Effectively Fine-tune to Improve Large Multimodal Models for Radiology Report Generation

Yuzhe Lu[1]*, **Sungmin Hong**[2], **Yash Shah**[2], **Panpan Xu**[3]
[1]Carnegie Mellon University [2]AWS GAIIC [3]AWS AI
yuzhelu@cs.cmu.edu, {hsungmin, syash, xupanpan}@amazon.com

Abstract

Writing radiology reports from medical images requires a high level of domain expertise. It is time-consuming even for trained radiologists and can be error-prone for inexperienced radiologists. It would be appealing to automate this task by leveraging generative AI, which has shown drastic progress in vision and language understanding. In particular, Large Language Models (LLM) have demonstrated impressive capabilities recently and continued to set new state-of-the-art performance on almost all natural language tasks. While many have proposed architectures to combine vision models with LLMs for multimodal tasks, few have explored practical fine-tuning strategies. In this work, we proposed a simple yet effective two-stage fine-tuning protocol to align visual features to LLM's text embedding space as soft visual prompts. Our framework with OpenLLaMA-7B achieved state-of-the-art level performance without domain-specific pretraining. Moreover, we provide detailed analyses of soft visual prompts and attention mechanisms, shedding light on future research directions.

1 Introduction

With the rapid progress in deep learning, last few years have witnessed growing interests in leveraging deep generative language models [34, 8, 1, 26] for the task of radiology report generation (RRG), whose goal is to automatically generate medical reports for an X-ray. This task is challenging as it requires 1) understanding the X-ray image fully and 2) generating clinically accurate texts. Given the impressive capabilities of LLMs on various natural language tasks, we are interested in leveraging them for the task of RRG. However, utilizing LLMs imposes unique challenges due to their huge memory requirements. Previously, the dominant strategy to combine visual and text modalities has been to add additional cross-attention layers to the language model [34, 19]. This paradigm becomes infeasible for LLMs as training these additional layers demands tremendous GPU resources. As a result, many have explored the framework that uses a lightweight mapping network to project visual features to LLM's text embedding space as soft visual prompts to condition it with visual context [25, 9, 18, 33, 23]. Under this framework, a common practice has been to freeze pretrained vision and language models and only train the mapping network for downstream tasks [25] such as VQA [33]. In this work, we challenge this practice by showing that fine-tuning the vision model consistently improves performance for RRG. To avoid distorting pretrained visual features, we propose a two-stage fine-tuning strategy that firstly warms up the mapping network and demonstrate this simple strategy further improves the clinical efficacy of our model. Our contributions are summarized as below:

- We demonstrate the applicability of leveraging large language models via a lightweight network mapping visual features as soft prompts for radiology report generation.

*Work done during an internship at Amazon

1st Workshop on Deep Generative Models for Health at NeurIPS 2023.

- We propose a simple yet effective two-stage fine-tuning strategy that improves the factuality of generated radiology reports under this framework. We show that with this fine-tuning protocol, our model provides SOTA-level performance without domain-specific pretraining.
- We perform detailed analysis on the behavior of our model to reveal potential challenges when utilizing even larger language models for multimodal tasks and future directions.

2 Related Work

Radiology Report Generation Our research is closely related to works that use deep learning approaches to generate radiology reports from X-ray images. Many developed domain-specific methods to improve supervised fine-tuning [8, 1, 26], while others explored other training paradigms such as reinforcement learning [22, 10] so as to explicitly optimize for clinical efficacy metrics. In this paper, we focus on supervised fine-tuning and note that our method is compatible with reinforcement learning techniques, which are often performed in addition to supervised fine-tuning.

Large Multimodal Models The last few years have witnessed the rise of both uni- and multi-modal foundational models [4]. These models [5, 28, 32], pretrained on millions or even billions of data, demonstrate impressive zero-shot capabilities and can often be easily adapted for downstream tasks via fine-tuning. More recently, LLMs have shown increasingly astonishing generation capabilities as they scale up to billions of parameters. However, it is challenging for vision-language generative models to keep up with the scale as collecting paired image-text data for training is naturally much harder. Thus, many recent works have focused on combining pretrained vision models and large language models for multimodal generation tasks. In this work, we followed a popular framework of combing pretrained vision models and large language models by using a mapping network to project visual features to langauge model's text embedding space as soft visual prompts [25, 13, 23, 18].

