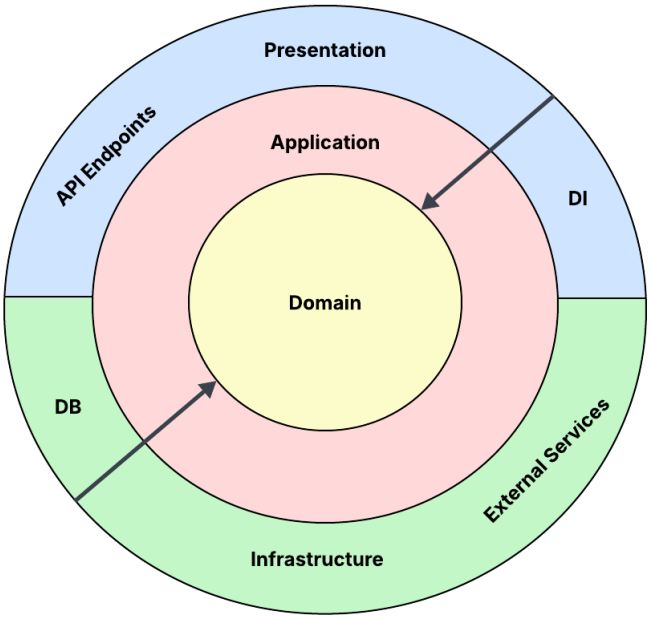


Figure 3.6: Clean architecture model adopted in the backend design.

3.3.1.2 RESTful API

The backend exposes its functionality through a **RESTful** API. **REpresentational State Transfer** (REST) is a widely adopted architectural style for designing networked applications, particularly web services, that use the HTTP protocol. While it is not a formal standard itself, it defines a set of guiding principles that promote resource-oriented communication using standard HTTP methods such as `GET`, `POST`, `PUT`, and `DELETE`.

In the context of VistaScan, REST was chosen for the Presentation layer due to its simplicity and stateless nature, meaning that each client request contains all the information needed for the server to process it, without relying on prior interactions. This design enables seamless integration with various frontend clients and external systems capable of making HTTP request to predefined routes.

An overview of the auto-generated API documentation using Swagger UI is shown in Figure 3.7. The documentation highlights the resource-centric structure of the routes and the clearly defined format of a request.

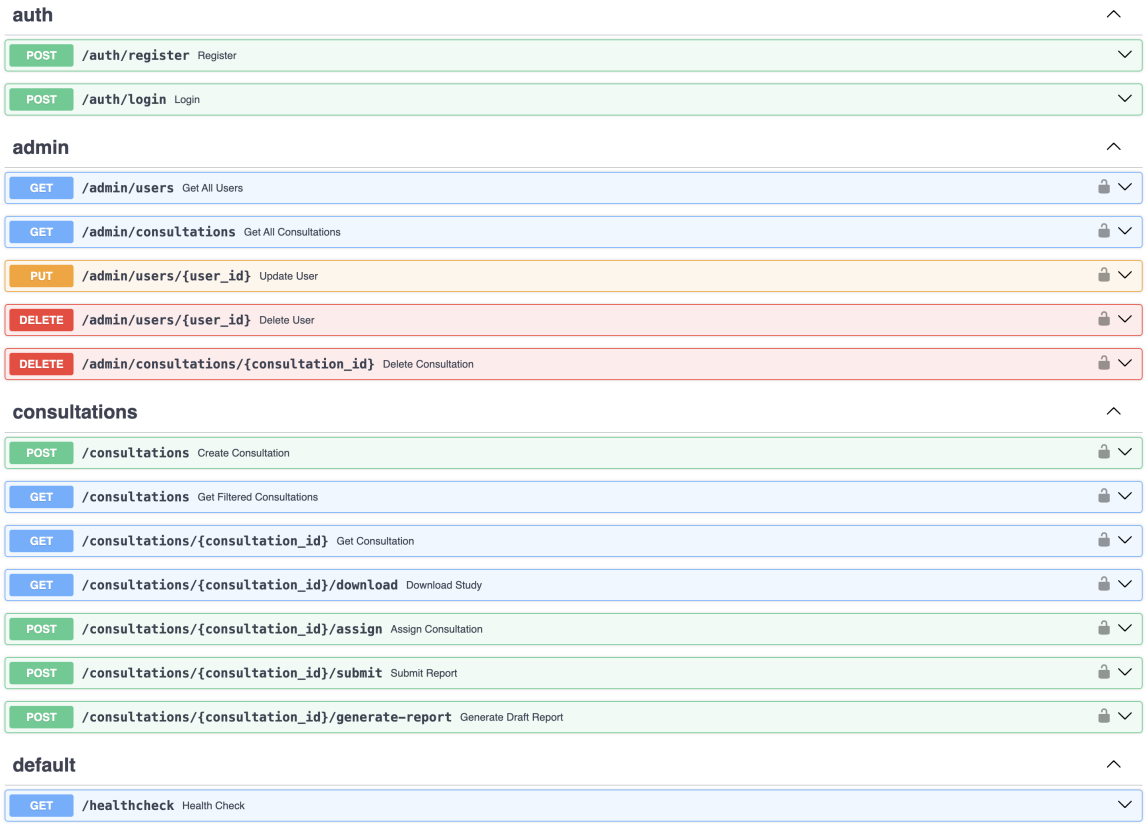


Figure 3.7: Swagger UI API documentation for VistaScan.

3.3.1.3 WebSocket Event-Driven Communication

To support real-time collaboration and instant notifications across the consultation workflow, VistaScan implements a **WebSocket event handling** system that enables bidirectional communication between the backend and connected clients.

