

# 3D-CT-GPT++: ENHANCING 3D RADIOLOGY REPORT GENERATION WITH DIRECT PREFERENCE OPTIMIZATION AND LARGE VISION-LANGUAGE MODELS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Automatically generating radiology reports from three-dimensional medical images, such as 3D CT scans, plays a crucial role in modern diagnostics. Current approaches for generating 3D reports often adopt video processing methods, which struggle to effectively capture the relationships along the Z-axis. Additionally, multimodal large language model-based methods for generating 3D image reports face significant limitations, particularly in terms of the image encoder’s ability to represent 3D structures and the hallucinations that arise in generated content. To address these challenges, we propose the 3D-CT-GPT++ model. This model integrates the optimized 3D image encoder CTViT-V, specifically designed for chest CT scans, and builds upon the LLaVA-1.5 architecture. Furthermore, we introduce *Direct Preference Optimization (DPO)*, where GPT-4 is used to score the outputs of our fully fine-tuned (SFT) model, creating a preference dataset for subsequent DPO training. DPO significantly reduces hallucinations in the report generation process, ensuring the generated reports are more aligned with clinical needs. We fine-tuned the model on both high-quality private and public datasets to ensure clinical relevance. Extensive experiments were conducted using standard natural language generation (NLG) evaluation metrics, including BLEU, METEOR, and ROUGE-L, to assess the report generation performance. Experimental results demonstrate that 3D-CT-GPT++ significantly outperforms existing methods in terms of accuracy, fluency, and clinical relevance, advancing the automation of 3D medical report generation.

## 1 INTRODUCTION

Medical imaging plays a critical role in modern diagnostics, providing clinicians with precise anatomical information for accurate medical decisions (Liu et al., 2024). Three-dimensional computed tomography (3D CT), in particular, offers richer spatial information compared to two-dimensional (2D) images, aiding in the detection of pathological details that traditional techniques may miss. However, current CT image interpretation relies heavily on manual analysis by radiologists, which is time-consuming, error-prone, and adds to the clinical workload (Farahani et al., 2017). While advances have been made in generating 2D image reports (Chen et al., 2022; 2020; Qin & Song, 2022), processing 3D images, such as 3D CT scans, remains challenging due to the complexity of spatial feature extraction and high computational costs (Li et al., 2023b). Maintaining slice consistency across multiple slices is a key issue, as it is crucial for accurate diagnosis. These challenges drive the need for models that can efficiently process 3D data while preserving spatial coherence and improving diagnostic accuracy.

Despite advancements in generating reports from 3D CT images, existing approaches still face significant challenges. As shown in Figure 1, models like RadMD (Wu et al., 2023) and M3D (Bai et al., 2024) employ 3D Vision Transformers (3DViT) for feature extraction from 3D CT scans. However, processing high-dimensional 3D images with 3DViT often requires compressing high-resolution data, leading to the potential loss of critical medical details that affect diagnostic

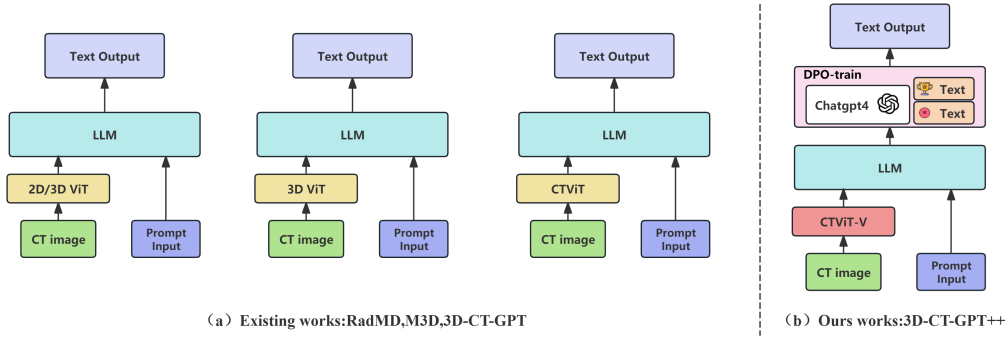


Figure 1: Comparison of architectures for RadMD, M3D-LaMed, 3D-CT-GPT, and our proposed 3D-CT-GPT++ models.

accuracy. To address these limitations, [Chen et al. \(2024a\)](#) introduced the 3D-CT-GPT model (see [Figure 1](#)), which integrates the CTViT encoder to enhance the extraction of spatial and temporal features. While this approach improves performance by treating 3D images as video sequences, the reliance on causal Transformers to extract temporal features limits the model’s ability to fully capture dependencies across slices, especially in complex multi-slice scenarios.

Moreover, a significant challenge in 3D medical report generation is the occurrence of hallucinations—where the model generates content that deviates from the actual data, resulting in inaccurate or irrelevant reports. This issue is particularly critical in the medical domain, where errors can have serious consequences. The hallucination problem is exacerbated by the lack of high-quality, aligned multimodal datasets necessary for effectively training models to produce clinically accurate reports ([Liu et al., 2023b](#)). Existing methods to mitigate hallucinations often involve Reinforcement Learning from Human Feedback (RLHF) ([Li et al., 2023d](#)), which relies on human-annotated data to guide the model’s outputs. However, RLHF can be resource-intensive and challenging to scale due to the high costs and time associated with collecting human feedback.

To address these challenges, As illustrated in [Figure 1\(b\)](#), we optimized the original CTViT model and proposed the CTViT-V model. This model introduces a slice Transformer and relative position encoding, enhancing the feature extraction capabilities of 3D CT images, particularly in capturing global slice dependencies. Building on this improvement, we combined the LLaVA 1.5 ([Liu et al., 2023a](#)) architecture with the 3D-CT-GPT model to propose the 3D-CT-GPT++ model, which more efficiently processes 3D CT image data and generates more accurate and comprehensive radiology reports. To reduce hallucinations and avoid the scalability issues of RLHF, we adopt *Direct Preference Optimization (DPO)* ([Rafailov et al., 2023](#)), using GPT-4 ([OpenAI et al., 2023](#)) to automatically score the outputs of our supervised fine-tuned (SFT) model. GPT-4 effectively mimics human judgment, providing a scalable alternative to human feedback. This approach creates a preference dataset for fine-tuning, enabling the model to generate clinically aligned reports without the high costs of manual annotations.

In summary, our main contributions are:

- We propose an enhanced CTViT-V model that incorporates a slice Transformer and relative position encoding to capture global dependencies across 3D CT slices. This improvement enhances spatial coherence and diagnostic accuracy while reducing computational overhead.
- We introduce the 3D-CT-GPT++ model, based on the LLaVA-1.5 architecture, which optimizes 3D CT image processing by effectively enhancing both spatial and temporal feature extraction. This leads to more accurate and context-aware radiology reports.
- We apply *Direct Preference Optimization (DPO)* to 3D medical imaging report generation, leveraging GPT-4 to create a preference dataset for fine-tuning. This approach provides a practical solution to reduce hallucinations in generated reports without incurring the high costs associated with human feedback.

## 2 RELATED WORK

**Multimodal Medical Large Models (MMLMs).** In recent years, the development of Multimodal Medical Large Models (MMLMs) has greatly advanced the automatic generation of medical imaging reports. Current methods for generating medical reports mainly include model aggregation, joint vision-language model generation, and end-to-end fine-tuning approaches. Model aggregation methods combine outputs from multiple models and use specially designed prompts to generate complete reports, as seen in ChatCAD (Wang et al., 2023b) and ChatCAD+ (Zhao et al., 2024). Joint vision-language model generation methods, such as XrayGPT (Thawkar et al., 2023) and XrayPULSE, extract image features through vision encoders and integrate them with language models to achieve effective multimodal fusion. End-to-end fine-tuning methods, such as Med-PaLM (Tu et al., 2023) and XrayGLM (Wang et al., 2023a), perform joint training on image and text data, significantly improving the model’s understanding and generation capabilities. Some of the leading models, such as LLaVA-Med (Li et al., 2023a), Med-PaLM2 (Singhal et al., 2023) and MedFlamingo Moor et al. (2023), have demonstrated strong performance in 2D image analysis by leveraging large-scale medical image datasets and language models. However, these models still face challenges when processing 3D images, such as CT and MRI scans, due to the complexity of spatial feature extraction and high computational costs. Although RadFM (Wu et al., 2023) and M3D-LaMed (Bai et al., 2024) have explored 3D image analysis, the generated reports still lack coherence and accuracy. Additionally, the 3D-CT-GPT () model also exhibits limitations in maintaining global dependencies between slices.

