Team B Paper

Michael Villarreal
EECS Department
University of Tennessee
Knoxville, U.S.
tvillarr@vols.utk.edu

Dylan Lewis
EECS Department
University of Tennessee
Knoxville, U.S.
dlewis37@vols.utk.edu

Lu Liu
EECS Department
University of Tennessee
Knoxville, U.S.
lliu58@vols.utk.edu

Sujit Kumar Tripathy
EECS Department
University of Tennessee
Knoxville, U.S.
stripat3@vols.utk.edu

Fred Yu
EECS Department
University of Tennessee
Knoxville, U.S.
zyu20@uthsc.edu

***Index Terms*—Automated Prompt-Based Image Segmentation, LVLM, Medical, Off-the-Shelf Models**

I. Introduction

Using machine learning models to build computer aided diagnosis (CAD) systems has been a goal of researchers since the recent explosion of computer vision algorithms. CAD systems help physicians examine medical images much faster than they are able to without using such systems. When physicians can examine scans faster, patients do not have to wait as long to get results, and more patients can be seen.

Using computer vision algorithms on medical images presents several challenges that are unique to doing machine learning on medical images. One major challenge is that medical images do not necessarily contain a clear view of objects that physicians may want to focus on. For example, when looking at a chest X-ray, a physician may want to look for abnormalities around the patient's heart, but the heart may be partially covered by ribs, lungs, or bones in the image. Machine learning algorithms can have a very hard time recognizing objects that are only partially visible in an image, which happens quite often in the medical space.

One way to make this less of a problem is to segment the medical images instead of trying to place bounding boxes over all objects in the image. When segmenting the image, you are trying to find masks that remove everything but a certain object from the image. When doing segmentation, masks can be found for these partially visible anatomical structures without needing the entire object to be visible.

Developing CAD systems has been a popular area of research for several years now, but one relatively unexplored area is developing CAD systems in which the doctor can ask a model questions about what a scan contains. This type of model would need a way to understand both images and text, and the relationship between these 2 modalities.

Such a model is known as a large language vision model (LLVM). These models are able to identify objects images using text, combining the advancements of natural language processing (NLP) and computer vision (CV) into a single model.

Companies such as Google, OpenAI, and Meta have spent millions of dollars to train extremely large machine learning models to perform both natural language processing and computer vision tasks. These models the weights found via intensive training are often made publicly available for anyone to use.

Our goal in this paper is to propose a model that combines these publicly available models for NLP and CV into a single LLVM that is capable of doing medical image segmentation, and using text prompts to identify which part of the image the user is asking about.

II. Related Work

A. Vision-Language Models

The introduction of Contrastive Language-Image Pre-training (CLIP) [15] has caused an explosion in vision-language models (VLMs). CLIP trains both an image encoder and text encoder by aligning their latent representation spaces together, connecting images to their associated text. To perform such training, large datasets of (image, text) pairs have been created [1], [15], [17], [18] to allow for better aligned embedding spaces.

Meta AI's Segment Anything Model (SAM) [9] can use bounding boxes, point coordinates, or even text prompts to create masks of objects in images. SAM can also generate masks for an entire image using SAM's segment anything mode. SAM 2 [16], SAM's successor, maintains the same functionality as SAM; however, SAM 2 provides drastic speedups and performance increases to the original model. These models provided previously unseen capabilities in the vision-language space. While CLIP was able to establish a semantic relationship between images and text, SAM established a relationship between text and the spatial information of individual objects in an image.

VLMs have seen significant attention in the general-purpose domain; however, they have also gained popularity in medical image segmentation. This is due to the common nature of medical images being paired with a corresponding medical report, making this a prime space for VLMs. Both CLIP- and SAM-based models in the medical space have been developed with MedCLIP [22] in 2021 and MedSAM [13] in 2024. Additionally, MedSAM has seen a successor in MedSAM2 [23], which utilizes SAM2, in 2024.

Identify applicable funding agency here. If none, delete this.

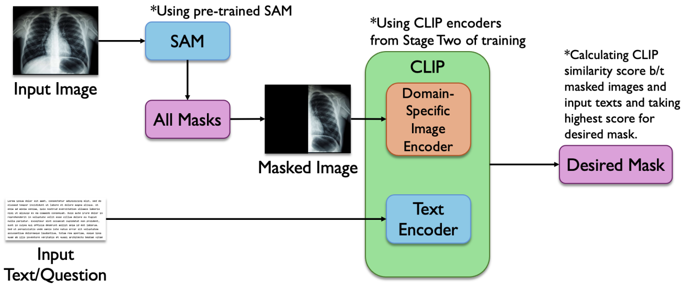
*Using pre-trained SAM
SAM
Input Image
All Masks
Masked Image
*Using CLIP encoders from Stage Two of training
CLIP
Domain-Specific Image Encoder
Text Encoder
*Calculating CLIP similarity score b/t masked images and input texts and taking highest score for desired mask.
Desired Mask
Input Text/Question

Fig. 1: The Base Pipeline of our LVLM for automated image segmentation. Our LVLM receives as input an image and a text prompt. The LVLM then uses an internal SAM model (in segment everything mode) to generate a set of masks from the image. These masks are then fed to an internal CLIP model that selects the desired mask based on the input text.

B. gScoreCAM.

While VLMs have made significant progress in linking images and text, their behavior can be unpredictable, making their decision-making processes difficult to interpret. These models often struggle with object localization in complex, multi-object scenes. For example, CLIP can be misled by simple changes in an image, raising concerns about how it processes information internally. A key advancement addressing these issues is gScoreCAM, a method that sheds light on what the CLIP model focuses on in an image [2]. By analyzing only the most critical features (the top 10 % of high-gradient channels), gScoreCAM reduces computational demands while enhancing accuracy. It performs exceptionally well in complex scenes, reliably identifying multiple objects and fine details, making it a faster and more effective tool for understanding CLIP's visual processing.