The system follows a role-based **publish-subscribe pattern**, where the backend server acts as the event publisher, broadcasting structured notification events to relevant subscribers. Key event types include consultation creation, expert assignment, status transitions, and report completion. The connection manager, defined as a singleton class, effectively routes these events using three distribution strategies:

- **Role-based broadcasting:** Events sent to all users with specific roles (e.g., new consultations broadcast to all experts and admins).
- **Targeted messaging:** Direct notifications to specific users (e.g., patient notified when their consultation is assigned to an expert).
- **Global broadcasting:** System-wide announcements distributed to all connected clients.

This communication layer improves the interactivity of the platform, ensuring that both patients and experts remain informed of relevant actions, without requiring manual refreshes or polling mechanisms.

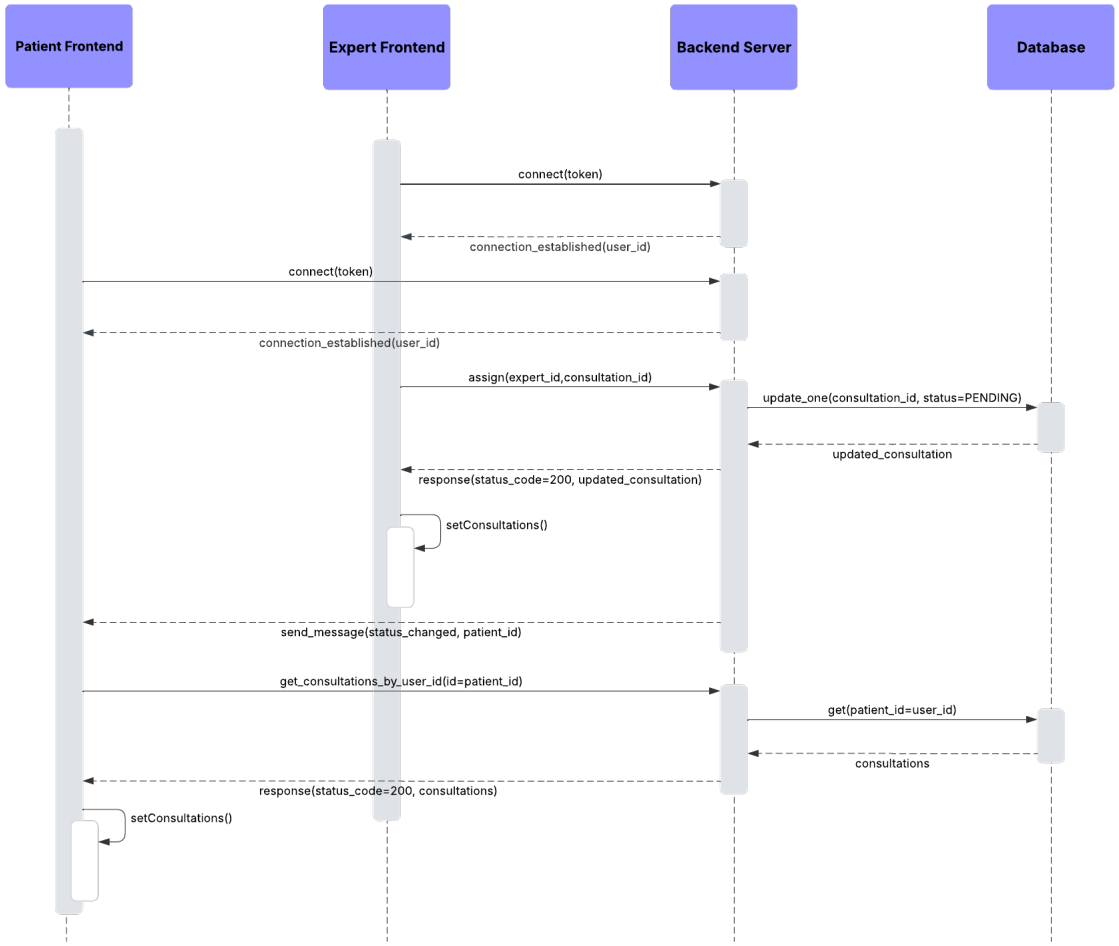


Figure 3.8: WebSocket event communication sequence diagram for claiming a pending consultation.

3.3.2 Model integration

The report generation model is integrated into VistaScan as an **external microservice**, which communicates with the application's backend via a RESTful API. In this setup, the backend acts as the client, sending imaging studies for annotating and receiving AI-generated text reports in response.

The model itself is implemented in **PyTorch** [PyT24], and all source code is written in Python. Therefore, to expose the model's inference functionality, the microservice is built using FastAPI as well, providing a clean and lightweight interface. The core endpoint, `/generate-report`, receives a medical image file as part of a `POST` request, processes it through the model's inference pipeline, and returns a structured text report as output.

This modular integration design ensures separation of concerns between the core application and the AI inference logic, enabling independent deployment, and

reusability for other potential medical workflows. An overview of the API definition and the model integration is shown in Listing 3.1.

```
# Load the model using a custom pipeline class
pipeline = CLIPXRGenPipeline.from_pretrained(
    model_path="/models/clip-xrgen-ft-mimic-cxr.pt",
    config_path="/configs/clip_xrgen_swin224_ft-mlpcls_bert.yaml",
)

# API route definition with predefined response model for automatic
   data validation
@app.post("/generate-report", response_model=ReportResponse)
async def generate_report(file: UploadFile = File(...)):
    # Check if a valid image file is provided
    if not file.content_type or not file.content_type.startswith('
   image/'):
        raise HTTPException(status_code=400, detail="File must be an
    image")

    try:
        # Load the binary content of the file
        image_bytes = await file.read()
        # Check if content is valid
        if len(image_bytes) == 0:
            raise HTTPException(status_code=400, detail="Empty image
    file")

        # Convert bytes to an image
        image = Image.open(io.BytesIO(image_bytes)).convert('RGB')

        # Generate the report using the model pipeline
        report = pipeline(image=image, num_beams=3, temperature=0.7)

        # Return the generated content with any additional defined
   metadata
        return ReportResponse(
            report=report,
            generated_at=datetime.now(),
        )

    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))
```

Listing 3.1: API endpoint definition for model inference integration.