3 Methods

In this section, we will cover the architectures, training objective, and our fine-tuning protocol to build Large Multimodal Models (LMM) from pretrained vision and large language models for RRG.

Mapping Network Our RRG model consists of three main components: a visual encoder, a mapping network, and a causal language model. The visual encoder takes in an X-ray and outputs features $\mathbf{I}$ of shape $L \times D_{img}$, where L is the number of visual tokens and D_{img} is the dimension of visual features. The mapping network we used is a single transformer decoder layer that takes on two roles, similar to [18]. The first is to perform attention pooling [17] on the visual sequence to learn a smaller number of tokens. This bottleneck structure is designed to extract features most relevant to the text modality and reduce the memory consumption of the autoregressive language model similar to [18]. In our experiments, we used 10 query vectors to produce 10 compact visual tokens from the original 225 visual tokens. The second is to project visual features of D_{img} to match the language model's text embedding dimension D_{txt}. Thus, the mapping network transforms visual features $\mathbf{I}$ of shape $L \times D_{img}$ to context $\mathbf{C}$ with N tokens embeddings of dimension D_{txt} that is subsequently used as soft prompts to the causal language model.

Fine-tuning Objective We fine-tune our LMM using the language modeling loss. Let θ denote the parameters of our model, and $Y = \{y_1, y_2, ...y_n\}$ denote the report. The objective can be written as:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{n} logp_\theta(y_t \mid \mathbf{C}, y_{i,i<t}) \tag{1}$$

where C is the projected visual context used to condition the language model's generation process.

As we scale to large language models with billions of parameters, it becomes challenging to fine-tune these models end-to-end. Thus, we leverage LoRA [14], a parameter-efficient fine-tuning method, to adapt large language models to the task of RRG. LoRA adds low-rank weight matrices to attention blocks and only trains these parameters when fine-tuning ($< 1\%$ of the original parameters).

Method	LM	LM Total	LM Tunable	2-Stage	MIMIC-CXR			
					BLEU4	ROUGE-L	F1-CXB-14	F1-CXB-5
BioVIL-T	CXRBERT	138M	138M	N/A	**6.9**	23.1	31.4	42.0
Ours	GPT2-S	117M	0.29M	✗	6.4	23.2	23.3	31.2
				✓	5.8	22.6	24.6	30.6
	GPT2-L	774M	1.47M	✗	6.3	23.2	26.8	37.7
				✓	6.5	**23.8**	28.4	39.6
	OpenLLaMA	7B	4.19M	✗	6.8	23.3	29.3	42.0
				✓	**6.9**	23.5	**32.0**	**42.2**

Table 1: We observe consistent performance gains when utilizing larger language models and the two-stage fine-tuning protocol. Our best-performing model with OpenLLaMA-7B achieves SOTA-level performance without extensive pretraining on the MIMIC-CXR dataset as in BioVIL-T [3].

Fine-tuning Vision Encoder Improves Performance While the common practice [25, 13, 23, 33] in recent works that combine visual and text foundational models is to freeze the visual encoder and fine-tune only the text decoder, we found fine-tuning the visual encoder together with the mapping network and text decoder consistently improves models' performance measured by clinical efficacy metrics (F1-CXB14 score), as shown in Table 1. Intuitively, fine-tuning the vision encoder allows it to extract features more related to the report. However, one natural concern over fine-tuning the vision encoder is catastrophic forgetting [16].