Building on the concept of object localization, MedCLIP-SAM framework uses the strengths of CLIP and SAM to improve medical image segmentation [10]. It integrates gScoreCAM's ability to pinpoint important regions in images with SAM's advanced segmentation capabilities to generate accurate segmentation masks based on text prompts. Further refining these results with weakly supervised learning, MedCLIP-SAM demonstrates impressive performance across medical imaging modalities like ultrasound, MRI, and X-ray. These advancements illustrate how methods like gScoreCAM, originally developed for general object localization, can be adapted to solve critical challenges in specialized fields like medical imaging.

III. Methodology

A. Base Pipeline

This work focuses on the task of automatic image segmentation, which involves extracting meaningful regions or segments from image inputs with minimal human intervention. Our goal is to develop a Large Vision-Language Model (LVLM) that is a flexible, multi-modal reasoning system using off-the-shelf models, capable of adapting beyond predefined label vocabularies, thereby allowing for greater flexibility and adaptability. Our LVLM consists of two pipelines: a base pipeline and a gScoreCAM [2] pipeline (see Sec. III-C. These pipelines are designed to balance efficiency and performance. The base pipeline is the main pipeline of our LVLM and is presented in Fig. 1. The base pipeline consists of two core components: a mask generation module (through SAM or SAM-based models) and a mask selection module (through CLIP or CLIP-based models). We first use SAM in the "segment everything" mode to generate a comprehensive set of candidate masks for a given input image. The generated masks are then fed into an image encoder to produce image embeddings, while the corresponding text inputs are processed by a text encoder to create text embeddings. Mask selection is performed by aligning these embeddings in the latent space through a CLIP-like model. Similarity scores are calculated between the input masks and text prompts, and the mask with the highest similarity score is selected as the desired mask.

This approach draws inspiration from the recent work of LLM-Seg [21], which demonstrates promising reasoning capabilities in image segmentation. LLM-Seg employs a three-stage process: first, mask proposals are generated through a mask generation module; second, embeddings for each mask proposal are extracted; and finally, a fusion module aligns each mask embedding with the corresponding text input embedding using attention mechanisms. In contrast, our pipeline simplifies this process by replacing the fusion module in LLM-Seg with a CLIP module for mask selection. Additionally, LLM-Seg requires a specially collected dataset to train LLM-Seg; however, our approach only requires standard (image, text) data pairs.

The ability to use off-the-shelf models allows us to test our LVLM in general-purpose settings (considering that most pre-trained CLIP and SAM models use general-purpose datasets); however, we also consider domain-specificity. In this work, we focus on the medical domain as an additional testbed for our LVLM's performance. Specifically, we explore using medical domain CLIP and SAM models in our LVLM through models such as MedCLIP [22] and MedSAM [13]. However, these models are trained on datasets that consist of a variety of medical images (chest X-rays, MRIs, CT scans, etc.). Thus, we include an additional component in our methodology of instilling specific domain knowledge into a CLIP model (we focus on CLIP, rather than SAM, as domain-specific training datasets are easier for CLIP) to evaluate any performance differences.

B. Training a Domain-Specific CLIP

We consider the medical domain for our domain-specific CLIP model and train the CLIP model (which requires both images and text) using the MIMIC-CXR-JPG [8] dataset. The MIMIC-CXR-JPG dataset is a derivation of the MIMIC-CXR dataset [7] that contains $377,110$ chest X-rays in JPG format instead of the original's DICOM format. Additionally, MIMIC-CXR-JPG includes labels (classes), which are derived from processing MIMIC-CXR's text reports for each chest X-ray; however, we use the text reports instead of the labels

Hyperparameter	Autoencoder	CLIP
Epochs	10	10
Learning Rate	1e-3	1e-3
Batch Size	128	128
Loss	MSE	CLIP
Optimizer	Adam	Adam

TABLE I: Autoencoder and CLIP Training Hyperparameters

during training. There are $227,827$ text reports; there is not a unique text report for each chest X-ray as some X-rays are different angles of the same patient with the same written observations.

For training a domain-specific CLIP model, we first decide on which component of CLIP to focus on. There are two major models in a CLIP model: the image encoder and the text encoder. For this work, we decide to only train/finetune the image encoder given most text encoders are trained on a large corpus of text and contain vast knowledge stored in their weights. Given this vast knowledge, we feel that training the text encoder would only hurt performance and would be reinventing the wheel; thus, we decide to only train the image encoder.

However, there are multiple ways to train the image encoder. Should we initialize the image encoder's weights from scratch and then train the CLIP model with a medical domain dataset? Would initializing the image encoder's weights and then training a CLIP model result in unstable training? Should we initialize the image encoder's weights with weights trained on another objective, and then train the CLIP model? Should we pretrain the image encoder's weights on medical images and then train a CLIP model? To gain insights into these questions, we conducted a series of ablation training experiments.

First, we decide to pretrain the image encoder on medical images using autoencoder training. Autoencoder training, if the training works, results in the encoder used during training to well represent the data being passed through it. This means the embedding space at the encoder's output also well represents the medical data. As CLIP training's goal is to align the embedding spaces of the image and text encoders, we believe autoencoder training will result in a good starting embedding space before further aligning the embedding space with text through CLIP training. However, do we initialize the image encoder's weights from scratch and then perform autoencoder training or do we transfer weights from another pretraining task and finetune the weights with medical images from autoencoder training? In our experiments, we explore both options. We indicate that by prefixing the names of our models in Sec. IV-A with either a FS (for From Scratch initialization weights) or FP (for Finetune Pretrained where the weights come from a model trained another task; in our case, the models were trained to classify images from ImageNet [3]).