3.3.3 Frontend

The frontend of a web application is the user-facing module responsible for displaying data, collecting user input, and enabling interaction with the underlying system. It serves as the interface through which users access the application's functionality, translating backend services and business logic into an intuitive and responsive user experience.

VistaScan's frontend is developed using **React** [Met24], a widely adopted JavaScript library created by Meta for building dynamic user interfaces. React promotes a **component-based architecture**, which UI is divided into reusable, self-contained

units called components. Each component manages its own logic and rendering behavior, which makes the interface easier to maintain, scale and text.

The application follows a **Single-Page Application** (SPA) model, meaning that it loads a single HTML page into the browser that is dynamically updated without full page reloads. This gives the option for client-side routing and significantly enhances the user experience by allowing fast transitions between views, such as navigating the dashboard or viewing consultation details.

React was also chosen for the strong developer experience it offers largely due to **JSX** (JavaScript XML), a syntax extension that allows developers to write HTML-like structures directly in JavaScript code. This simplifies the structure and readability of components, and integrates naturally with React's state management and hooks system, which control how the UI updates in response to user actions, and trigger optimized DOM re-renderings.

3.3.3.1 State Management

To efficiently manage application state and asynchronous data fetching, VistaScan adopts **Redux Toolkit Query** (RTK Query) [Red24], a modern data fetching and caching library built on top of Redux Toolkit. RTK query simplifies the handling of server-side data and API interactions by abstracting away boilerplate logic while ensuring data consistency across the application.

The library's caching mechanisms are particularly beneficial for medical consultation workflows, where patient data and imaging studies must remain synchronized between components. When consultation statuses change through WebSocket events, RTK query automatically invalidates relevant cache entries and refetches updated information, ensuring users always view the current consultation states without the need of manual refresh.

3.3.3.2 Authentication and Authorization

The application implements **JWT-based (JSON Web Token)** authentication to secure access to medical consultation data and enforce role-based functionality. JWT is a compact, URL-safe token format used to securely transmit user identity and claims between client and server. Upon successful login or register, the frontend receives a JWT token from the backend containing user credentials and role information, which is then securely stored locally and automatically attached to all API requests as a Bearer token.

Frontend route protection utilizes a custom Route component that verifies the token validity before rendering protected components. The application distinguishes between public routes (e.g., `/login` or `/register`), authenticated routes requiring valid tokens (e.g., `/dashboard`).

Authorization is implemented through role-based access control, where JWT role claims (`PATIENT`, `EXPERT`, `ADMIN`) determine component visibility and feature availability. This ensures that patients correctly access their consultation data and history, experts can see pending cases and review already assigned ones, and admins maintain system control over all entities.

Token lifecycle management includes automatic logout for expired tokens and consistent authentication across both HTTP API calls and WebSocket connections, providing a unified application security solution for serving all users.

3.4 User manual

In order to facilitate the first interaction with the application and ensure a clearer understanding of its interface, a user manual is included for VistaScan, which provides a guided overview of the core functionalities and main interaction flows. It is designed to help users navigate the system efficiently and take full advantage of its capabilities for streamlining radiology consultation.

3.4.1 Account registration and login

The first step in using VistaScan is to access the user account. Upon navigating to the application's homepage (Figure 3.9), the user is greeted with an overview of the platform's features, along with the options to either log in to an already existing account, or register a new one. The registration process implicitly creates a patient account, since the access for medical experts is granted exclusively though administrator-assigned credentials.