Two-Stage Fine-tuning Improves Performance Even Further Following the spirit of [16], we proposed a similar two-stage fine-tuning strategy to avoid distorting the pretrained visual features. It is rather simple and intuitive: instead of fine-tuning the vision encoder from the very beginning, we first freeze it for one epoch to let the mapping network learn to align the two different modalities; we then unfreeze the vision encoder for the remaining epochs. We allocate only one epoch for the first stage tuning because 1) the first epoch incurs the most loss and thus most gradient updates to the vision model, and 2) we don't want extra training time. Note that we ran the same number of epochs for both one-stage and two-stage fine-tuning experiments; also, the LoRA weights are tuned in both stages since they are also randomly initialized.

The intuition behind this two-stage protocol is that, since we have a randomly initialized mapping network, we could easily distort the visual features when we fine-tune it with the mapping network at the same time, particularly at the beginning of fine-tuning where the error of the mapping network leads to excessive updates to the vision encoder. [16] shares the same intuition but targets the image classification setting and requires fine-tuning on downstream datasets to convergence twice, first using only linear head and then using both linear head and vision encoder. This setting requires double the training time of vanilla fine-tuning. Moreover, when we experimented with it in our problem setting, we found that the two-stage protocol in [16] provides limited gains and can even hurt the performance (Appendix C). We posit that in our case, tuning the mapping network until convergence will overfit it to the language model's representations, which makes the subsequent adaptation of the vision encoder more challenging. Finally, we verify visual features after two-stage fine-tuning indeed preserve inter-class differences better than those after single-stage fine-tuning (Appendix B).

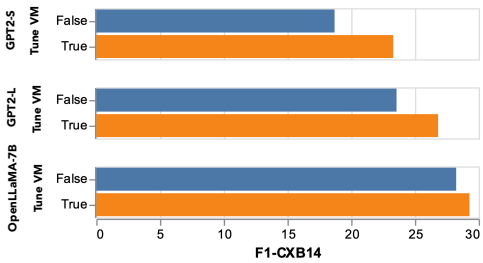

Figure 1: We observed that fine-tuning the visual encoder together with the remaining model components consistently improves F1-CXB14 score.

4 Experiments and Results

In this section, we cover our experimental setup and provide analysis our results.

Datasets We used the MIMIC-CXR dataset [15] following preprocessing from [11] . Since we aimed to generate the *findings* section of the report, we disregarded ones with a missing *findings* section. As most images have frontal views, we followed [2] and only used AP/PA views. In practice, radiologists frequently refer to previous images when drafting reports for the current X-ray. Training models using these reports without providing previous images encourages hallucinating a non-existent prior. To focus on describing the details in the image, we followed the spirit of [30] and curated a subset that doesn't use additional context other than the current image. Specifically, we used information from the *comparison* section from the radiology report and filtered out reports whose comparison section explicitly states previous sessions were referred to. After this step, we have curated a cross-sectional dataset consisting of 85,802 training, 682 validation, and 1259 test data, which is roughly half of the original dataset, and each comprising of a single image and report pair.

Vision and Language Models For the visual encoder, we utilized the same vision encoder E_{img} as [3], a pretrained ResNet50. We chose it since this model was pretrained on high-resolution (480×480) chest X-rays and presumably learned more discriminative representations. For the language model, we utilized three models of different scales, GPT2-S (117M), GPT2-L (774M) [29], and OpenLLaMA-7B (7B) [12]. We include the fine-tuning details in Appendix A.

Evaluation Metrics To evaluate the quality of generated radiology reports, we use both Natural Language Generation (NLG) metrics and Clinical Efficacy (CE) metrics. For NLG Metrics, we report BLEU4 [27] and ROUGE-L [21]. To evaluate the factual completeness and correctness of the generated report, we used the F1CheXBert score, which uses CheXBert [31], a deep learning based chest radiology labeler that outputs 14 relevant medical observations from a given report. F1CheXBert is essentially the F1 score between the output of CheXBert on generated report $\hat{y}$ and ground truth report y. To make the comparison to prior works easier, we also report F1CheXBert for the 5 most prevalent observations. We denote these metrics as F1-CXB-14 and F1-CXB-5 respectively.