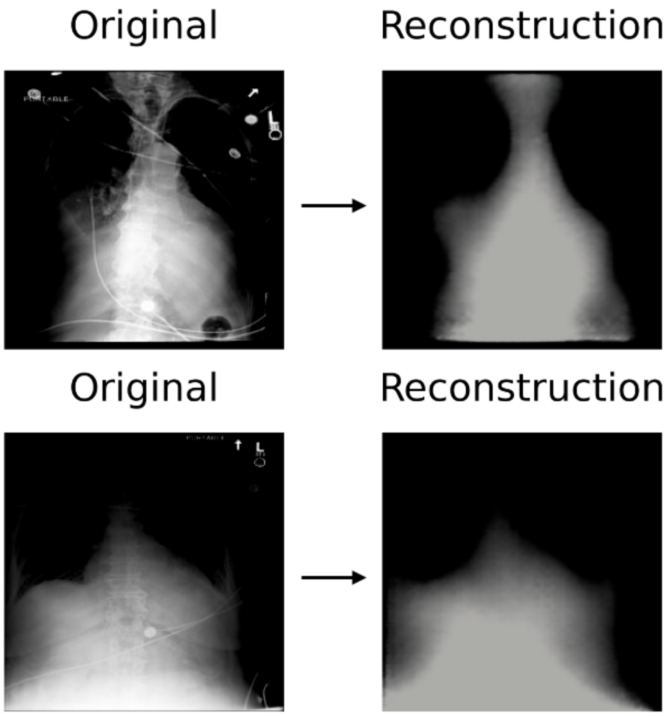

Fig. 2: Example reconstructions of MIMIC-CXR-JPG images on an autoencoder-trained ResNet-50 [5] image encoder with from scratch weights.

The hyperparameters we use during autoencoder training is given in Table I. Given autoencoder training is unsupervised, only the JPG-format chest X-rays of MIMIC-CXR-JPG are passed through the image encoder and decoder. We present example reconstructions from a trained ResNet-50 [5] image encoder on the MIMIC-CXR-JPG images in Fig. 2. The weights of the ResNet-50 were initialized from scratch. We observe from the example reconstructions that while the image encoder is able to reconstruct the general shape of the original image, there is significant loss in finer details.

Additionally, we trained seven CLIP models with different training settings to explore the best way to train the CLIP model, while also verifying our choices to use autoencoder training before CLIP training and to not train the text encoder as well. Specific details of the seven proposed CLIP training settings are given in Table II along with their training outcomes measured in the validation losses decrease from beginning to the end of the training cycle.

We determined "Setting 2" as our best CLIP-trained baseline LVLM to be carried over to the evaluation trials downstream, based on the results from our preliminary testing of performing CLIP trainings under settings 1-7, as shown in the "Outcome" column in Table II and more specifically in Fig. 3 and in Fig. 4. We note that through settings 1-5, the image encoders are based on the Swin-T [12] backbone, among which the pipeline with setting 2 achieved the highest validation reduction of 0.052. This observation may have implied that auto-encoder pretraining on CXR images makes an impact on improving the model's performance in matching domain-specific textual and visual cues. One shall also notice that although setting 6 has the best CLIP training outcome

Settings	Training Components	Outcome
1	(Image/Text) Projection heads	0.042
2	Proj. heads and image encoder	0.052
3	Proj. heads and both (image/text) encoders	0.0
4	Proj. heads and image encoder (“From Scratch”)	0.0
5	Proj. heads and image encoder (ImageNet-1K)	0.0
6*	Proj. heads and image encoder (“From Scratch”)	0.118
7*	Proj. heads and image encoder (ImageNet-1K)	0.023

TABLE II: CLIP Training Settings and Outcomes: The table demonstrates seven different training settings proposed for the final base pipeline LVLM. Training outcomes corresponding to different settings, defined as the **reduction in validation loss** from epoch 1 to 10, are listed along the third column (rounded to the 3rd decimal place).

Note: For settings with “*”, the image encoder has the ResNet-50 [5] architecture. Image encoders in all other settings have the same architecture as Swin-T [12]. CLIP trainings under settings 1-3 all started with image encoders from Auto-encoder pretraining (on MIMIC-CXR). “From Scratch” suggests that the image encoder weights were from random initialization at the start of the CLIP training. For completeness of the ablation study, we should have done all trainings as setting 1-5 (Swin-ViT backbone) with a ResNet-50 backbone, which leaves the space for future work.

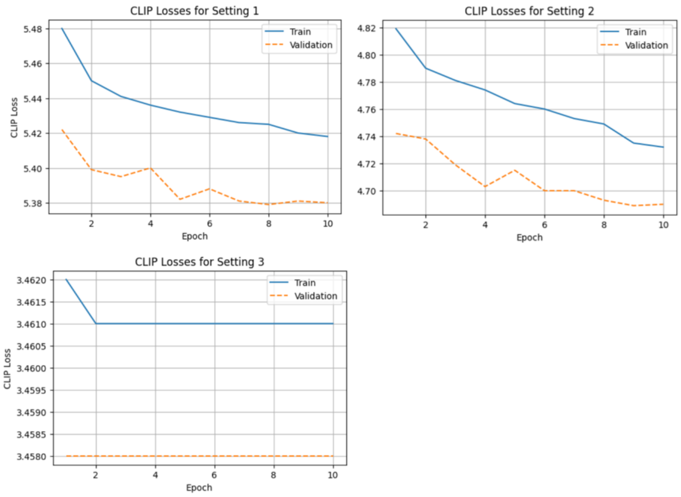

Fig. 3: Loss curves of the proposed settings 1-3 (with image encoders from autoencoder pretraining): Curves each shows the training (blue) and validation (orange, dashed) losses during the entire training schedule.

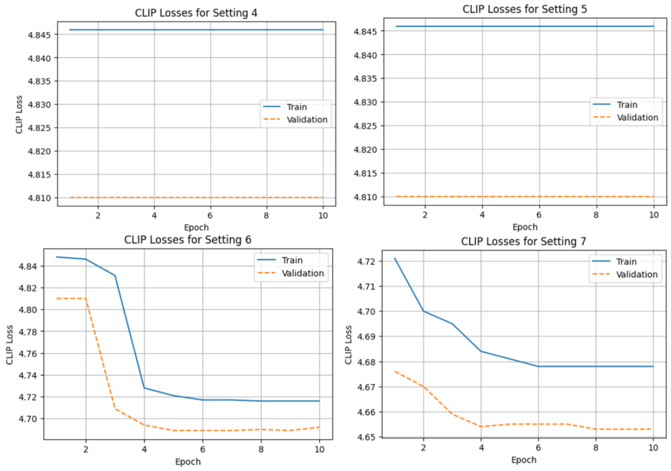

Fig. 4: Loss curves of the proposed settings 4-7 (with image encoders “from scratch” or from ImageNet-1K pretraining): Curves each shows the training (blue) and validation (orange, dashed) losses during the entire training schedule.

of a 0.118 validation loss reduction among all seven training experiments, the fact that image encoder for setting 6 is from the ResNet-50 [5] backbone, which is different from the previous settings 1-5 (all with a Swin-T backbone) makes the competition unfair between setting 2 and setting 6 (and hence is not sufficient to disvalue the importance of domain-specific autoencoder pretraining). For future work, we aim to include the experiments of CLIP training settings 1-5 (especially setting 2) on the image encoder from the ResNet-50 backbone to complete the ablation study.