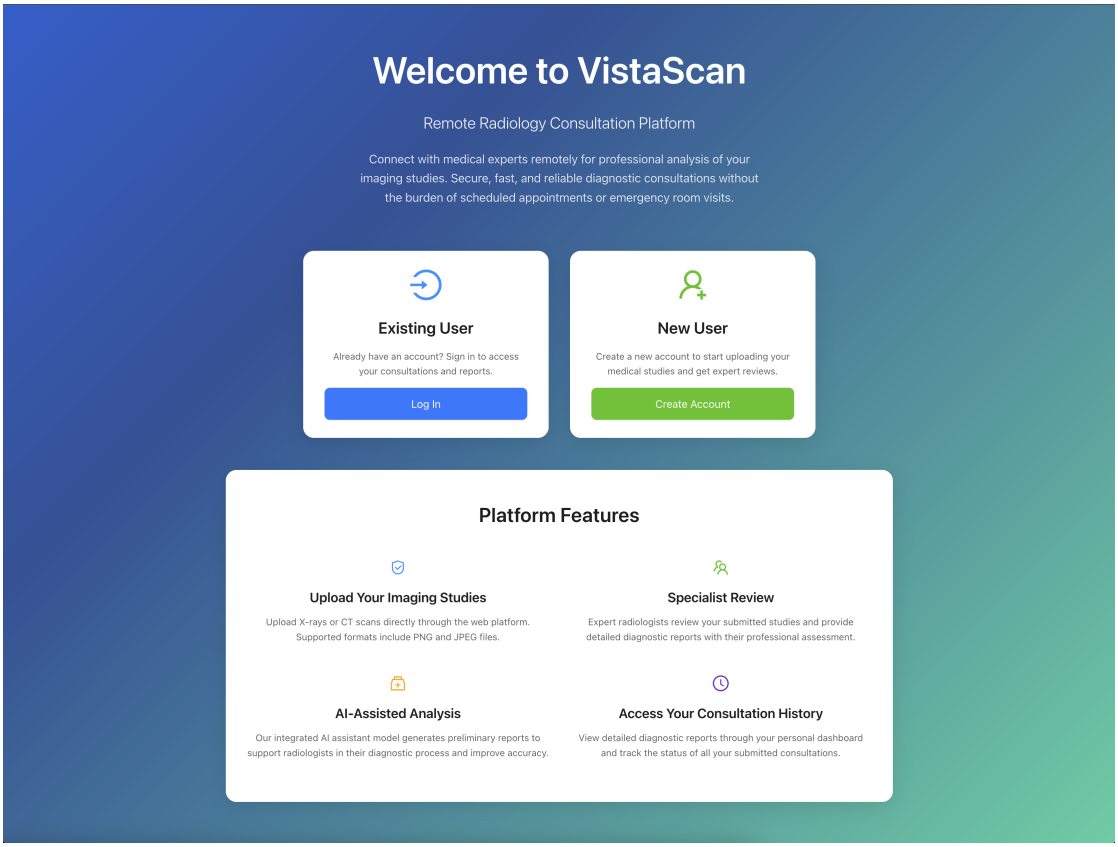


Figure 3.9: VistaScan's landing page, offering the user the options to navigate to the login or register portals.

3.4.2 Consultation dashboard interaction

Once successfully authenticated, the user is directed to a central dashboard, adjusted to his role and permissions (Figure 3.10. Patients are presented with the option of initiating a new radiology consultation, as well as viewing the status and

history of their previous submissions. Experts, on the other hand, have access to a list of pending consultations submitted by active patients, along with the options to review ongoing cases or revisit completed ones.

3.4.3 Imaging study submission and review

From the dashboard, the patient can upload an imaging study (e.g., chest X-ray), using the dedicated upload component. The submission process is straightforward, allowing the user to either drag and drop a file or browse the local device. Once the upload is initiated and validated, a new consultation is automatically created marked as *pending*, making it immediately visible to the available radiologists.

A radiologist expert can then access the pending submissions, inspect the uploaded images, and choose to begin a review. Once selected, the case is assigned only to the specific expert and no longer accessible to others. In the review interface, the radiologist uses the available annotation tools to support the diagnostic process (Figure 3.11).

Therefore, the expert has the option to invoke the integrated AI assistant model to generate a preliminary report, that highlights possible findings. This draft is meant to be used as guidance and the radiologist is expected to refine it, ensuring the report accurately reflects the patient's condition and provides appropriate recommendations. If the expert decides however to not intervene upon the generated draft, an automated note is included in the final submitted report to inform the patient regarding potential inaccuracies of the final verdict.

After the report is submitted, the consultation is updated to *completed*, and the result becomes immediately available to the patient in the consultation view (Figure 3.12).

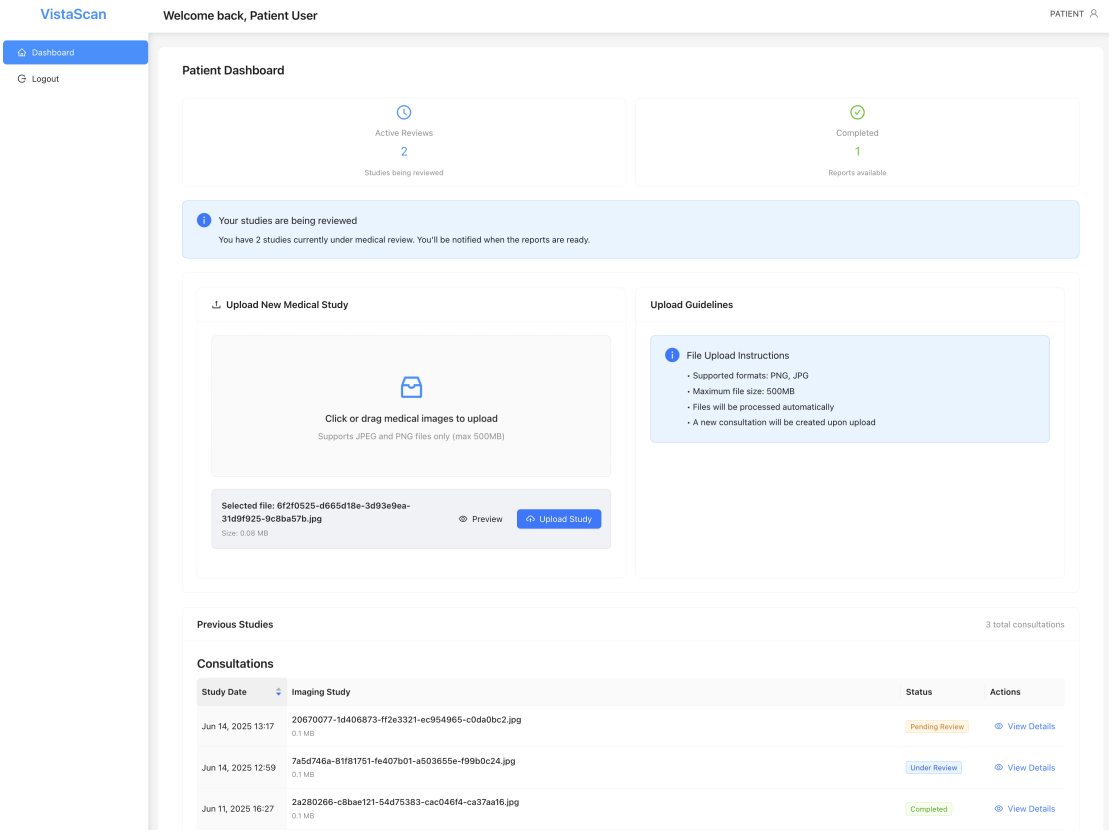
3.4.4 Platform management

System oversight is ensured through the admin dashboard. In addition to having access to all core flows of the platform, from login to consultation review, the admin user can manage all registered accounts (Figure 3.13). This includes creating new users, particularly for adding expert accounts, updating user details, and deleting accounts when necessary.