**Preference Optimization and Reinforcement Learning.** Reinforcement Learning from Human Feedback (RLHF) (Li et al., 2023d) has been widely applied to improve the output quality of Large Language Models (LLMs). By incorporating human preference data, RLHF allows models to gradually learn to generate more reliable and useful outputs, especially in multimodal tasks. However, RLHF faces several challenges when applied to multimodal tasks, particularly in aligning different modalities, such as text, images, and videos. Designing an effective reward system is crucial, as poor reward design can lead to models generating inaccurate or irrelevant content. Additionally, scaling RLHF is costly, especially when collecting large-scale preference data. For instance, Sun et al. (2023) reported that collecting 10,000 human-labeled preference data points for LLaVA-RLHF cost approximately \$3,000. Li et al. (2023c) also encountered scalability issues when applying GPT-4V to preference modeling, particularly when handling video inputs. Ahn et al. (2024) proposed using Supervised Fine-Tuning (SFT) models for self-evaluation, although this approach has not yet been fully validated for complex video-related tasks.

## 3 METHODOLOGY

In this section, we introduce the 3D-CT-GPT++ model, designed to enhance the automatic generation of radiology reports from 3D medical images, specifically chest CT scans. As shown in Figure 2, the model leverages the optimized 3D image encoder, CTViT-V, to efficiently process complex volumetric data while a pre-trained large language model (LLM) generates coherent and contextually accurate reports. Additionally, Direct Preference Optimization (DPO) is integrated to ensure that the generated reports align with clinicians’ diagnostic preferences, and feedback from GPT-4 is incorporated to further refine the reports’ accuracy and coherence.

### 3.1 THE 3D ENCODER: CTViT-V

To efficiently process 3D CT scans and capture global dependencies across slices, we propose an enhanced encoder architecture called CTViT-V, building upon the original CTViT model (Hamamci et al., 2023). Figure 2(a) illustrates the architecture of the proposed CTViT-V model. Our key improvements are:

- **Slice Transformer:** We introduce a slice Transformer module that allows for full attention across all slices, enabling the model to capture global context and dependencies in the Z-axis direction. Unlike the causal temporal attention in the original CTViT, which limits interactions to adjacent slices, our approach allows for bidirectional information flow, enhancing the model’s ability to understand complex spatial relationships.

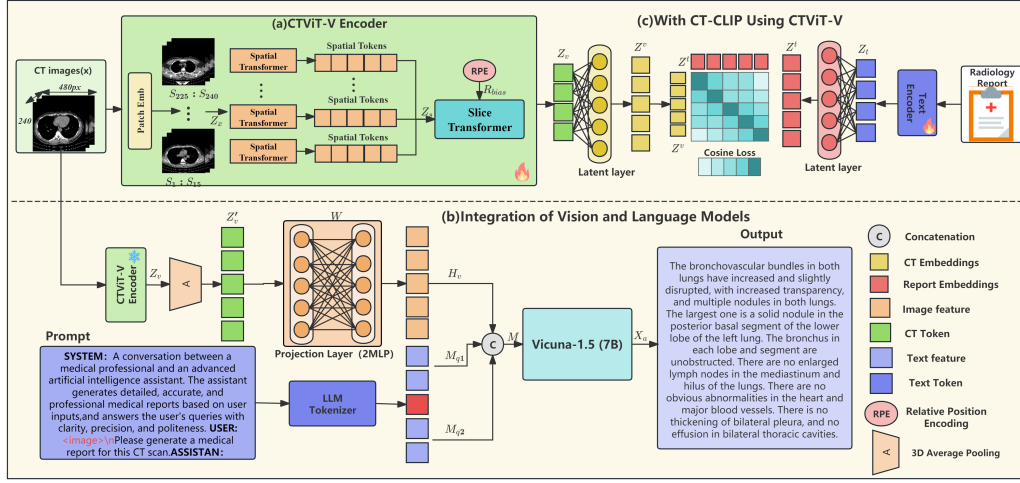


Figure 2: Architecture of 3D-CT-GPT++: The diagram illustrates how the CTViT-V encoder integrates with an LLM to generate radiology reports. CT images are processed through the Slice Transformer, and the visual features are transformed into language embeddings through an MLP, eventually generating the report.

- **Relative Position Encoding:** To better model the positional relationships between slices, we incorporate relative position encoding (Shaw et al., 2018) within the slice Transformer. This allows the model to recognize the relative distances and positions of slices, improving spatial coherence and the accuracy of feature extraction.
- **Computational Efficiency:** By optimizing the attention mechanisms and incorporating efficient architectural designs inspired by ViViT (Arnab et al., 2021) and C-ViViT (Villegas et al., 2022), we reduce the computational overhead associated with processing high-resolution 3D volumes. This makes it feasible to process multi-slice CT scans without significant loss of detail.

Detailed processes are as follows:

$$Z_s = T_s(Z_x) \quad \text{where} \quad Z_x \in \mathbb{R}^{B \times S \times H \times W \times C} \quad (1)$$

Here,  $Z_x$  represents the input patches of the CT volume,  $T_s$  is the spatial Transformer, and  $Z_s$  is the spatially encoded representation. Subsequently, the Slice Transformer, enhanced with relative position encoding, models the dependencies between the slices:

$$Z_v = T_d(Z_s + R_{\text{bias}}) \quad (2)$$

where  $T_d$  is the Slice Transformer, and  $R_{\text{bias}}$  represents the relative position encoding. This step allows the model to capture both local and global spatiotemporal features, resulting in improved pathological feature detection.

**Overall Encoding Process** The encoded features are then pooled using 3D average pooling to reduce both spatial and temporal resolution:

$$Z'_v = \text{AvgPool3D}(Z_v) \quad (3)$$

The pooled tensor is reshaped to flatten the spatial and temporal dimensions, producing compact feature maps suitable for integration with the language model. For detailed steps, please refer to Appendix A.1, where the full algorithm is provided.

### 3.2 VISION-LANGUAGE INTEGRATION

After obtaining the encoded visual features  $Z'_v$  from the CTViT-V encoder, we integrate these features with a pre-trained large language model (LLM) to generate radiology reports. The overall architecture follows the LLaVA-1.5 framework (Liu et al., 2023a), with modifications to accommodate the 3D visual features. As shown in Figure 2(b), we combine the pre-trained Vicuna-1.5 (7B) (Chiang et al., 2023) language model with the CTViT-V encoder. The visual features are transformed into language embedding tokens  $H_v$  using a lightweight 2-layer MLP projection matrix  $W$ , matching the dimensionality of the word embeddings in the LLM:

$$H_v = W \cdot Z'_v. \quad (4)$$

During the report generation process, the prompt text (e.g., a clinician’s question or instruction) is combined with an image placeholder to form the input prompt. This prompt is tokenized by the LLM’s tokenizer, generating text tokens  $M_q$ . These tokens are split into  $M_{q1}$  and  $M_{q2}$  around the image placeholder. The visual tokens  $H_v$  are concatenated with these text tokens to form the complete input:

$$M = \text{concat}([M_{q1}, H_v, M_{q2}]). \quad (5)$$

The LLM processes this input to generate the output tokens, which are decoded into the final radiology report  $X_a$ :

$$X_a = \text{LLM}(M). \quad (6)$$

For a detailed description of the vision-language integration and further steps involved, please refer to Appendix A.2.

### 3.3 DIRECT PREFERENCE OPTIMIZATION (DPO)

After training, we have developed a model capable of generating reports from 3D CT lung medical images. However, like other generative models, ours still encounters challenges such as hallucination, where the generated reports may include information inconsistent with real clinical scenarios. To address this, we introduce *Direct Preference Optimization (DPO)* into 3D CT medical image report generation, drawing inspiration from LLaVA-Hound-DPO (Zhang et al., 2024). Our approach utilizes GPT-4 to score the outputs of our supervised fine-tuned (SFT) model, creating a preference dataset that guides DPO training, ensuring the generated reports align more closely with clinicians’ diagnostic preferences.