C. *gScoreCAM-Based Pipeline*

The base inference pipeline described in the previous subsection utilizes SAM to generate multiple possible masks in “segment everything” mode, followed by the CLIP objective, which processes embeddings from the fine-tuned image encoder and text encoder prompts to calculate similarity (CLIP) scores. The mask with the highest similarity score is selected. However, as discussed in [2] and [10], this approach can lead to suboptimal segmentation performance due to challenges in accurately computing similarity scores with the CLIP objective.

This prompted us to explore a gScoreCAM-based pipeline, as illustrated in figure 5. This approach is adapted from the zero-shot segmentation framework proposed in the MedCLIP-SAM paper [10]. Unlike the base pipeline, which directly depends on SAM-generated masks and CLIP similarity scores, the gScoreCAM-based method incorporates an additional step to enhance the mask generation process.

In this improved pipeline, gScoreCAM generates saliency maps from the CLIP image encoder layers, highlighting regions of interest based on the alignment between text prompts and image features. These saliency maps are then refined using Conditional Random Fields (CRF) to create bounding boxes that serve as input prompts for SAM [19]. By providing more precise and contextually relevant prompts, this method substantially enhances the quality of the resulting segmentation masks, addressing the limitations of relying exclusively on CLIP similarity scores.

IV. Experiments and Results

We outline the experiment setup for our pipeline in Sec. IV-A with details such as the test datasets (Sec. IV-A1), evaluation metric (Sec. IV-A2), and the selected models for

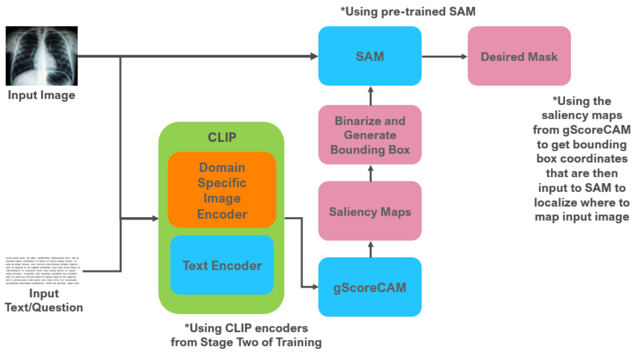
*Using pre-trained SAM
Input Image
SAM
Desired Mask
Binarize and Generate Bounding Box
*Using the saliency maps from gScoreCAM to get bounding box coordinates that are then input to SAM to localize where to map input image
CLIP
Domain Specific Image Encoder
Text Encoder
Saliency Maps
gScoreCAM
Input Text/Question
*Using CLIP encoders from Stage Two of Training

Fig. 5: gScoreCAM-based inference pipeline

evaluating the effectiveness of our pipeline (Sec. IV-A3). We provide results for all selected models on the test datasets in Sec. IV-B. Both pipelines use the same settings for the test datasets and evaluation metric, but slightly differ on selected models (see Sec. IV-A3 for details).

A. *Experiment Setup*

1) Test Datasets: We evaluate the effectiveness of our pipeline in two settings: a general-purpose setting and a domain-specific setting focusing on the medical domain.

General-Purpose Setting. We use a hand-picked subset of the COCO [11] dataset (referred to as COCO) as every image contains common objects pretrained models will most likely be familiar with (depending on their pretraining dataset). Using COCO `val2017`, we select 100 total images belonging to one of four human-perceived difficulty levels (with 25 images per level): easy, medium, hard, and very hard. The difficulty levels represent how easy/hard a human believes selecting the correct, desired mask from the image should be for our LVLM. For example, an image with only one object in front of a simple (i.e. one color) background gets placed in easy, while an image with many different objects against a busy (i.e. many colors or unclear transitions between background objects) background gets placed in hard or very hard. COCO images tend to have multiple, labeled objects; however, for our testing, we select one object per image. We use the selected object's class as the text input prompt to our LVLM and compare the mask returned by our LVLM to the ground-truth mask for the selected object from COCO.

Domain-Specific Medical Setting. We use a randomly sampled subset of a chest X-ray dataset [14] (referred to as CXR). This dataset contains chest X-rays with segmentation masks for the lungs. Matching our COCO test dataset, we select 100 chest X-rays to evaluate our LVLM on. To select the 100 X-rays, we randomly sample 100 from the provided test split. We refrain from placing the X-rays into difficulty levels, similar to our COCO test set, due to a lack of medical expertise to make such judgment calls. We simply use "lungs" as the text input prompt to our LVLM and compare our LVLM's selected mask to the ground-truth mask.

Model Name	Image Backbone	Pretraining Dataset
OA-RN50	ResNet-50 [5]	OpenAI [15]
OA-RN101	ResNet-101 [5]	OpenAI
OA-ViT-B	ViT-B-16 [4]	OpenAI
OA-ViT-L	ViT-L-14 [4]	OpenAI
CC-RN50	ResNet-50	cc12m [1]
LA-ViT-B	ViT-B-16	laion400m [17]
MC-RN50	ResNet-50	MedCLIP [22]
MC-SwinT	Swin-T [12]	MedCLIP
BC-ViT-B	ViT-B-16	TreeOfLife-10m [18]
FS-RN50 (ours)	ResNet-50	MIMIC-CXR-JPG [8]
FP-RN50 (ours)	ResNet-50	MIMIC-CXR-JPG
FS-ViT-B (ours)	ViT-B-16	MIMIC-CXR-JPG
FP-ViT-B (ours)	ViT-B-16	MIMIC-CXR-JPG

TABLE III: CLIP models used to evaluate our LVLM.