3.5 Future enhancements

While the proposed application represents a strong candidate for a remote consultation platform, significantly streamlining the consultation workflow, there still remains room for improvement. The underlying architectural choices and technologies make the system highly scalable and well-prepared for future expansion of its core functionalities.

One key enhancement necessary for real-world clinical adoption is support for multiple imaging file formats. Although the system currently accepts commonly used compressed formats such as PNG or JPEG, these formats may not retain the full fidelity of the original medical images. In certain scenarios, this limitation may as well prevent potential clients from fully benefiting from the platform. Supporting



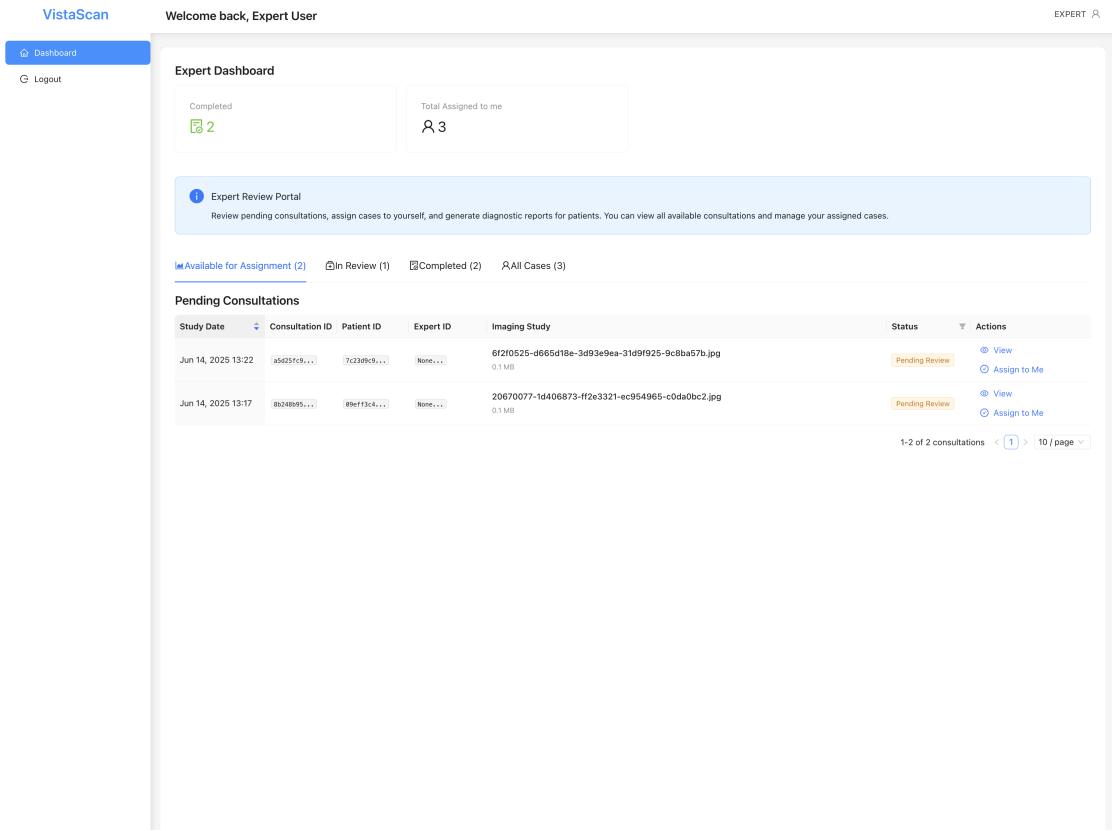


Figure 3.10: Role-based consultation dashboards in VistaScan, for accessing and managing medical cases.