#### 3.3.1 GPT-4 FOR SCORING AND CONSTRUCTING THE PREFERENCE DATASET

To construct the preference dataset, we first use the trained SFT model to generate a large number of medical reports from preprocessed public and private datasets. Then, these reports are evaluated using the GPT-4 API. The scoring process involves inputting the ground truth reports, model-generated reports, and detailed descriptions as supporting evidence, as illustrated in Figure 3.

As shown in Figure 3(A), we sample multiple outputs from the 3D-CT-GPT++(SFT) model using a temperature setting of 1.0, ensuring diverse responses. For each 3D CT chest image and its corresponding prompt, we generate six report outputs using the 3D-CT-GPT++(SFT) model. In Figure 3(B), GPT-4 evaluates the outputs based on the evaluation prompt (*eval\_prompt*), which includes both the ground truth and model-generated reports, providing feedback in the form of language-based explanations and numerical scores. The detailed evaluation prompt used for scoring is presented in Figure 4 in Appendix A.5.

We randomly select reports with scores  $\geq 3$  as positive examples and treat reports with scores below 3 as negative examples. If all responses are uniformly scored above or below 3, the instance is excluded from the dataset. Finally, based on these scores, we construct a preference dataset. Formally, the dataset is represented as:

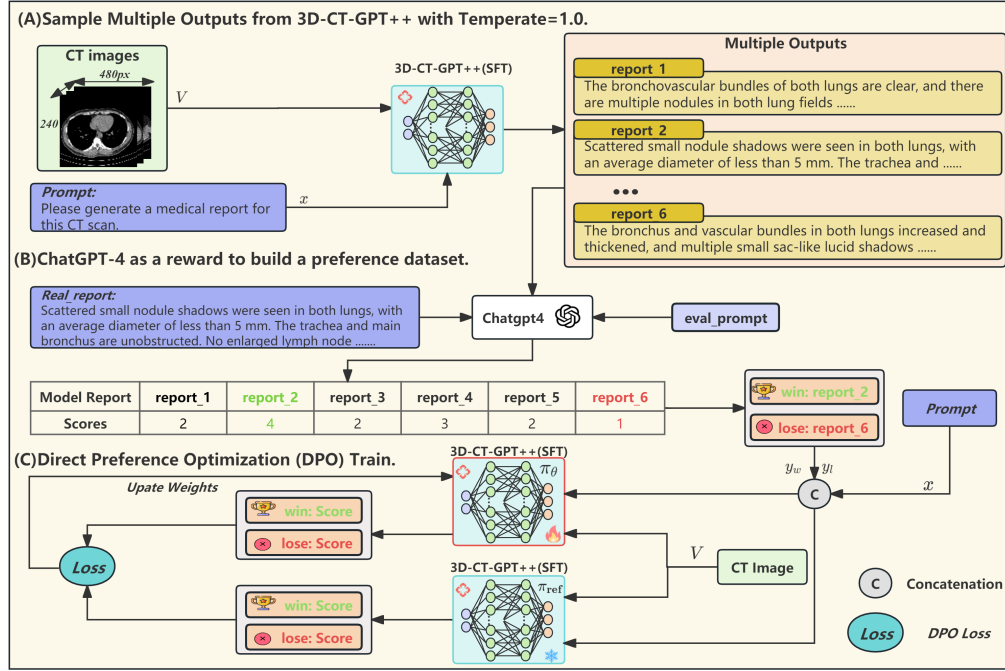


Figure 3: Overview of the Direct Preference Optimization (DPO) process for generating medical reports from 3D CT lung images. (A) Sampling diverse outputs from the model. (B) Evaluation by GPT-4, comparing outputs to ground truth. (C) DPO training objective optimizing the policy model ( $\pi_\theta$ ) against the reference model ( $\pi_{ref}$ ).

$$\mathcal{D}_{DPO} = \{(V, x, y_w, y_l)\} \quad (7)$$

where  $V$  represents the 3D CT lung image,  $x$  is the prompt, and  $y_w$  and  $y_l$  are the positive and negative model-generated reports, respectively.

### 3.3.2 DPO TRAINING OBJECTIVE

The DPO objective is defined as follows:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(V, x, y_w, y_l) \sim \mathcal{D}_{DPO}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_w | x, V)}{\pi_{ref}(y_w | x, V)} - \beta \log \frac{\pi_\theta(y_l | x, V)}{\pi_{ref}(y_l | x, V)} \right) \right]$$

As illustrated in Figure 3(C),  $\pi_\theta$  is the policy model to be optimized, and  $\pi_{ref}$  is the reference model initialized with SFT weights. Here,  $\sigma$  is the logistic function, and  $\beta$  is set to 0.1. This objective function optimizes the model by maximizing the log probability ratio between positive and negative samples, thereby enhancing the quality and factual consistency of the generated reports.

To clearly illustrate the DPO training process, we provide detailed pseudocode in Appendix A.3.

## 3.4 DATASET

**Data Collection** For this study, we utilized a subset of the publicly available CT-RATE dataset <sup>?</sup>, which includes 25,692 non-contrast chest CT volumes. After various reconstruction techniques, this dataset was expanded to 50,188 volumes representing 21,304 unique patients, paired with corresponding radiology reports, abnormality labels, and metadata. From this, we selected 8,070 cases for our experiments. Additionally, we collected 2,000 3D chest CT scans and their



corresponding radiology reports from a well-known international hospital, which we refer to as *Dataset-XY*. These scans cover a wide age range (20 to 88 years) and have a mean axial resolution of 512x512 pixels, with slices per volume ranging from 100 to 600. Both datasets have been anonymized and de-identified prior to use.

**Data Preprocessing** We performed extensive preprocessing to ensure the quality and consistency of the image and report data. For *Dataset-XY*, we applied standard de-identification protocols, removed duplicates, and filtered out irrelevant text in the reports. For the CT images, we excluded low-resolution and redundant scans, followed by manual review to further ensure consistency and uniformity. Both datasets were converted to Hounsfield Units (HU) using metadata slope and intercept values, and cropped to the range  $[-1000 \text{ HU}, +200 \text{ HU}]$  to reflect the diagnostic limits. Volumes were resampled to uniform spacing of 0.75 mm along the x and y axes and 1.5 mm along the z axis, and resized to a consistent resolution of 240x480x480. For a summary of dataset statistics, including the number of cases and average report length, see Table 3 in Appendix A.4.

### 3.5 TRAINING PROCESS

We divided the training process into four main stages.

**Stage 1: Image Encoder Training** In this stage, we trained the CTViT-V encoder on a large-scale 3D CT scan dataset using the CT-CLIP framework. CT-CLIP, a 3D adaptation of the CLIP architecture, was employed for self-supervised contrastive learning, aligning image and text embeddings in a shared latent space, as illustrated in Figure 2(c).

**Self-Supervised Contrastive Learning:** We utilized contrastive learning to maximize the similarity between matching 3D CT scans and corresponding radiology reports (positive pairs) while minimizing similarity between non-matching pairs (negative pairs). This method enables the model to learn effective, discriminative representations of the 3D data.

**Image Encoder Implementation:** CTViT-V extracts low-dimensional CT tokens from volumetric images, which are projected into a shared 512-dimensional space. This enables contrastive learning with the text encoder, ensuring robust image representation.

**Text Encoder Integration:** For the text encoder, we used a pre-trained CXR-BERT, which encodes radiology reports into the same 512-dimensional space for contrastive alignment.

**Training Objective:** The primary training objective is to align image and text embeddings by maximizing the cosine similarity between positive pairs and minimizing the similarity between negative pairs. This optimizes the CTViT-V encoder for extracting features from 3D CT images, ensuring effective medical report generation.

**Stage 2: Pre-training** In this stage, the model was trained to understand the relationship between 3D CT image features and their corresponding reports by analyzing a large set of 3D CT image-report pairs. During this phase, both the image encoder and the language model were frozen, and we focused on training the projection layer. The training was conducted using a custom-built dataset. Due to the scarcity of paired 3D CT images and reports, we were unable to perform large-scale alignment training typical of multimodal models. Instead, we employed an interactive approach across multiple data types to address this challenge.