Main Results In Table 1, we show our main experimental results. We observed consistent performance gains on both NLG and CE metrics as we scaled our language model from GPT2-S to OpenLLaMA-7B. Moreover, we found that our two-stage fine-tuning strategy effectively improved F1-CXB14 scores for all models, with an average improvement of 1.9 points. To contextualize our models' performance, we reproduced a SOTA method, BioVIL-T [3]. Despite BioVIL-T having access to 2x more data from MIMIC-CXR for task-specific pretraining, we found our best-performing model with OpenLLaMA-7B and two-stage fine-tuning manages to outperform it. We believe that our framework of leveraging off-the-shelf LMMs provides promising results for the task of radiology report generation. We provide a qualitative evaluation in Appendix D.

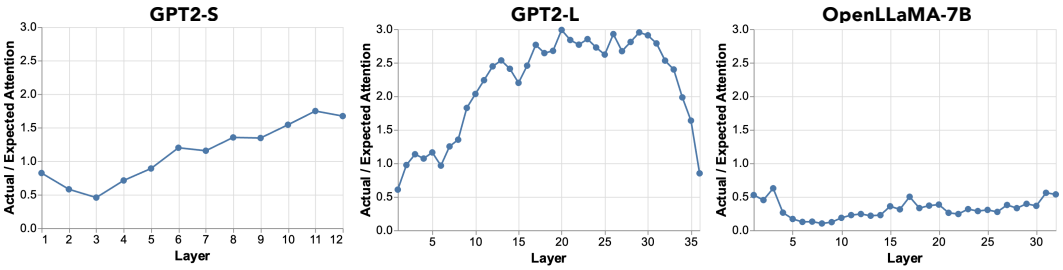

Figure 2: We visualize the average attention weights allocated to the visual prompts across all attention layers. One critical observation is that OpenLLaMA-7B gives much less attention to soft visual prompts than smaller GPT2 models. This suggests that OpenLLaMA-7B, and potentially larger language models in general, might be less grounded to visual information.

Attention Allocation While we are able to outperform BioVIL-T with OpenLLaMA-7B, we admit the performance gain when scaling from GPT2-L to OpenLLaMA-7B is smaller than we expected. Meanwhile, we observed OpenLLaMA-7B had a much lower validation loss than BioVIL-T, despite their similar test performances. We hypothesize OpenLLaMA-7B might overly rely on its language modeling ability and pay less attention to the soft visual prompts. To verify this hypothesis, we visualized the attention to the visual soft prompts for three models in Figure 2. Concretely, each newly generated token has an attention matrix O of shape $H \times L$ at each layer, where H is the

number of heads and L is the number of generated tokens. As the length of generated tokens grows, absolute attention weights allocated to the visual tokens naturally get diluted. Thus, we adjusted $O\prime$ using a multiplier L/K, where K is the length of visual prompts. Finally, we compute the amount of attention allocated to visual prompts in $O\prime$ by averaging across attention heads and test samples. Thus, an average attention of 1 means the visual prompt gets attention proportional to its length.

$$\alpha_i = \sum_{j=1}^{K} attention[i][j] \Big/ \frac{K}{K+i}, \quad i \in 1, ...n \tag{2}$$

Interestingly, we found that OpenLLaMA-7B pays much less attention to the visual prompt on average compared to GPT2-S and GPT2-L, validating our previous hypothesis. Intuitively, with its teacher-forcing scheme, the language modeling loss 1 can be too easy for LLMs with billions of parameters. Essentially, since LLM is too good at predicting the next word, it becomes hard to align visual tokens to LLM's text embedding space with the loss alone. In fact, this is also implicitly confirmed in Figure B, where we can see that the average pair-wise similarity between test samples is higher overall than GPT2-L (darker color means lower similarity). This serves as evidence that the language modeling loss did not effectively force the mapping network of OpenLLaMA to separate the visual features of samples from different classes.