2) Evaluation Metric: For each image, we compare the intersection over union (IoU) of our LVLM's returned mask to the ground-truth mask of the selected object. For the COCO test dataset, we report the mean IoU (mIoU) across all 100 images and the mIoU for each difficulty level. For the CXR test set, we report mIoU across all 100 chest X-rays.

3) Selected Models: We select a variety of pretrained models, which serve the purpose of evaluating the effectiveness of our LVLM but also act as baselines for our trained domain-specific image encoders. There are two major models in our LVLM: the segmentation model and the CLIP model.

Segmentation Models. For the segmentation models, we select models in the SAM [9] family. This is because our base pipeline requires use of SAM's segment anything mode and their proven superior performance. For the base pipeline, we evaluate our LVLM with SAM and SAM 2 [16]. Even though SAM 2 is overall better at segmentation than SAM [16], we consider both to compare their performances in the specific setting of providing masks to a CLIP model to be selected. While SAM and SAM 2 are trained to segment whole images, we want to evaluate if segment anything mode's generated masks line up with text-prompt-desired masks.

For the gScoreCAM pipeline (which uses bounding box prompts for segmentation), we evaluate our LVLM on SAM 2 and MedSAM [13]. We do not use MedSAM with the base pipeline as MedSAM does not have a segment anything mode. Additionally, we drop SAM as MedSAM is a medical-specific version of SAM. By choosing SAM 2 and MedSAM we can compare the performance of a general-purpose model and domain-specific model on both evaluation datasets.

CLIP Models. We evaluate our LVLM swapping in a variety of off-the-shelf CLIP models provided by the OpenCLIP framework [6] (this technically includes our domain-specific CLIP models) or the MedCLIP framework [22]. We provide a full list of selected CLIP models in Table III. This is a comprehensive list for both pipelines; however, not every CLIP model is evaluated on for the base and gScoreCAM pipelines. We provide a name for each model based on the CLIP image encoder backbone and pretraining dataset; these names are used to succinctly refer to the models when discussing their performance results in Sec. IV-B. We evaluate our pipelines using six general-purpose CLIP models (OA-RN50, OA-RN101, OA-ViT-B, OA-ViT-L, CC-RN50, and LA-ViT-B) and seven domain-specific CLIP models (MC-RN50, MC-SwinT, BC-ViT-B, FS-RN50, FP-RN50, FS-ViT-B, and FP-ViT-B) where both kinds of CLIP models are tested on general-purpose and domain-specific test datasets (see Sec IV-B for details). The FS-RN50, FP-RN50, FS-ViT-B, and FP-ViT-B CLIP models contain the image encoder backbones we train on the MIMIC-CXR-JPG [8] dataset (see Sec. III for details). All models except for MC-RN50 and MC-SwinT, which come from MedCLIP, originate from OpenCLIP.

We primarily evaluate on ResNet-50 [5] and ViT-B-16 [4] models given their moderate size allows them to be trained in a single GPU, they have proven performance on various computer vision tasks [4], [5], and are representatives of their respective model families. For the general-purpose CLIP models, we focus on models trained using OpenAI's dataset [15] given their work is quintessential to the CLIP family. However, we consider models trained on other general-purpose datasets, Conceptual 12m (cc12m) [1] and Laion400m (laoin400m) [17], and larger models, ResNet-101 and ViT-L-14, to evaluate any performance differences. For the domain-specific CLIP models (aside from our four domain-specific CLIP models), we use two MedCLIP [22] models: one with a ResNet-50 image backbone and one with a Swin-T [12] image backbone. Regarding a transformer-based [20] backbone, MedCLIP only provides a Swin-T model so the performance comparisons are not exact to the ViT-B or ViT-L CLIP models. The MedCLIP models only work in our base pipeline due to the strict architecture requirements of gScoreCAM [2] so we include BioCLIP [18] as an additional domain-specific model for testig the gScoreCAM pipeline (we also test BioCLIP in the base pipeline for completeness). BioCLIP considers biology, rather than medicine, so BioCLIP's domain is only medicine-adjacent so we may not see any performance increases compared to the general-purpose models when testing on the CXR dataset.

Given that our focus is on the image encoder of a CLIP model, we simply use the default text encoder for each CLIP model (provided either by OpenCLIP or MedCLIP). As we use the default, we do not list the text encoder in Table III.

B. Results

We separate and discuss the results of our LVLM based on the two pipelines: the base pipeline and the gScoreCAM pipeline. We include visual examples of good and bad predictions on the base pipeline and gScoreCAM pipeline (with discussion) in Fig. 6.

1) Base Pipeline: Table IV presents results on our COCO dataset. All values are mIoU scores for the specific category where All is the mIoU over every image, and E (Easy), M (Medium), H (Hard), and VH (Very Hard) are the mIoUs over the specific difficulty category. The best overall model is OA-RN101, which achieves the highest mIoU over all images in our COCO test set but also achieves the highest mIoU for three of the four difficulty categories. Interestingly enough, OA-RN101 performs better with SAM compared to SAM 2, despite claims that SAM 2 is a better overall model [16]. The CLIP models pretrained on OpenAI's dataset generally perform better than the other general-purpose models, CC-RN50 and LA-ViT-B.

The domain-specific models (including our trained domain-specific models) perform significantly worse compared to the general-purpose models. This could be because COCO is a general-purpose dataset and the domain-specific models have lost the world knowledge (or general-purpose knowledge) to perform better at selecting the mask. Overall, our trained domain-specific CLIP models outperform the other domain-specific models (MC-RN50, MC-SwinT, and BC-ViT-B); however, our trained models achieve very poor performance compared to the general-purpose models. Our FP-RN50 model performs the best out of the four, giving a nod to start with pretrained weights using transfer learning.