**Stage 3: Fine-tuning** In this stage, we further refined the model to align 3D CT image features with specific radiology reports. We employed two fine-tuning strategies:

**LoRA-based Lightweight Fine-tuning** During this phase, we employed LoRA (Low-Rank Adaptation) (Hu et al., 2021) for lightweight fine-tuning. The image encoder remained frozen, while parts of the language model and projection layers were fine-tuned. This efficient approach helped avoid overfitting.

**Supervised Fine-tuning (SFT)** Additionally, during this phase, we applied supervised fine-tuning (SFT) (Brown et al., 2020), optimizing all parameters of the language model using labeled data. This contrasts with parameter-efficient methods like LoRA, which adjust only a subset of layers. The number of training epochs was carefully selected through cross-validation to prevent overfitting.

**Stage 4: Direct Preference Optimization (DPO)** After fine-tuning, the model underwent *Direct Preference Optimization (DPO)* to ensure that generated reports aligned more closely with clinicians’ diagnostic preferences.

**GPT-4 Scoring and Preference Dataset** GPT-4 was used to score the generated reports based on factual consistency and coherence, producing a preference dataset. Reports with scores above a threshold were used as positive examples, and those below the threshold served as negative examples, as described in Section 3.3.1.

**DPO Training** Using this preference dataset, the model was trained to optimize report generation by adjusting the likelihood of producing higher-scoring reports, following the DPO objective. This helped the model generate more clinically relevant and accurate reports, as described in Section 3.3.2.

## 4 EXPERIMENTS

In this section, we present the experimental setup, results, and analysis to evaluate the performance of our proposed model, 3D-CT-GPT++.

### 4.1 EXPERIMENTAL SETUP

Our experiments consist of four main stages: Image Encoder Training, Model Pre-training, Fine-tuning, and Direct Preference Optimization (DPO). Detailed hardware configurations, hyperparameter settings, and implementation specifics are provided in Appendix B.1. To ensure robustness and stability, we ran five experiments for each model variant and computed the average scores, minimizing the influence of random factors and ensuring more reliable performance evaluations. This methodology was consistently applied across all experiments. The evaluation metrics used in our analysis are detailed in Appendix B.2. Additionally, for each model configuration, specific training details—such as dataset preparation, number of training epochs, batch sizes, and optimizer settings—are provided in Appendix B.3 to ensure transparency and reproducibility. All experiments were conducted with a temperature setting of 0.7, except for those in the section 4.2

### 4.2 RESULTS AND ANALYSIS

**Overall Performance** As shown in Section (A) of Table 1, our model 3D-CT-GPT++ (SFT+DPO) achieves the best performance across all metrics compared to its counterparts. It significantly outperforms 3D-CT-GPT++ (LoRA) and 3D-CT-GPT++ (SFT). For instance, BLEU-4 improves from 10.50 and 10.16 in the LoRA and SFT variants to 13.32 with our model, indicating enhanced accuracy and coherence in generated reports. Similarly, ROUGE-L increases from 0.3306 and 0.3199 to 0.3692, suggesting that our model produces reports with structures more closely aligned with reference texts.

**Comparison with Existing Models** As shown in Section (B) of Table 1, our model 3D-CT-GPT++ significantly outperforms existing models RadFM and M3D (from the literature), as well as our baseline 3D-CT-GPT, across all comparable metrics. For example, our model achieves a ROUGE-L score of 0.3692, surpassing RadFM’s 15.51, M3D’s 19.55, and the baseline’s 0.3353. The METEOR score also improves from the baseline’s 0.3308 to 0.3542, notably higher than M3D’s 0.1438. These results highlight the effectiveness of our approach, attributed to the advanced encoder architecture and the incorporation of DPO, which better align the model’s outputs with human preferences.

**Impact of Temperature on Performance** Section (C) of Table 1 explores the effect of varying the temperature parameter during inference on the performance of 3D-CT-GPT++. A temperature of 0.4 yields the highest scores for most metrics, including BLEU-1, BLEU-4, ROUGE-1, and METEOR. Specifically, at temperature 0.4, the model achieves a BLEU-4 score of 14.47 and a ROUGE-L score of 0.3807. However, higher temperatures (e.g., 0.8 and 0.9) lead to a slight decline in performance metrics. This is expected, as higher temperatures introduce more randomness, increasing the diversity of the generated text but potentially reducing the overlap with reference texts, as measured



Table 1: Performance comparison of 3D-CT-GPT and M3D across different training strategies and datasets. The table presents the evaluation metrics BLEU-1, BLEU-4, ROUGE-1, ROUGE-2, ROUGE-L, and METEOR for different models, training strategies, and datasets. The best and second-best results for each metric are highlighted in **bold** and underlined, respectively. Sections are marked with labels (A) to (D) for easy reference in the text.

Model / Method	BLEU-1	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
<b>(A) Overall Performance</b>						
3D-CT-GPT++ (LoRA)	<u>55.98</u>	<u>10.50</u>	<u>0.4561</u>	<u>0.2209</u>	<u>0.3306</u>	<u>0.3061</u>
3D-CT-GPT++ (SFT)	<u>54.65</u>	<u>10.16</u>	<u>0.4505</u>	<u>0.2123</u>	<u>0.3199</u>	<u>0.2995</u>
3D-CT-GPT++ (SFT+DPO)	<b>56.76</b>	<b>13.32</b>	<b>0.5117</b>	<b>0.2467</b>	<b>0.3692</b>	<b>0.3542</b>
<b>(B) Comparison with Existing Models</b>						
RadFM (Literature Results)	10.21	-	-	-	0.1551	-
M3D (Literature Results)	15.15	-	-	-	0.1955	0.1438
3D-CT-GPT (Baseline)	<u>52.17</u>	11.49	0.4711	0.2224	<u>0.3353</u>	<u>0.3308</u>
3D-CT-GPT++	<b>56.76</b>	<b>13.32</b>	<b>0.5117</b>	<b>0.2467</b>	<b>0.3692</b>	<b>0.3542</b>
<b>(C) Impact of Temperature</b>						
Temperature 0.4	<b>57.87</b>	<b>14.47</b>	<b>0.5327</b>	<b>0.2593</b>	<b>0.3807</b>	<b>0.3695</b>
Temperature 0.5	<u>57.12</u>	<u>14.13</u>	<u>0.5250</u>	<u>0.2559</u>	<u>0.3784</u>	<u>0.3651</u>
Temperature 0.6	56.79	13.48	0.5148	0.2466	0.3705	0.3559
Temperature 0.7	56.76	13.32	0.5117	0.2467	0.3692	0.3542
Temperature 0.8	56.93	13.42	0.5068	0.2463	0.3675	0.3486
Temperature 0.9	55.61	13.16	0.5013	0.2415	0.3616	0.3494
<b>(D) Impact of Data Quantity on DPO Performance</b>						
3D-CT-GPT++ (1)	<b>56.76</b>	<b>13.32</b>	<b>0.5117</b>	<b>0.2467</b>	<b>0.3692</b>	<b>0.3542</b>
3D-CT-GPT++ (2)	<u>55.38</u>	<u>12.41</u>	<u>0.4857</u>	<u>0.2309</u>	<u>0.3479</u>	<u>0.3339</u>
3D-CT-GPT++ (SFT)	54.65	10.16	0.4505	0.2123	0.3199	0.2995

by BLEU and ROUGE scores. Based on these observations, a temperature of 0.4 appears to provide the best balance between diversity and accuracy for our task. However, depending on the specific requirements of report generation—such as the need for more deterministic outputs—a temperature of 0.7 may still be preferable.