Conclusion and Future Work In this paper, we utilized a lightweight framework to build large multimodal models from off-the-shelf vision and large language models for the task of RRG. To maximize the performance gains, we proposed two-stage fine-tuning to align visual features to LLM's embedding space as soft prompts and demonstrated this strategy consistently improves clinical accuracy for language models across different scales. By visualizing the attention weights, we found the soft visual prompt doesn't receive consistent attention, especially when using larger language models. This could make the generated report less grounded. In the future, we plan to investigate more sophisticated ways of incorporating visual features into LLMs. We think visual-conditioned prefixing tuning [20] can be a promising solution.

References

[1] Omar Alfarghaly, Rana Khaled, Abeer Elkorany, Maha Helal, and Aly Fahmy. Automated radiology report generation using conditioned transformers. *Informatics in Medicine Unlocked*, 24:100557, 2021.

[2] Shruthi Bannur, Stephanie Hyland, Qianchu Liu, Fernando Perez-Garcia, Maximilian Ilse, Daniel C Castro, Benedikt Boecking, Harshita Sharma, Kenza Bouzid, Anja Thieme, et al. Learning to exploit temporal structure for biomedical vision-language processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15016–15027, 2023.

[3] Benedikt Boecking, Naoto Usuyama, Shruthi Bannur, Daniel C Castro, Anton Schwaighofer, Stephanie Hyland, Maria Wetscherek, Tristan Naumann, Aditya Nori, Javier Alvarez-Valle, et al. Making the most of text semantics to improve biomedical vision–language processing. In *European conference on computer vision*, pages 1–21. Springer, 2022.

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Ángel Alexander Cabrera, Erica Fu, Donald Bertucci, Kenneth Holstein, Ameet Talwalkar, Jason I Hong, and Adam Perer. Zeno: An interactive framework for behavioral evaluation of machine learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2023.

[7] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[8] Zhihong Chen, Yan Song, Tsung-Hui Chang, and Xiang Wan. Generating radiology reports via memory-driven transformer. *arXiv preprint arXiv:2010.16056*, 2020.

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[10] Jean-Benoit Delbrouck, Pierre Chambon, Christian Bluethgen, Emily Tsai, Omar Almusa, and Curtis P Langlotz. Improving the factual correctness of radiology report generation with semantic rewards. *arXiv preprint arXiv:2210.12186*, 2022.

[11] Jean-benoit Delbrouck, Khaled Saab, Maya Varma, Sabri Eyuboglu, Pierre Chambon, Jared Dunnmon, Juan Zambrano, Akshay Chaudhari, and Curtis Langlotz. ViLMedic: a framework for research at the intersection of vision and language in medical AI. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 23–34, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[12] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023.

[13] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven CH Hoi. From images to textual prompts: Zero-shot vqa with frozen large language models. *arXiv preprint arXiv:2212.10846*, 2022.

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[15] Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317, 2019.

[16] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

[17] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[20] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[21] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[22] Guanxiong Liu, Tzu-Ming Harry Hsu, Matthew McDermott, Willie Boag, Wei-Hung Weng, Peter Szolovits, and Marzyeh Ghassemi. Clinically accurate chest x-ray report generation. In *Machine Learning for Healthcare Conference*, pages 249–269. PMLR, 2019.

[23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[26] Aaron Nicolson, Jason Dowling, and Bevan Koopman. Improving chest x-ray report generation by leveraging warm-starting. *arXiv preprint arXiv:2201.09405*, 2022.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[29] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[30] Vignav Ramesh, Nathan A Chi, and Pranav Rajpurkar. Improving radiology report generation systems by removing hallucinated references to non-existent priors. In *Machine Learning for Health*, pages 456–473. PMLR, 2022.

[31] Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Y Ng, and Matthew P Lungren. Chexbert: combining automatic labelers and expert annotations for accurate radiology report labeling using bert. *arXiv preprint arXiv:2004.09167*, 2020.