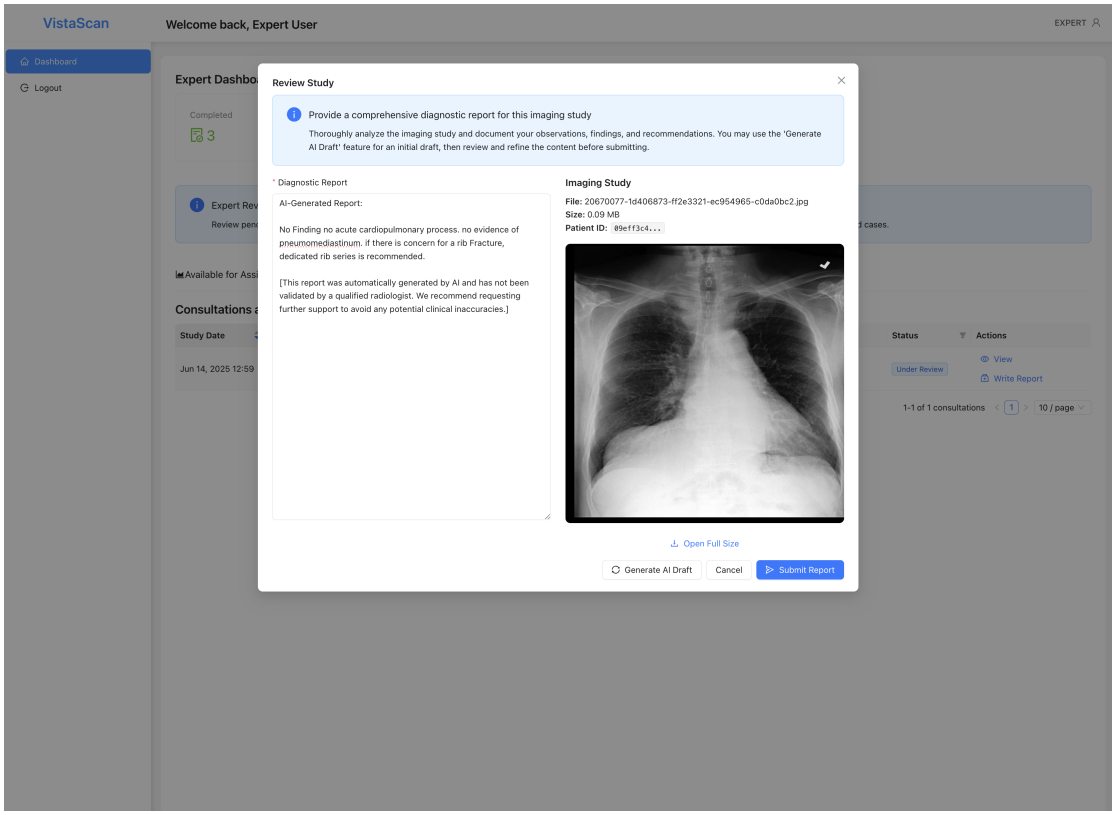


Figure 3.11: Expert view of consultation details in VistaScan with additional functionality for reviewing consultations and annotating imaging studies.

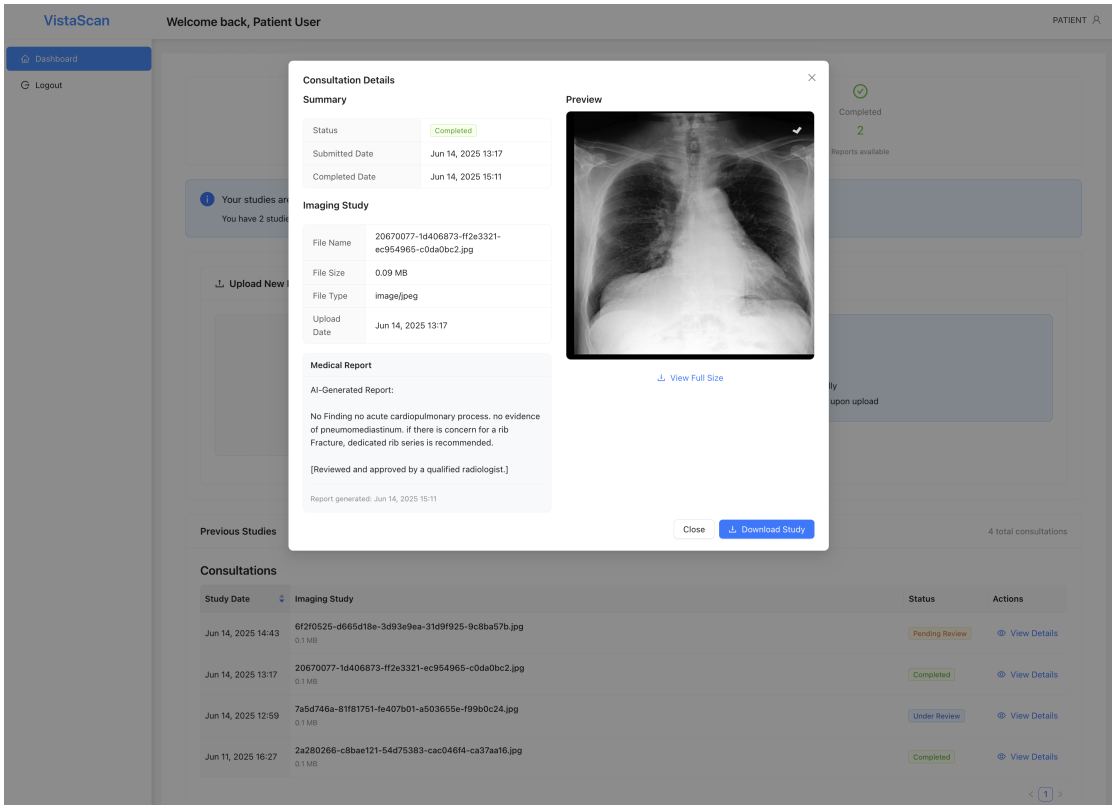


Figure 3.12: Patient view of consultation details in VistaScan, displaying the uploaded imaging study details and the submitted diagnostic report.