**Impact of Data Quantity on DPO Performance** Section (D) of Table 1 compares the supervised fine-tuned model 3D-CT-GPT++ (SFT), the DPO-trained model using the initial data selection method 3D-CT-GPT++ (1), and the DPO-trained model with a refined data selection strategy 3D-CT-GPT++ (2). The refined strategy in 3D-CT-GPT++ (2) aimed to increase contrast by selecting the highest-scoring candidate as the positive example and the lowest-scoring as the negative; if no candidate scored  $\geq 3$ , we used the real report as the positive example. However, this refined strategy resulted in decreased performance compared to 3D-CT-GPT++ (1); for example, BLEU-4 decreased from 13.32 to 12.41. Similarly, [Chen et al. \(2024b\)](#) introduced a self-play method for DPO training, preferring ground-truth cases over model-generated responses. [Zhang et al. \(2024\)](#) found that their LLaVA-Hound-DPO ([Zhang et al., 2024](#)) model showed a 3% accuracy decline compared to SFT models when using self-play ([Chen et al., 2024b](#)). These findings suggest that while reward incorporation benefits complex tasks, extreme contrasts between examples may hinder performance. Our results support this, indicating that in DPO training, the quality and representativeness of preference data are more important than the quantity or extremity of examples.

### 4.3 ABLATION STUDY

We conducted an ablation study to evaluate the impact of different model configurations and datasets during the DPO phase. First, 3D-CT-GPT++ (LoRA) was compared with (a) 3D-CT-GPT++ (CTViT) and (b) 3D-CT-GPT++ (CT-RATE+LoRA). LoRA achieved a higher BLEU-1 (55.98 vs. 52.17) but slightly lower BLEU-4 (10.50 vs. 11.49) compared to (a), indicating better lexical diversity but some trade-offs in sequence generation. Comparing (b) and LoRA, the custom dataset used in (b) led to higher BLEU-1 (61.43) but much lower BLEU-4 (4.58), showing improved lexical

Table 2: Ablation study results for 3D-CT-GPT++ (LoRA), (CTViT), and (CT-RATE+LoRA) during the DPO phase, highlighting the impact of dataset choice, MLP unfreezing, and learning rate adjustments. Best results are in **bold**, second-best are underlined.

Ablation Study	BLEU-1	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
(a) 3D-CT-GPT++(CTViT)	52.17	<b>11.49</b>	<b>0.4711</b>	<b>0.2224</b>	<b>0.3353</b>	<b>0.3308</b>
(b) 3D-CT-GPT++ (CT-RATE+LoRA)	<b>61.43</b>	4.58	0.3364	0.1643	0.2599	0.1791
3D-CT-GPT++ (LoRA)	<u>55.98</u>	<u>10.50</u>	<u>0.4561</u>	<u>0.2209</u>	<u>0.3306</u>	<u>0.3061</u>
(c) Unfreeze MLP(2)	54.89	12.46	0.4847	0.2336	0.3496	0.3405
(d) Learning Rate $3 \times 10^{-7}$ (2)	<b>55.76</b>	10.81	0.4644	0.2197	0.3310	0.3134
3D-CT-GPT++ (DPO)(2)	<b>54.99</b>	<b>12.68</b>	<b>0.4873</b>	<b>0.2363</b>	<b>0.3516</b>	<b>0.3445</b>

diversity but poorer performance on longer sequences. We further analyzed the effect of unfreezing the MLP (c) and adjusting the learning rate (d). Unfreezing the MLP improved BLEU-4 (12.46) and ROUGE metrics, enhancing accuracy and relevance, while a lower learning rate resulted in reduced BLEU-4 (10.81). The 3D-CT-GPT++ (DPO)(2) configuration balanced performance well, achieving a BLEU-4 of 12.68 and an improved ROUGE-L of 0.3516, indicating optimal trade-offs for long-text generation.

#### 4.4 QUALITATIVE ANALYSIS

We compare generated reports from various versions of 3D-CT-GPT++ (DPO, SFT, LoRA, and the baseline model) with the real medical report. Examples are provided in Figure 5 of Appendix C. The 3D-CT-GPT++ models consistently produce more accurate and detailed reports, capturing clinical findings and using appropriate medical terminology. Compared to baseline models, 3D-CT-GPT++ shows clear improvements in report quality and accuracy.

## 5 CONCLUSION

We have introduced 3D-CT-GPT++, a novel model for radiology report generation that leverages advanced encoder architectures and Direct Preference Optimization to achieve superior performance. Through comprehensive experiments and ablation studies, we demonstrate the model’s effectiveness and potential for clinical applications.

### 5.1 LIMITATIONS AND FUTURE WORK

While 3D-CT-GPT++ has demonstrated significant advancements in radiology report generation, several limitations remain. Our dataset is restricted to lung CT images, limiting the model’s ability to generalize to other medical conditions and rare cases. Additionally, we have not conducted large-scale clinical validations, necessitating clinician feedback to assess the practical effectiveness of the generated reports. Our evaluation relies primarily on standard Natural Language Generation (NLG) metrics, which do not fully capture the clinical relevance and utility of the reports. Moreover, the model’s high parameter count and computational complexity may hinder deployment in resource-constrained environments, increasing training and inference costs.

Future work will address these limitations through several initiatives. We plan to expand our dataset to include diverse medical imaging types, such as X-rays and MRIs, to enhance generalization. Collaborating with medical institutions, we aim to conduct large-scale clinical trials to gather clinician feedback and optimize model performance in real-world settings. We will also develop evaluation metrics focused on clinical relevance and improve model interpretability to ensure accuracy and clarity for clinicians. Additionally, we will explore more efficient model architectures and optimization techniques to reduce computational requirements, facilitating deployment in healthcare environments. Our ultimate goal is to develop a versatile large-scale model capable of handling various medical imaging types, supporting diverse clinical applications, and generating precise and comprehensive radiology reports. By addressing these limitations and pursuing these future directions, we aim to enhance the performance and practicality of 3D-CT-GPT++, promoting its adoption in clinical radiology.

## REPRODUCIBILITY STATEMENT

We have made significant efforts to ensure the reproducibility of our results. All details regarding the architecture of 3D-CT-GPT++, including model parameters, training settings, and hyperparameter configurations, are clearly documented in the main text and Appendix.

## REFERENCES

- Daechul Ahn, Yura Choi, Youngjae Yu, Dongyeop Kang, and Jonghyun Choi. Tuning large multimodal models for videos using reinforcement learning from ai feedback, 2024. URL <https://arxiv.org/abs/2402.03746>.
- Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021.
- Fan Bai, Yuxin Du, Tiejun Huang, Max Q-H Meng, and Bo Zhao. M3d: Advancing 3d medical image analysis with multi-modal large language models. *arXiv preprint arXiv:2404.00578*, 2024.
- Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL <https://arxiv.org/abs/2005.14165>.
- Hao Chen, Wei Zhao, Yingli Li, Tianyang Zhong, Yisong Wang, Youlan Shang, Lei Guo, Junwei Han, Tianming Liu, Jun Liu, and Tuo Zhang. 3d-ct-gpt: Generating 3d radiology reports through integration of large vision-language models, 2024a. URL <https://arxiv.org/abs/2409.19330>.
- Zhihong Chen, Yan Song, Tsung-Hui Chang, and Xiang Wan. Generating radiology reports via memory-driven transformer. *arXiv preprint arXiv:2010.16056*, 2020.
- Zhihong Chen, Yaling Shen, Yan Song, and Xiang Wan. Cross-modal memory networks for radiology report generation. *arXiv preprint arXiv:2204.13258*, 2022.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models, 2024b. URL <https://arxiv.org/abs/2401.01335>.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- Navid Farahani, Alex Braun, Dylan Jutt, Todd Huffman, Nick Reder, Zheng Liu, Yukako Yagi, and Liron Pantanowitz. Three-dimensional imaging and scanning: Current and future applications for pathology. *Journal of Pathology Informatics*, 8(1):36, 2017. ISSN 2153-3539. doi: <https://doi.org/10.4103/jpi.jpi.32.17>. URL <https://www.sciencedirect.com/science/article/pii/S2153353922004400>.
- Ibrahim Ethem Hamamci, Sezgin Er, Anjany Sekuboyina, Enis Simsar, Alperen Tezcan, Ayse Gul nihan Simsek, Sevval Nil Esirgun, Furkan Almas, Irem Dogan, Muhammed Furkan Dasdelen, et al. Generatect: Text-conditional generation of 3d chest ct volumes. *arXiv preprint arXiv:2305.16037*, 2023.

- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.
- Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023a.
- Johann Li, Guangming Zhu, Cong Hua, Mingtao Feng, Basheer Bennamoun, Ping Li, Xiaoyuan Lu, Juan Song, Peiyi Shen, Xu Xu, et al. A systematic collection of medical image datasets for deep learning. *ACM Computing Surveys*, 56(5):1–51, 2023b.
- Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, and Lingpeng Kong. Silk: Preference distillation for large visual language models, 2023c. URL <https://arxiv.org/abs/2312.10665>.
- Zihao Li, Zhuoran Yang, and Mengdi Wang. Reinforcement learning with human feedback: Learning dynamic choices via pessimism, 2023d.
- Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.
- Chang Liu, Yuanhe Tian, Weidong Chen, Yan Song, and Yongdong Zhang. Bootstrapping large language models for radiology report generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18635–18643, Mar. 2024. doi: 10.1609/aaai.v38i17.29826. URL <https://ojs.aaai.org/index.php/AAAI/article/view/29826>.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023b.
- Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Cyril Zakka, Yash Dalmia, Eduardo Pontes Reis, Pranav Rajpurkar, and Jure Leskovec. Med-flamingo: a multimodal medical few-shot learner, 2023. URL <https://arxiv.org/abs/2307.15189>.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondrasiuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan,

- Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2023.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040>.
- Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W18-6319>.
- Han Qin and Yan Song. Reinforced cross-modal alignment for radiology report generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 448–458, 2022.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://arxiv.org/abs/2305.18290>.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations, 2018. URL <https://arxiv.org/abs/1803.02155>.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaeckermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Agüera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. Towards expert-level medical question answering with large language models, 2023. URL <https://arxiv.org/abs/2305.09617>.
- Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Salmon: Self-alignment with principle-following reward models. *arXiv preprint arXiv:2310.05910*, 2023.
- Omkar Thawkar, Abdelrahman Shaker, Sahal Shaji Mullappilly, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, Jorma Laaksonen, and Fahad Shahbaz Khan. Xraygpt: Chest radiographs summarization using medical vision-language models. *arXiv preprint arXiv:2306.07971*, 2023.



- Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaeckermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Chuck Lau, Ryutaro Tanno, Ira Ktena, Basil Mustafa, Aakanksha Chowdhery, Yun Liu, Simon Kornblith, David Fleet, Philip Mansfield, Sushant Prakash, Renee Wong, Sunny Virmani, Christopher Semturs, S Sara Mahdavi, Bradley Green, Ewa Dominowska, Blaise Agueray Arcas, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Karan Singhal, Pete Florence, Alan Karthikesalingam, and Vivek Natarajan. Towards generalist biomedical ai, 2023.
- Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022.
- Rongsheng Wang, Yaofei Duan, Junrong Li, Patrick Pang, and Tao Tan. Xrayglm: The first chinese medical multimodal model that chest radiographs summarization. <https://github.com/WangRongsheng/XrayGLM>, 2023a.
- Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. Chatcad: Interactive computer-aided diagnosis on medical image using large language models. *arXiv preprint arXiv:2302.07257*, 2023b.
- Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Towards generalist foundation model for radiology. *ArXiv*, abs/2308.02463, 2023. URL <https://api.semanticscholar.org/CorpusID:260611504>.
- Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. *arXiv preprint arXiv:2404.01258*, 2024.
- Zihao Zhao, Sheng Wang, Jinchun Gu, Yitao Zhu, Lanzhuju Mei, Zixu Zhuang, Zhiming Cui, Qian Wang, and Dinggang Shen. Chatcad+: Towards a universal and reliable interactive cad using llms. *IEEE Transactions on Medical Imaging*, 2024.

## A ALGORITHM

### A.1 3D CT IMAGE ENCODER ALGORITHM

This algorithm describes the process of encoding 3D CT images into spatial-temporal feature representations, which is a key step in extracting meaningful visual information for report generation.

---

#### Algorithm 1: 3D CT Image Encoding Process

---

**Input:** Normalized 3D CT image  $x \in \mathbb{R}^{B \times 1 \times 240 \times 480 \times 480}$ , where  $B$  is the batch size and 240 is the normalized number of slices.

**Output:** Encoded features  $Z'_v \in \mathbb{R}^{B \times (8 \times 8 \times 8) \times 512}$ , where  $(8 \times 8 \times 8)$  are the reduced dimensions after pooling and 512 is the embedding dimension for each patch.

- 1 **Step 1: Patch Segmentation and Embedding**
  - 2 Divide the 3D CT image  $x$  into non-overlapping patches of size  $15 \times 30 \times 30$ :
  - 3  $Z_x \leftarrow \text{PatchEmbedding}(x)$  where  $Z_x \in \mathbb{R}^{B \times 16 \times 16 \times 16 \times 512}$
  - 4 **Step 2: Spatial Encoding**
  - 5 Apply the spatial transformer  $T_s$  to each patch:
  - 6 **for**  $s = 1$  **to** 16 **do**
  - 7    $Z_s \leftarrow T_s(Z_x)$  // Encode spatial dependencies
  - 8 Stack the encoded patches:
  - 9  $Z_s \in \mathbb{R}^{B \times 16 \times 16 \times 16 \times 512}$
  - 10 **Step 3: Slice Encoding**
  - 11 Apply relative position bias  $R_{bias}$  and use slice transformer  $T_d$ :
  - 12 **for**  $z_i \in Z_s$  **do**
  - 13    $A_{bias\_slice} \leftarrow R_{bias}(16)$
  - 14    $Z_v \leftarrow T_d(Z_s, A_{bias\_slice})$  // Encode slice-wise dependencies
  - 15 After slice encoding:
  - 16  $Z_v \in \mathbb{R}^{B \times 16 \times 16 \times 16 \times 512}$
  - 17 **Step 4: Apply 3D Average Pooling**
  - 18 Apply 3D average pooling with kernel size  $2 \times 2 \times 2$ :
  - 19  $Z_v \leftarrow \text{AvgPool3D}(Z_v, \text{kernel size} = 2)$  where  $Z_v \in \mathbb{R}^{B \times 8 \times 8 \times 8 \times 512}$
  - 20 **Step 5: Reshape the Tensor**
  - 21 Reshape pooled tensor to merge dimensions:
  - 22  $Z'_v \leftarrow \text{Reshape}(Z_v, [B, (8 \times 8 \times 8), 512])$  where  $Z'_v \in \mathbb{R}^{B \times (8 \times 8 \times 8) \times 512}$
  - 23 **Step 6: Return Projected CT Tokens**
  - 24 **return**  $Z'_v$
- 

### A.2 VISUAL-LANGUAGE INTEGRATION PROCESS

Once the 3D CT images are encoded into feature representations, the next step is to integrate these features with a pre-trained large language model (LLM). This process is outlined in the following algorithm, which corresponds to the integration and report generation components discussed in Section 3.2 of the main paper.

**Algorithm 2:** Visual-Language Integration Process (3D-CT-GPT++)**Input:** Normalized 3D CT image  $X_v \in \mathbb{R}^{B \times C \times S \times H \times W}$ , Query text  $Q$ **Output:** Generated report  $X_a$ 


---

```

1 1: Extract Visual Features
2  $Z_v \leftarrow g(X_v)$  // Extract visual features using trained CTViT encoder (Algorithm 1)
3  $H_v \leftarrow W(Z_v)$  // Project visual features into language embedding space using 2-layer MLP
4 2: Process Query Text
5  $Q_{tokens} \leftarrow \text{LLM\_Tokenizer}(Q)$  // Tokenize the query text
6  $M_q \leftarrow \text{LLM\_Embedding}(Q_{tokens})$  // Map tokens into word embeddings
7  $M_{q1}, M_{q2} \leftarrow \text{Split}(M_q)$  // Split text embedding into two parts around image placeholder
8 3: Concatenate Text and Visual Features
9  $M \leftarrow \text{Concat}([M_{q1}, H_v, M_{q2}])$  // Concatenate visual and text embeddings
10 4: Generate Output using LLM
11  $X_{a.tokens} \leftarrow g(M)$  // Pass the combined embeddings into the LLM to generate output tokens
12  $X_a \leftarrow \text{LLM\_Decoder}(X_{a.tokens})$  // Decode output tokens into final report
13 5: Return Generated Report
14 return  $X_a$ 