[32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[33] Tom van Sonsbeek, Mohammad Mahdi Derakhshani, Ivona Najdenkoska, Cees GM Snoek, and Marcel Worring. Open-ended medical visual question answering through prefix tuning of language models. *arXiv preprint arXiv:2303.05977*, 2023.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

A Implementation Details

We used AdamW optimizer [24] for supervised fine-tuning. We train our vision encoder and mapping network end-to-end while applying LoRA to our language model. Specifically, we added low-rank matrices of dimension 8 to the query Q and value V in each transformer block. We also used gradient-checkpointing [7] to further reduce memory consumption. With these techniques, we were able to use a batch size of 16 for all of our experiments. For learning rate, we experimented with $[2\text{e}{-5}, 5\text{e}{-5}, 1\text{e}{-4}]$ and selected the best learning rate by best F1-CXB-14 score on validation set. We found $1\text{e}{-4}$ works best for GPT2-S and $5\text{e}{-5}$ works best for GPT2-L and OpenLLaMA-7B when freezing the vision encoder and used the same learning rates for other experiments with the same language model. We trained all models for 50 epochs. We used the checkpoint with the best validation F1-CXB-14 score for testing. In terms of hardware, we were able to fine-tune the OpenLlama-7B model on a single 24G GPU instance while also fully fine-tuning our vision encoder.

B Soft Visual Prompts

To verify our intuition that two-stage fine-tuning helps to preserve pretrained visual features, we propose to analyze the soft visual prompts of input images. Our idea is that if pretrained features are preserved better, difference between positive sample pairs' similarity scores and negative sample pairs' similarity scores should be larger on average, where we define positive pairs as samples with the same labels, and negative pairs as samples with different labels.

Each visual prompt consists of K visual tokens of dimension D_{text}. To compute similarity between visual prompts, we flatten it into a long vector of dimension $K \times D_{text}$. To quantify the difference between positive pairs and negative pairs on average, we propose to use the following metric:

$$\Delta = \frac{\frac{1}{|P|} \sum_{i,j \in P} cos(F_i, F_j)}{\frac{1}{|N|} \sum_{p,q \in N} cos(F_p, F_q)} - 1 \tag{3}$$

where P denotes the set of positive sample pairs while N denotes the set of negative sample pairs.

We present our analysis in Fig B. Each subplot is a cosine similarity matrix of all test samples' soft visual prompts. We sort test samples by their labels so that clusters of positive pairs are better visualized. We make two observations from these plots. Firstly, tuning vision model makes the visual prompts more distinguishable in general, as evidenced the smaller cosine similarity scores on average. Secondly, two-stage fine-tuning preserves the similarity between positive pairs better, as evidenced by the consistently higher Δ value of two-stage fine-tuning than vanilla fine-tuning in Fig B.

C Additional Results on Two-Stage Fine-tuning

In addition to our proposed two-stage fine-tuning setting, we also experimented with the setting where we fine-tune the model for the full amount of epochs for each stage, similar to [16] and showed the results in Table 2. We observed mixed results between the two in terms of clinical efficacy scores. We think our proposed setting is more desirable since it doesn't require additional training epochs.

D Qualitative Results

We provide several examples of generated reports from our OpenLLaMA model in Fig. D, using Zeno [6].

E Confidence

Additionally, we analyzed the confidence scores of generated reports, which is less explored in previous works. Since a common issue of LLMs is hallucination, we investigate if the model is aware of hallucination (defined as false positive cases) by looking at its confidence scores for true positive