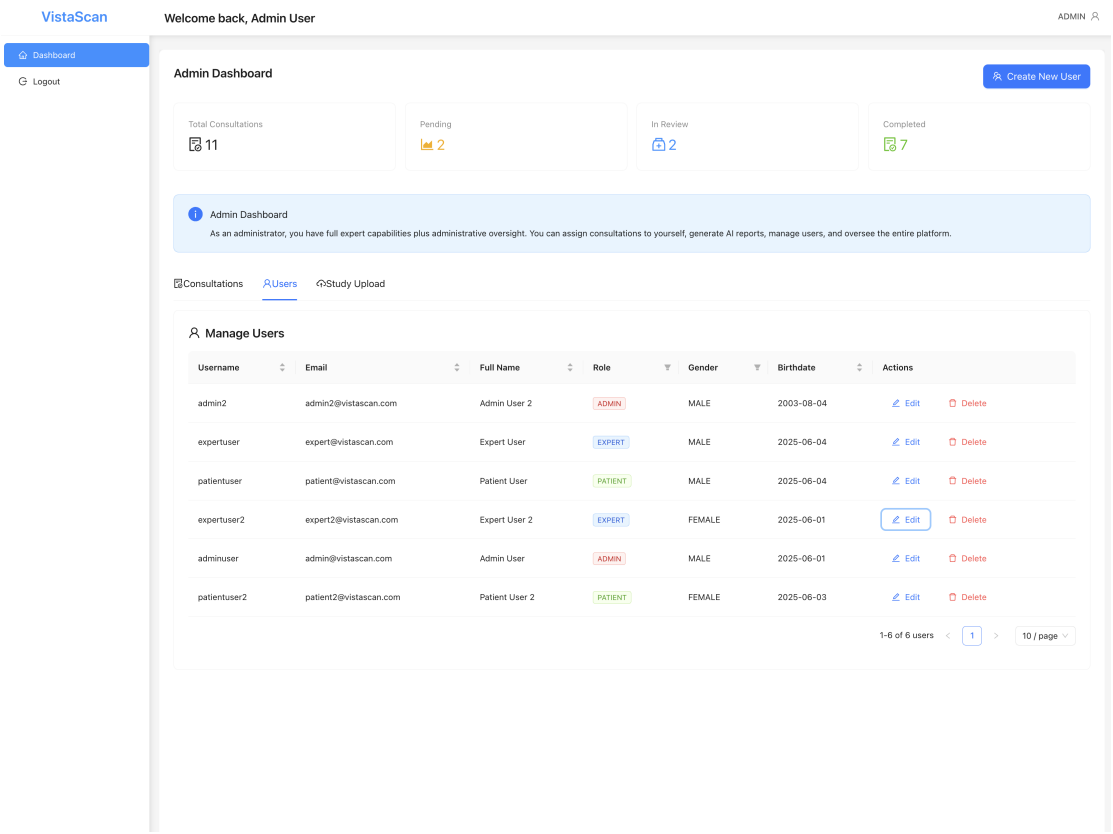


Figure 3.13: Admin dashboard view for managing users and consultation workflows in VistaScan.

widely adopted formats like DICOM is therefore essential to ensure compatibility with standard radiology workflows.

Another critical area of improvement relates to the explainability of the integrated intelligent system and the displayed information. As emphasized throughout the design and methodology of the proposed solution, transparency is a fundamental aspect in medical applications. In this context, both the model's API and the user interface should be extended to provide more informative feedback about the system's decision-making process. For example, displaying confidence scores associated with each predicted finding in the generated report can help radiologists interpret the AI output more effectively and use it as a reliable reference during diagnosis.

Conclusions

This thesis introduced **CLIP-XRGen**, a hybrid multimodal framework designed for both vision-language understanding and conditional text generation in the medical domain. The model leverages a novel training objective that incorporates concept-level supervision to enhance cross-modal alignment, resulting in semantically meaningful representations. This contrastive-based pretraining approach addresses key challenges in medical imaging and reporting, resulting in a robust backbone that is later adapted to downstream tasks, including medical concept classification and radiology report generation, using a decoupled training strategy.

Experimental results demonstrated that CLIP-XRGen effectively learns concept-aware representations, achieving competitive performance in aligning chest X-ray images with associated textual descriptions and accurately identifying medical conditions. While the generated reports may be less fluent and contain more noise than those produced by specialized models, our prompt-guidance strategy remains valid through the concept-centered nature of the generated sequences.

To validate its practical applicability, **VistaScan** was developed, a scalable and flexible web-based application designed as a remote radiology consultation platform. By integrating CLIP-XRGen into a complete diagnostic workflow, the system offers decision-making support to radiologists and facilitates faster evaluations. This contributes to mitigating persistent inefficiencies in modern healthcare, including overcrowded medical facilities and limited access to specialized care.

Despite the promising outcomes, several limitations reveal valuable directions for future work. Enhancing model explainability remains a key priority for building trust among clinicians. Furthermore, integrating more advanced language models may also improve the coherence and reliability of generated reports. Incorporating medical knowledge sources may further enrich contextual understanding and support deeper multimodal alignment. On the application side, the platform offers room for scaling toward more specialized workflows and broader clinical use cases.

Ultimately, this work marks a step forward in the development of intelligent multimodal systems for clinical support. Although it has not yet reached the performance level required for widespread adoption, it demonstrates the potential of contrastive pretraining and concept-guided generation as a foundation for future solutions in AI-assisted healthcare.

Bibliography

[BPA+24] Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, Mark Ibrahim, Melissa Hall, Yunyang Xiong, Jonathan Lebensold, Candace Ross, Srihari Jayakumar, Chuan Guo, Diane Bouchacourt, Haider Al-Tahan, Karthik Padthe, Vasu Sharma, Hu Xu, Xiaoqing Ellen Tan, Megan Richards, Samuel Lavoie, Pietro Astolfi, Reyhane Askari Hemmat, Jun Chen, Kushal Tirumala, Rim Assouel, Mazda Moayeri, Arjang Talattof, Kamalika Chaudhuri, Zechun Liu, Xilun Chen, Quentin Garrido, Karen Ullrich, Aishwarya Agrawal, Kate Saenko, Asli Celikyilmaz, and Vikas Chandra. An Introduction to Vision-Language Modeling. *CoRR*, abs/2405.17247, 2024.

[Coc05] Alistair Cockburn. Hexagonal architecture. `https://alistair.cockburn.us/hexagonal-architecture`, 2005. Accessed: 2025-06-11.

[CSCW20] Zhihong Chen, Yan Song, Tsung-Hui Chang, and Xiang Wan. Generating Radiology Reports via Memory-driven Transformer. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1439–1449, Online, November 2020. Association for Computational Linguistics.

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[DKR+16] Dina Demner-Fushman, Marc D. Kohli, Marc B. Rosenman, Sonya E. Shooshan, Laritza Rodriguez, Sameer K. Antani, George R. Thoma, and Clement J. McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *J. Am. Medical Informatics Assoc.*, 23(2):304–310, 2016.