Table V presents results on our CXR dataset. This dataset is not split into multiple categories so only one value is presented, which is mIoU over all images in the CXR test set. We observe an interesting pattern regarding the results: the mIoU scores swap back and forth from ∼0.27 to ∼0.16 if the segmentation model is SAM or SAM 2, respectively. This surfaces a major issue with the base pipeline of our LVLM: the performance hinges entirely on the performance of SAM. Given SAM is being used to generate the masks that are then selected based on the input text, if SAM fails to generate good masks, then our internal CLIP has no ability to make up SAM's failures, leading the CLIP model to fail as well. Given the mIoU values stay consistent across most CLIP models (except for five of the last six), we believe the results are more of an evaluation of the SAM model rather than our LVLM as a whole.

These results do however allow us to gauge the ability of the internal CLIP models, specifically, which ones are failing. MC-SwinT, BC-ViT-B, FS-RN50, FP-RN50, and FP-ViT-B all fail to reach that ∼0.27 or ∼0.16 benchmark for SAM or SAM 2, respectively. This gives us some insight that these CLIP models are failing to select the correct mask. The only CLIP models to fail are the domain-specific ones, which is an unfortunate result as we expected these models to perform better compared to the general-purpose models on the CXR dataset. Additionally, our FP-RN50 and FP-ViT-B models perform terribly (achieving 0 mIoU scores for both SAM and SAM 2). This is a fascinating result considering FP-RN50 is our best performing model on the COCO dataset, which means FP-RN50 understands more about general objects than about the kinds of images it has been trained on; this is a counter-intuitive result and should be further investigated.

In Fig. 7, we give a visual example of a high performing test case of the base pipeline on the CXR dataset where this example receives an IoU score of 0.48. We observe that SAM segments the lungs separately, where in this example SAM segments out the left lung individually and misses the right

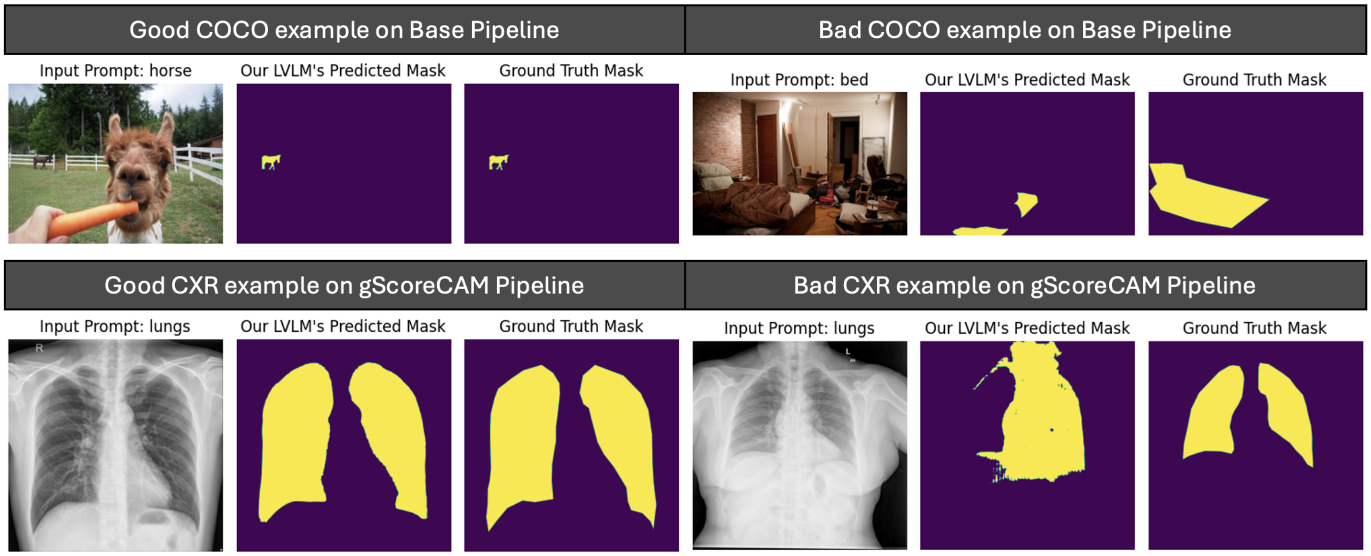
Good COCO example on Base Pipeline
Input Prompt: horse
Our LVLM's Predicted Mask
Ground Truth Mask
Bad COCO example on Base Pipeline
Input Prompt: bed
Our LVLM's Predicted Mask
Ground Truth Mask
Good CXR example on gScoreCAM Pipeline
Input Prompt: lungs
Our LVLM's Predicted Mask
Ground Truth Mask
Bad CXR example on gScoreCAM Pipeline
Input Prompt: lungs
Our LVLM's Predicted Mask
Ground Truth Mask

Fig. 6: Visual examples of our LVLM's predicted masks for good and bad test cases on the base pipeline and gScoreCAM pipeline. The good and bad test cases for both pipelines are on the left and right, respectively, while the base pipeline is the top row of images. We provide the original image and text prompt, predicted mask, and ground truth mask (in that order) side-by-side. For all cases, the CLIP model has a ResNet-50 image encoder pretrained on OpenAI's dataset [15]. For the base pipeline, SAM is the segmentation model. MedSAM is used for the gScoreCAM pipeline. In the good COCO example, our LVLM is able to segment and return the hourse in the background very well achieving an IoU score of 0.93 on this test case. In the bad COCO example, our LVLM captures only the exposed corners of the bed (presumably due to the high contrast to the surroundings) and fails to capture the entire bed. For the bad COCO example, our LVLM achieves only an IoU of 0.06. In the good CXR example, our LVLM is able to return a mask very close to the ground truth mask (with an IoU of 0.93). Despite the lungs being visually separate, our LVLM returns masks for both lungs. In the bad CXR example, our LVLM returns a mask that covers the right lung, but fails to capture the left lung and includes the spinal cord; this mask achieves an IoU of 0.24. The bad CXR image has lung cavities with less contrast to the surrounding areas compared to the good CXR image, which we believe partly explains the poor performance.