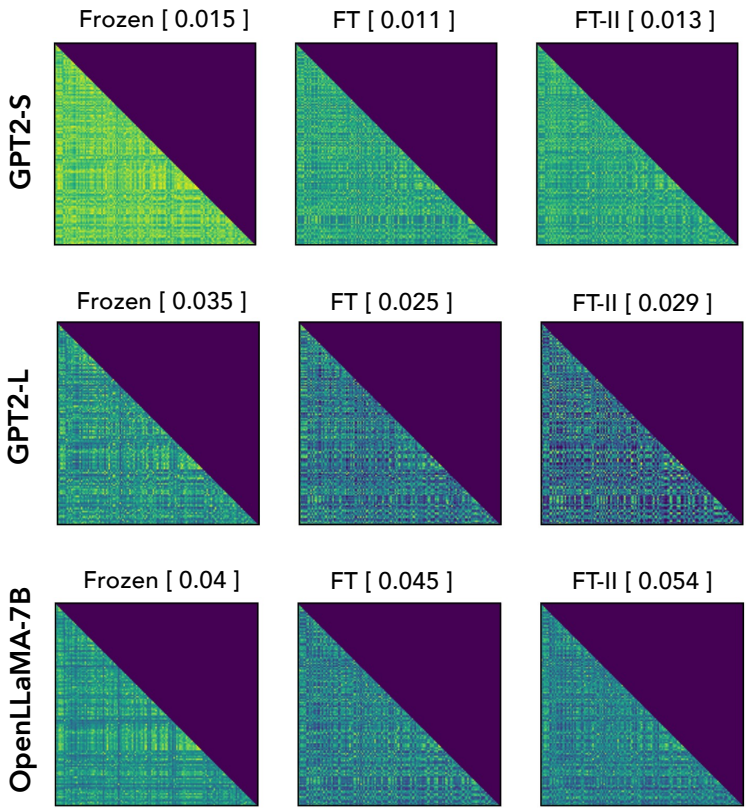

Figure 3: Pairwise cosine similarity between test samples' visual soft prompts. Test samples are sorted based on their ground truth labels. The number in square bracket denotes the relative difference between mean cosine similarity between positive pairs and that of negative pairs as detailed in Eq. 3. Brighter color denotes higher cosine similarity. Here, Frozen means vision model is not tuned, FT denotes vanilla fine-tuning, and FT-II denotes two-stage fine-tuning.

Method	LM	LM Total	LM Tunable	Stage-1 Epochs	MIMIC-CXR			
					BLEU4	ROUGE-L	F1-CXB-14	F1-CXB-5
Ours	GPT2-S	117M	0.29M	1	5.8	22.6	24.6	30.6
				50	5.9	23.1	24.1	34.5
	GPT2-L	774M	1.47M	1	6.5	23.8	28.4	39.6
				50	7.1	24.0	28.8	38.1
	OpenLLaMA	7B	4.19M	1	6.9	23.5	32.0	42.2
				50	6.41	23.0	30.5	42.3

Table 2: We show the results for two slightly different experimental settings of two-stage fine-tuning.

and false positive cases respectively. We provide a starting point for analyzing model confidence distribution by considering the average confidence *AC* of a generated report $Y = \{y_1, y_2, ...y_n\}$, defined as the following:

$$AC(\mathbf{Y} \mid \theta) = \frac{1}{n} \sum_{t=1}^{n} p_\theta(y_t \mid \mathbf{C}, y_{i,i<t}) \tag{4}$$

where θ is the model parameter and $\mathbf{C}$ is the visual context. In Figure E, we provide a visualization of average confidence distribution grouped by classes for all three of our models. We found that the AC distribution of true positives and false positives tend to have overlapping supports. This unfortunately makes it extremely challenging to signal potential mistakes to the end user.