```

---

## A.3 DIRECT PREFERENCE OPTIMIZATION (DPO)

**Algorithm 3:** Direct Preference Optimization (DPO) Training Algorithm

---

**Input :** Pretrained model  $\pi_\theta$ , Reference model  $\pi_{\text{ref}}$ , GPT-4 scoring function, Dataset  $\mathcal{D}_{\text{DPO}}$ , Learning rate  $\alpha$ , Number of epochs  $T$ , Temperature parameter  $\tau$ , Scaling parameter  $\beta = 0.1$

**Output:** Optimized model  $\pi_\theta$ 

```

1 for each epoch  $t = 1$  to  $T$  do
2   for each data sample  $(V, x) \in \mathcal{D}_{\text{DPO}}$  do
3     Generate multiple outputs  $y_1, \dots, y_6$  from model  $\pi_\theta$ ;
4     Score each output using GPT-4:  $s_i = \text{GPT-4\_score}(y_i)$  for  $i = 1, \dots, 6$ ;
5     Select  $y_w$  where  $s_w \geq 3$  as the positive example;
6     Select  $y_l$  where  $s_l < 3$  as the negative example;
7     if no  $y_w$  with  $s_w \geq 3$  or no  $y_l$  with  $s_l < 3$  then
8       continue to next sample;
9     else
10      Compute  $\Delta = \beta \left( \log \frac{\pi_\theta(y_w|x, V)}{\pi_{\text{ref}}(y_w|x, V)} - \log \frac{\pi_\theta(y_l|x, V)}{\pi_{\text{ref}}(y_l|x, V)} \right)$ ;
11      Compute DPO loss:  $\mathcal{L}_{\text{DPO}} = -\log \sigma(\Delta)$ ;
12      Update model parameters:  $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{DPO}}$ ;
13    end
14  end
15 end
16 return Optimized model  $\pi_\theta$ ;

```

---

## A.4 DATASET STATISTICS

Table 3: Dataset statistics for CT-RATE and Dataset-XY.

Dataset	CT-RATE			Dataset-XY		
	Train	Test	Val	Train	Test	Val
<b>Images</b>	6456	807	807	1508	190	188
<b>Reports</b>	6456	807	807	1508	190	188
<b>Avg. Length (words)</b>	198.7	196.0	198.9	88.4	88.6	88.9

## A.5 DETAILED EVALUATION PROMPT

Your task is to act as an impartial and objective assessor of CT reports. Utilizing the ground truth and the model-generated CT report, your evaluation should focus on the following aspects:

- **Accuracy**: Does the model-generated report accurately reflect the findings and diagnosis in the real CT report?
- **Completeness**: Does the model-generated report cover all key details and findings? Does it omit any important information?
- **Clarity**: Is the model-generated report clear? Are there any repetitions, unclear descriptions, or grammatical errors?
- **Consistency**: Is the model-generated report consistent with the conclusions and descriptions in the real report?

**Input**:

Real CT Report: {real\_report}  
 Model-Generated Report: {model\_report}

**Output Format**:

Explanation: <brief judgement of prediction>  
 Score: <an integer score of quality from 1-5>

**Samples**:

**Input**:

Real CT Report: {The bronchovascular bundles of both lungs are clear, no obvious major lesions are seen, the bronchus of each lobe and segment are unobstructed, no enlarged lymph nodes are seen in the mediastinum and hilus of both lungs, no obvious abnormalities are seen in the heart and major blood vessels, no thickening of bilateral pleura, and no effusion is seen in bilateral thoracic cavity. Multiple slightly larger lymph nodes were seen in both armpits, with the largest one having a short diameter of 8 mm.}

Model-Generated Report: {The bronchovascular bundles of both lungs are clear, no obvious major lesions are seen, the bronchus of each lobe and segment are unobstructed, a few cord shadows are visible in the lower lobes of the left and right lung, multiple small lymph nodes are scattered in both hilars, with a short diameter of less than 5 mm, some aorta calcification is visible, and no significant effusion is seen in bilateral thoracic cavity.}

**Output**:

Explanation: <The model-generated report is partially accurate, but it includes additional findings not mentioned in the real CT report, such as cord shadows in the lower lobes of the lungs, scattered small lymph nodes in both hilars, and aorta calcification. These discrepancies affect the accuracy and consistency of the report. The report is clear and grammatically correct, but it is not complete as it omits the finding of multiple slightly larger lymph nodes in both armpits from the real report.>  
 Score: <3/5>

Figure 4: Detailed evaluation prompt used by GPT-4 for assessing radiology reports. This prompt includes both the ground truth and the model-generated reports, along with instructions for providing feedback and scoring.

## B DETAILED EXPERIMENTAL SETUP

### B.1 HARDWARE CONFIGURATION

Our experiments were conducted on various GPU configurations to ensure efficiency in training and fine-tuning. For the image encoder training phase, we utilized an L20 GPU (48GB), training with 4 batches, which occupied about 46GB of memory, with a learning rate set to  $1.25 \times 10^{-6}$ . In the pre-training phase, a single RTX 3090 GPU (24GB memory) was used with a learning rate set to  $1 \times 10^{-3}$  and a batch size of 1. During the instruction fine-tuning phase, the learning rate and batch size were adjusted to  $2 \times 10^{-4}$  and 1, respectively, utilizing the Adam optimizer, cosine learning rate scheduler, and bfloat16 precision. The pre-training process required approximately 14GB of GPU memory, while the fine-tuning phase (using LoRA) required about 22GB. In the SFT phase, we employed a single NVIDIA A100 GPU (80GB memory), with each batch utilizing approximately 68GB of memory. During the DPO phase, we adopted a hybrid training strategy, where the GPU occupied around 28GB of memory, while 30 CPU cores were used for computation, with a learning rate set to  $5 \times 10^{-7}$ .

## B.2 EVALUATION METRICS

To evaluate our radiology report generation model, we use standard NLG metrics: BLEU (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), and ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004), computed using the sacrebleu (Post, 2018), nltk, and rouge\_score libraries. BLEU measures n-gram overlap up to four in length, focusing on precision, with scores ranging from 0 to 100. In contrast, METEOR and ROUGE scores typically range from 0 to 1. METEOR accounts for synonyms, stemming, and paraphrasing, balancing precision and recall to capture both accuracy and completeness. ROUGE-1, ROUGE-2, and ROUGE-L measure unigram, bigram, and longest common subsequence overlap, respectively, assessing fluency and coherence. These metrics collectively provide a comprehensive evaluation of text quality, focusing on precision, recall, and structural similarity. While they don’t directly measure clinical accuracy, they are widely accepted for assessing the linguistic quality of generated reports in medical applications.

## B.3 TRAINING DETAILS FOR RESULT AND ABLATION

### B.3.1 OVERALL PERFORMANCE EXPERIMENTAL SETUP

To evaluate the performance of different versions of 3D-CT-GPT++ (LoRA, SFT, and SFT+DPO), we used the private Dataset-XY, with all models utilizing the CTViT-V image encoder. The training details are as follows: The image encoder was trained on Dataset-XY (train) for 6000 steps, divided into 4 batches, taking approximately 3.3 hours. For the pre-training phase, we conducted 5 epochs of single-batch training on Dataset-XY (train), lasting around 1.3 hours. In the fine-tuning phase, both LoRA and SFT models were fine-tuned on Dataset-XY (train) for 2 epochs with a single batch. LoRA training required approximately 1.4 hours, while SFT took about 3.6 hours, resulting in the 3D-CT-GPT++ LoRA and SFT versions. Lastly, the DPO phase, based on the SFT model, was trained using a preference dataset generated from Dataset-XY (test), taking approximately 2.4 hours and yielding the final version, 3D-CT-GPT++ (SFT+DPO).