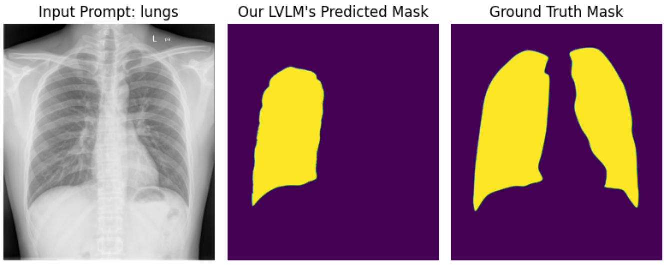

Fig. 7: A test example of the base pipeline on the CXR dataset that achieves a higher IoU score (0.48) compared to the other CXR test images. We observe a pitfall of using SAM in segment anything mode where SAM can return separate masks for the individual lungs (with provided example, SAM outputs a mask for the left lung and fails to segment the right lung entirely). This inhibits the abilities of our LVLM to perform well as the CLIP model has no way of rectifying this issue.

lung entirely (not shown in the image, but through our own testing). SAM returning a mask with only one of the lungs naturally leads to performance decreases in the final output as the CLIP model has no way to rectify the issue and can only select the mask with a single lung. We observe this behavior in other high scoring examples where SAM returns masks that separate the lungs; this can be explained given the spinal cord in the chest X-rays producing a natural border that SAM could pick up on. We observe in the good examples for the gScoreCAM pipeline (see Fig. 6 for details), gScoreCAM is able to overcome this pitfall.

2) gScoreCAM Pipeline: Table VII shows the results on the COCO dataset. For ResNet-based models, the performance was similar to the base pipeline. However, for ViT models, the gScoreCAM pipeline performed worse. This could be because, as noted in the *gScoreCAM* paper [2], the method is more suited for CNN architectures. It creates saliency maps by combining channels from the last convolutional layer before the fully connected layer. Although gScoreCAM was adapted to use information from the attention layer in ViT models, the authors suggest that CAM-based methods like HilaCAM might work better for these models.

Table VIII presents the results on the CXR dataset. The mean IOU scores improved significantly, almost doubling, for most of the models, including variants of both ResNet and ViT. This improvement is likely due to gScoreCAM's ability to localize objects more effectively in complex, multi-object

Model	SAM	All (↑)	E (↑)	M (↑)	H (↑)	VH (↑)
OA-RN50	SAM	0.26	0.51	0.24	0.14	0.22
	SAM 2	0.25	0.54	0.21	0.11	0.16
OA-ViT-B	SAM	0.32	0.50	0.26	0.23	**0.33**
	SAM 2	0.26	0.62	0.18	0.18	0.11
OA-RN101	SAM	**0.34**	0.48	**0.31**	**0.29**	**0.33**
	SAM 2	0.28	0.51	0.22	0.18	0.27
OA-ViT-L	SAM	0.33	**0.65**	0.27	0.18	0.27
	SAM 2	0.28	0.55	0.22	0.21	0.18
CC-RN50	SAM	0.23	0.55	0.12	0.18	0.12
	SAM 2	0.24	0.58	0.18	0.14	0.11
LA-ViT-B	SAM	0.30	0.57	0.30	0.11	0.27
	SAM 2	0.27	0.56	0.25	0.23	0.09
MC-RN50	SAM	0.04	0.08	0.04	0.0	0.05
	SAM 2	0.04	0.09	0.04	0.0	0.04
MC-SwinT	SAM	0.01	0.02	0.0	0.01	0.0
	SAM 2	0.04	0.15	0.0	0.0	0.01
BC-ViT-B	SAM	0.02	0.06	0.0	0.04	0.01
	SAM 2	0.05	0.16	0.0	0.04	0.01
FS-RN50 (ours)	SAM	0.03	0.10	0.01	0.0	0.0
	SAM 2	0.05	0.13	0.08	0.01	0.01
FP-RN50 (ours)	SAM	0.08	0.23	0.11	0.0	0.02
	SAM 2	0.10	0.27	0.07	0.04	0.06
FS-ViT-B (ours)	SAM	0.07	0.08	0.04	0.10	0.06
	SAM 2	0.09	0.16	0.10	0.08	0.02
FP-ViT-B (ours)	SAM	0.05	0.15	0.01	0.03	0.01
	SAM 2	0.05	0.20	0.00	0.00	0.01

TABLE IV: Results on COCO dataset.

Model	SAM	All (↑)
OA-RN50	SAM	**0.28**
	SAM 2	0.16
OA-ViT-B	SAM	**0.28**
	SAM 2	0.16
OA-RN101	SAM	0.27
	SAM 2	0.16
OA-ViT-L	SAM	**0.28**
	SAM 2	0.16
CC-RN50	SAM	0.27
	SAM 2	0.16
LA-ViT-B	SAM	0.27
	SAM 2	0.16
MC-RN50	SAM	0.27
	SAM 2	0.16
MC-SwinT	SAM	0.01
	SAM 2	0.01
BC-ViT-B	SAM	0.02
	SAM 2	0.0
FS-RN50 (ours)	SAM	0.12
	SAM 2	0.07
FP-RN50 (ours)	SAM	0.0
	SAM 2	0.0
FS-ViT-B (ours)	SAM	0.26
	SAM 2	0.15
FP-ViT-B (ours)	SAM	0.0
	SAM 2	0.0

TABLE V: Results on CXR dataset.

TABLE VI: Performance results, measured in mean Intersection over Union (mIoU), for various SAM and CLIP model combinations on the Base Pipeline. **Bold text** indicates best model performance in a particular category, while blue text indicates the best performance for our trained domain-specific models.

scenes. This also resulted in generating better prompts, such as bounding boxes or points on highlighted contours, which helped SAM2 create more accurate segmentation masks.

The pipeline with LA-ViT-B and BC-ViT-B models, fine-tuned on the LAION-400M and Tree-of-Life datasets, performed the worst on both the COCO and CXR datasets. Given that BC-ViT-B is only loosely connected to medical domains, its performance on the CXR dataset was expected to be worse.