[EKK+21] Mark Endo, Rayan Krishnan, Viswesh Krishna, Andrew Y. Ng, and Pranav Rajpurkar. Retrieval-Based Chest X-Ray Report Generation Using a Pre-trained Contrastive Language-Image Model. In Subhrajit Roy, Stephen Pfohl, Emma Rocheteau, Girmaw Abebe Tadesse, Luis Oala, Fabian Falck, Yuyin Zhou, Liyue Shen, Ghada Zamzmi, Purity Mugambi, Ayah Zirikly, Matthew B. A. McDermott, and Emily Alsentzer, editors, *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, pages 209–219. PMLR, 04 Dec 2021.

[(ES11] European (ESR. Good practice for radiological reporting. Guidelines from the European Society of Radiology (ESR). *Insights into Imaging*, 2:93–96, 04 2011.

[GAG+13] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[HLL+23] Xiaodi Hou, Zhi Liu, Xiaobo Li, Xingwang Li, Shengtian Sang, and Yijia Zhang. MKCL: Medical Knowledge with Contrastive Learning model for radiology report generation. *Journal of Biomedical Informatics*, 146:104496, 2023.

[IRK+19] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David Mong, Safwan Halabi, Jesse Sandberg, Ricky Jones, David Larson, Curtis Langlotz, Bhavik Patel, Matthew Lungren, and Andrew Ng. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:590–597, 07 2019.

[JLP+24] Alistair Johnson, Matthew Lungren, Yifan Peng, Zhiyong Lu, Roger Mark, Seth Berkowitz, and Steven Horng. MIMIC-CXR-JPG - chest radiographs with structured labels (version 2.1.0). `https://doi.org/10.13026/jsn5-t979`, 2024. RRID:SCR_007345.

[JPB+19] Alistair Johnson, Tom Pollard, Seth Berkowitz, Nathaniel Greenbaum, Matthew Lungren, Chih-ying Deng, Roger Mark, and Steven Horng. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6:317, 12 2019.

[LGBADF18] Jason Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific Data*, 5:180251, 11 2018.

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[LLG+20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[LLSH23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models . In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023.

[LLXH22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation . In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 2022.

[Mar17] Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall Press, USA, 1st edition, 2017.

[Met24] Meta. React Documentation, 2024. Accessed: 2025-06-09.

[Mih25] Tudor-Octavian Mihăiță. CLIP-XRad: Learning Multimodal Representations of Chest X-rays through Contrastive Pretraining with Medical Concept Alignment. *Studia Studia Universitatis Babeş-Bolyai Informatica*, page submitted for publication, 2025.

[Min24] MinIO Inc. MinIO Documentation, 2024. Accessed: 2025-06-01.

[Mon24] MongoDB Inc. MongoDB Documentation, 2024. Accessed: 2025-06-01.

[Pal08] Jeffrey Palermo. The Onion Architecture. `https://jeffreypalermo.com/tag/onion-architecture/`, July 2008. Accessed: 2025-06-11.

[PyT24] PyTorch Team. PyTorch Documentation, 2024. Accessed: 2025-06-01.

[Ram24] Sebastián Ramirez. FastAPI Documentation, 2024. Accessed: 2025-06-01.

[RCR22] Vignav Ramesh, Nathan A. Chi, and Pranav Rajpurkar. Improving Radiology Report Generation Systems by Removing Hallucinated References to Non-existent Priors. In Antonio Parziale, Monica Agrawal, Shalmali Joshi, Irene Y. Chen, Shengpu Tang, Luis Oala, and Adarsh Subbaswamy, editors, *Proceedings of the 2nd Machine Learning for Health symposium*, volume 193 of *Proceedings of Machine Learning Research*, pages 456–473. PMLR, 28 Nov 2022.

[Red24] Redux Toolkit Team. RTK Query Overview, 2024. Accessed: 2025-06-09.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training , 2018.

[SCSM24] Phillip Sloan, Philip Clatworthy, Edwin Simpson, and Majid Mirmehdi. Automated Radiology Report Generation: A Review of Recent Advances. *IEEE Reviews in Biomedical Engineering*, 18:368–387, 2024.

[SJR+20] Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Ng, and Matthew Lungren. Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1500–1519, Online, November 2020. Association for Computational Linguistics.

[TBF22] Ajay Kumar Tanwani, Joelle K. Barral, and Daniel Freedman. RepsNet: Combining Vision with Language for Automated Medical Reports. In Linwei Wang, Qi Dou, P. Thomas Fletcher, Stefanie Speidel, and Shuo Li, editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2022 - 25th International Conference, Singapore, September 18-22, 2022, Proceedings, Part V*, volume 13435 of *Lecture Notes in Computer Science*, pages 714–724. Springer, 2022.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Proceedings of the 31st International Conference*

on Neural Information Processing Systems, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[WWAS22] Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. MedCLIP: Contrastive learning from unpaired medical images and text. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3876–3887, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[YGH+23] Kihyun You, Jawook Gu, Jiyeon Ham, Beomhee Park, Jiho Kim, Eun K. Hong, Woonhyuk Baek, and Byungseok Roh. CXR-CLIP: Toward Large Scale Chest X-ray Language-Image Pre-training . In Hayit Greenspan, Anant Madabhushi, Parvin Mousavi, Septimiu Salcudean, James Duncan, Tanveer Syeda-Mahmood, and Russell Taylor, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pages 101–111, Cham, 2023. Springer Nature Switzerland.

[ZLW+23] Zihao Zhao, Yuxiao Liu, Han Wu, Yonghao Li, Sheng Wang, Lin Teng, Disheng Liu, Zhiming Cui, Qian Wang, and Dinggang Shen. CLIP in medical imaging: A comprehensive survey. *CoRR*, abs/2312.07353, 2023.