### B.3.2 COMPARISON WITH EXISTING MODELS

In the *Comparison with Existing Models* experiment, we compared 3D-CT-GPT (baseline), 3D-CT-GPT++, RadFM, and M3D. For 3D-CT-GPT, based on the architecture from Chen et al. (2024a), we first trained the CTViT encoder using Dataset-XY (train) for 6,000 steps, followed by 5 epochs of pre-training and 2 epochs of LoRA fine-tuning, adhering strictly to the original 3D-CT-GPT model structure. For 3D-CT-GPT++, we used the model form described in Appendix B.3.1, specifically the 3D-CT-GPT++ (SFT+DPO) version as the final form. Although RadFM supports both 2D and 3D inputs, we were unable to fine-tune the model on our dataset due to time and resource constraints. Fine-tuning RadFM would require additional adaptations, as its data processing format differs from ours. Similarly, M3D, as a multi-task model, requires segmentation as part of its pipeline, which involves significant pre-processing and annotated segmentation data, exceeding the scope of this work. Furthermore, M3D’s data format is incompatible with ours, preventing direct fine-tuning. In this study, we chose to evaluate RadFM and M3D based on their published results in the literature. Future work may involve fine-tuning these models on our dataset for a more direct comparison.

### B.3.3 IMPACT OF TEMPERATURE ON PERFORMANCE

For the Impact of Temperature on Performance phase, we employed the 3D-CT-GPT++ (SFT+DPO) model form described in Appendix B.3.1. This version was selected as the final configuration of the 3D-CT-GPT++ model. We conducted experiments varying the temperature parameter from 0.4 to 0.9 to assess its effect on the model’s report generation capabilities.

### B.3.4 IMPACT OF DATA QUANTITY ON DPO PERFORMANCE

Both 3D-CT-GPT++ (1) and 3D-CT-GPT++ (SFT) followed the training setup outlined in Appendix B.3.1. The pre-training and fine-tuning phases for 3D-CT-GPT++ (2) were identical to 3D-CT-GPT++ (1). The key distinction lies in the DPO phase, where a refined data selection strategy (Strategy 2) was employed. In Strategy 2, the aim was to enhance contrast between



positive and negative examples by selecting the highest-scoring candidate as the positive example and the lowest-scoring candidate as the negative example. If no candidate scored  $\geq 3$ , the real report was used as the positive example. This strategy was designed to ensure that the selected examples better represented model preferences and human-aligned outputs, leading to more effective optimization during the DPO phase. DPO training for 3D-CT-GPT++ (2) was conducted under the same experimental configuration and batch settings as 3D-CT-GPT++ (1), ensuring consistency across all experiments. For details regarding Strategy 1, please refer to section 3.3.1, which provides an in-depth explanation of how GPT-4 was used for scoring and constructing the preference dataset.

## B.4 ABLATION STUDY

### B.4.1 ABLATION STUDY EXPERIMENTAL SETUP

In the ablation study conducted during the Direct Preference Optimization (DPO) phase, we followed the experimental setup described in Appendix B.3.1. This study aimed to assess the contributions of different components in the model, with specific results shown in Table 2. The experimental configurations are outlined as follows:

- **Baseline Setup (3D-CT-GPT++, CTViT):** The original 3D-CT-GPT (CTViT) model was trained on Dataset-XY (train), following the same procedure as described in B.3.2.
- **3D-CT-GPT++ (CT-RATE+LoRA):** Trained the encoder (CTViT-V) on the public dataset CT-RATE (train) for 4 batches and 10,000 steps, which took 13 hours. Pre-training was then conducted on CT-RATE (train) for 5 epochs with a single batch, taking 3.8 hours. Afterward, LoRA fine-tuning was performed on Dataset-XY (train), which took 1.4 hours.
- **LoRA Fine-tuning (3D-CT-GPT++):** We applied the LoRA-based fine-tuning method to the 3D-CT-GPT++ model. This involved training for 2 epochs on Dataset-XY (train), which took approximately 1.4 hours, followed by evaluation on Dataset-XY (test), as described in B.3.1.
- **Refined DPO Model (3D-CT-GPT++ DPO (2)):** The training setup for the refined DPO model (3D-CT-GPT++ DPO (2)) was the same as for 3D-CT-GPT++ (1), but with a more selective dataset for DPO training, following the strategy described in B.3.4 (Strategy 2).
- **Ablation Setting (b) - Unfreezing the Multi-Layer Perceptron (MLP):** In this configuration, we unfroze the MLP during the DPO phase. The model was trained using the same experimental setup as in Appendix B.3.1, but the MLP was unfrozen during the DPO phase. The dataset for the DPO phase was generated using Strategy 2.
- **Ablation Setting (c) - Adjusting the Learning Rate to  $3 \times 10^{-7}$ :** In this configuration, the learning rate was adjusted to  $3 \times 10^{-7}$  during the DPO phase. The dataset for DPO training was the same as in Strategy 2.

Each experiment was repeated five times, and the average results were reported to ensure robustness and reliability. All configurations were evaluated with consistent batch sizes and learning rate settings, using the Adam optimizer throughout the training process.

## C QUALITATIVE ANALYSIS

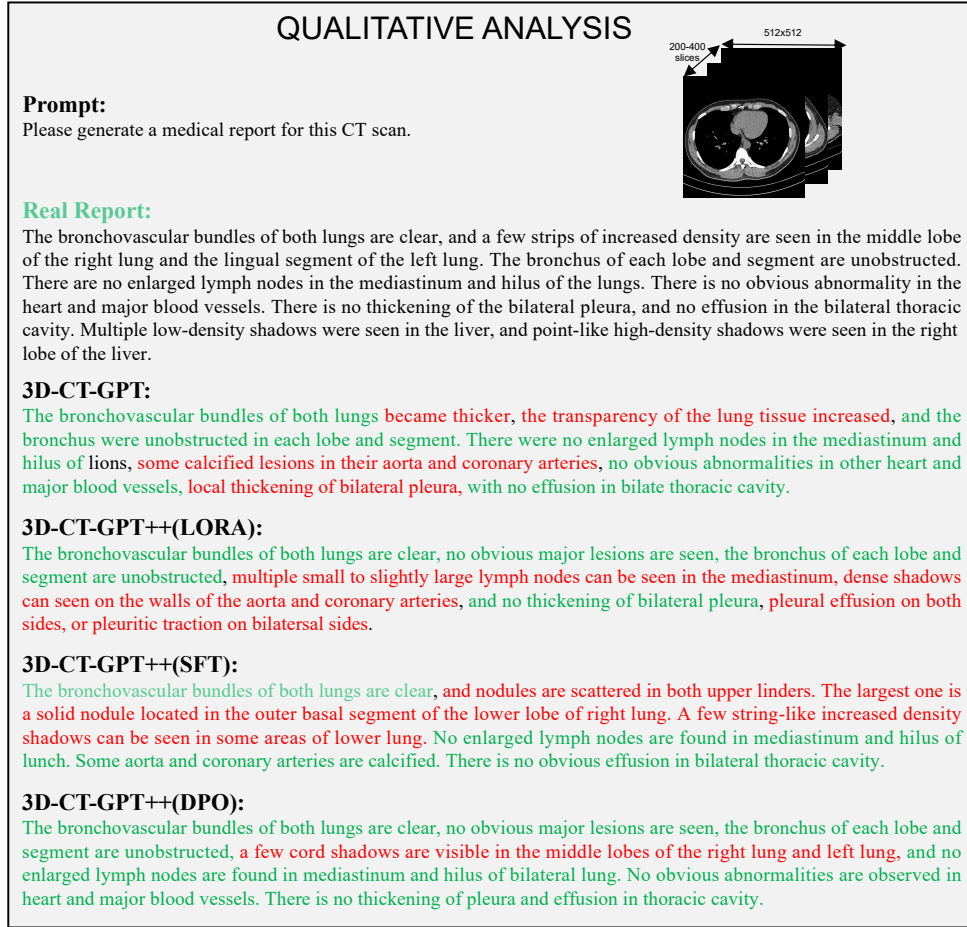


Figure 5: Comparison of generated reports from different versions of 3D-CT-GPT++ (DPO, SFT, LoRA, and baseline) with the real medical report. Correct answers and hallucinations are highlighted. Each model generated a report based on the same CT scan, highlighting differences in clinical accuracy, detail, and medical terminology. The real report serves as a reference for assessing alignment with actual clinical observations.