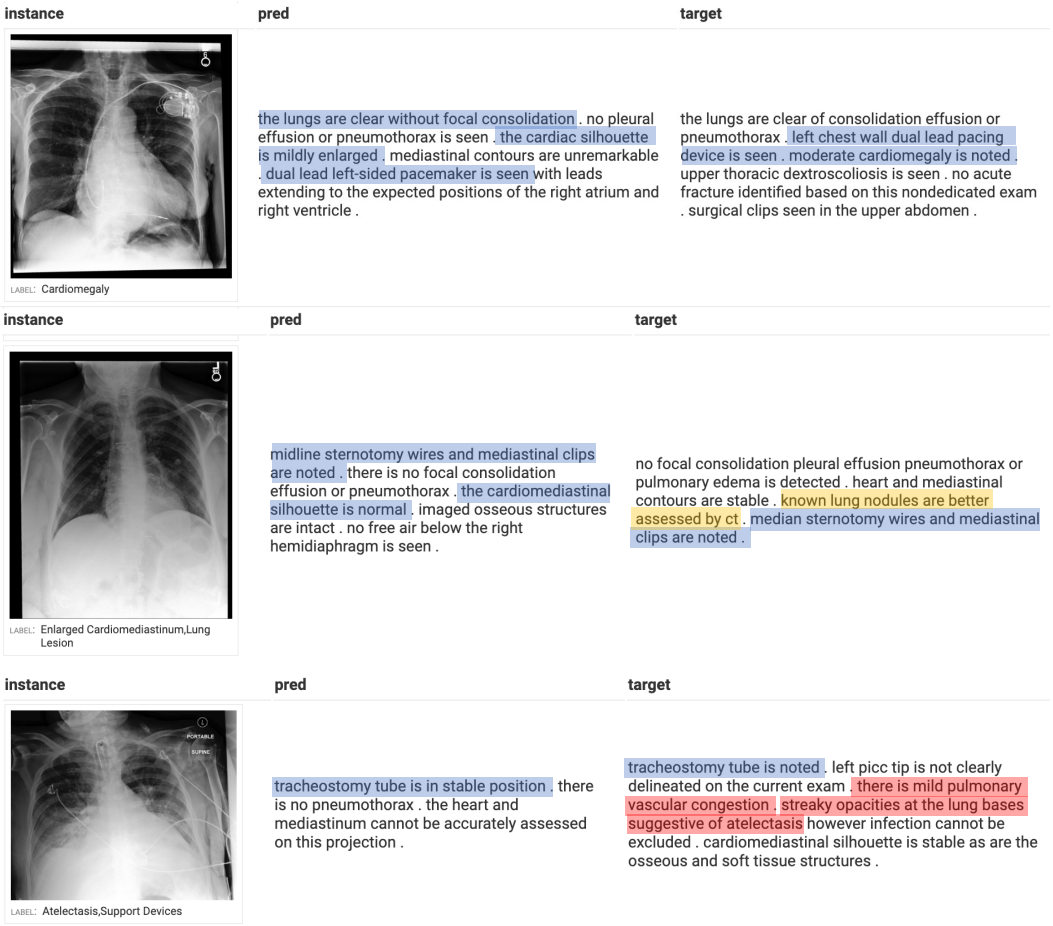

Figure 4: The first row showcases a case where the model accurately identifies cardiomegaly (enlarged heart). The second row demonstrates an interesting case where a medical observation is mentioned according to outside knowledge but not well observed in the current X-ray, which renders the miss acceptable. The third row shows a failure case where our model fails to capture critical medical observations.

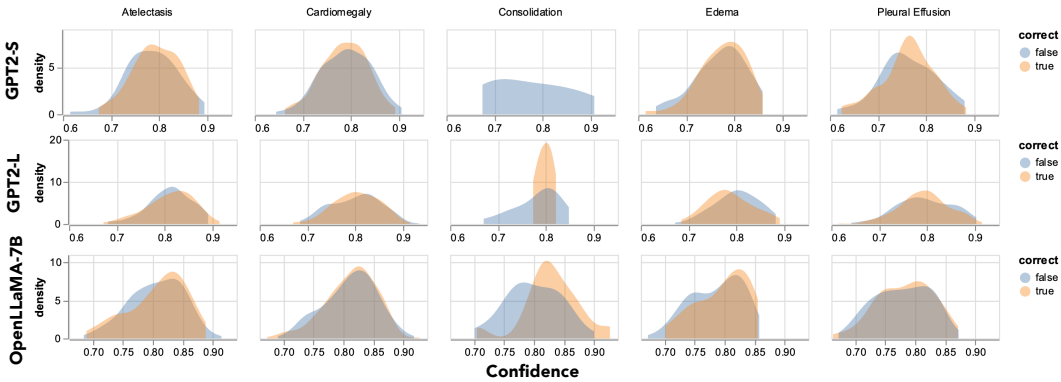

Figure 5: Distribution of confidence scores for true and positive cases.