V. Conclusion

In this project, we developed an LVLM that uses off-the-shelf SAM and CLIP models to perform the task of prompt-based automated image segmentation. We test our LVLM in general-purpose and domain-specific, where we consider the medical domain, settings. Our LVLM has a base pipeline where SAM, in segment anything mode, generates masks for CLIP to select one of those masks based on the input prompot. Our LVLM also has a gScoreCAM pipeline, which uses gScoreCAM (a technique to visualize what the CLIP gradients are looking at based on the input image and text) to obtain bounding boxes for SAM to generate a mask, which is then output by our LVLM. Based on our testing, we find our LVLM has significant room for improvement given the low IoU scores obtained for a variety of SAM and CLIP models. Additionally, we train our own domain-specific CLIP models through a domain-specific image encoder and evaluate our CLIP models in the LVLM. We find our CLIP models overall perform worse compared to other pretrained CLIP models. We hypothesize the difference in perform is due to the lack of training data given the other pretrained CLIP models are trained on significantly larger datasets; however, more testing is required to verify that claim.

A. *Next Steps*

A major area that we would change is how we went about testing our various models. For example, we selected CLIP Setting 2 using the training and validation loss curves. Generally speaking, this is not a best practice as the loss curves do not tell the full story. Instead, we would change how we selected the CLIP setting by actually testing the CLIP model on some example cases to see how our model performs end-to-end. This would allow us to compare end performance metrics, instead of just loss values. Additionally, we would include more visualizations within our testing. Raw numbers are nice to have, but only looking at IoU scores also does not tell the full story. We would visualize the selected masks selected by the various testing models to gain a better understanding of what the models are capturing.

A vital component of our LVLM (particularly in the base pipeline) is the SAM model; however, we did not do much

Model	SAM	All (↑)	E (↑)	M (↑)	H (↑)	VH (↑)
OA-RN50	MedSAM	0.24	0.49	0.11	0.12	**0.27**
	SAM 2	0.22	0.53	0.05	0.11	0.23
OA-ViT-B	MedSAM	0.02	0.04	0.01	0.02	0.0
	SAM 2	0.02	0.04	0.02	0.03	0.01
OA-RN101	MedSAM	0.24	0.47	0.12	0.15	0.24
	SAM 2	**0.30**	**0.62**	**0.20**	**0.17**	0.26
OA-ViT-L	MedSAM	0.03	0.05	0.01	0.03	0.02
	SAM 2	0.03	0.05	0.0	0.03	0.03
CC-RN50	MedSAM	0.09	0.24	0.06	0.02	0.05
	SAM 2	0.09	0.28	0.08	0.0	0.02
LA-ViT-B	MedSAM	0.03	0.07	0.0	0.03	0.01
	SAM 2	0.04	0.10	0.03	0.03	0.01
BC-ViT-B	MedSAM	0.03	0.07	0.0	0.0	0.06
	SAM 2	0.04	0.09	0.0	0.0	0.06
FS-RN50 (ours)	MedSAM	0.07	0.19	0.02	0.03	0.06
	SAM 2	0.07	0.23	0.01	0.03	0.02
FP-RN50 (ours)	MedSAM	0.04	0.11	0.03	0.01	0.04
	SAM 2	0.04	0.14	0.01	0.0	0.01
FS-ViT-B (ours)	MedSAM	0.01	0.03	0.0	0.01	0.0
	SAM 2	0.01	0.05	0.0	0.01	0.0
FP-ViT-B (ours)	MedSAM	0.05	0.13	0.01	0.02	0.04
	SAM 2	0.05	0.11	0.0	0.03	0.04

TABLE VII: Results on COCO dataset.

Model	SAM	All (↑)
OA-RN50	MedSAM	0.54
	SAM 2	0.49
OA-ViT-B	MedSAM	0.38
	SAM 2	0.39
OA-RN101	MedSAM	0.49
	SAM 2	0.45
OA-ViT-L	MedSAM	0.45
	SAM 2	0.40
CC-RN50	MedSAM	**0.57**
	SAM 2	0.42
LA-ViT-B	MedSAM	0.01
	SAM 2	0.02
BC-ViT-B	MedSAM	0.07
	SAM 2	0.11
FS-RN50 (ours)	MedSAM	0.28
	SAM 2	0.23
FP-RN50 (ours)	MedSAM	0.11
	SAM 2	0.07
FS-ViT-B (ours)	MedSAM	0.04
	SAM 2	0.05
FP-ViT-B (ours)	MedSAM	0.12
	SAM 2	0.12

TABLE VIII: Results on CXR dataset.

TABLE IX: Performance results, measured in mean Intersection over Union (mIoU), for various SAM and CLIP model combinations on the gScoreCAM Pipeline. **Bold text** indicates best model performance in a particular category, while blue text indicates the best performance for our trained domain-specific models.

testing on the SAM model. There are various parameters you can give the SAM model, which affects how it generates masks in segment anything mode. We believe doing a parameter search for SAM and SAM 2 (there does not appear to be any parameters we can change for MedSAM) through a grid-based search could improve our LVLM performance. We believe testing SAM/2 in this way could be useful as a particular issue we noticed during end-to-end evaluation is SAM/2 would often give us a decent number of masks (¿30); however, a number of these masks would be tiny. We want to investigate if obtaining larger masks from SAM/2 results in better performance and masks more in-line with expectations from the text prompt.

There are several smaller areas we would like to investigate as well. One of those is the autoencoder training. We used autoencoder training to position the image encoder's embedding space at a good starting point; however, we should verify this decision using performance results rather than loss curves like we did. After verifying with performance results and if there is merit to pretraining with an autoencoder objective, we could also investigate pretraining with more sophisticated techniques such as with diffusion models. Another area is when selecting the masks with the CLIP model. Our current methodology involves giving the CLIP model masked images, where the images are just the mask from SAM; this typically results in a mostly black image. Could the significant number of black pixels throw off the CLIP model? Thus, we would like to investigate if cropping said image to just the remaining relevant pixels might increase performance.

Similarly, in the 'gScoreCAM' pipeline, we anticipated strong performance from both ResNet and ViT models, as demonstrated in the *gScoreCAM* paper. However, we found that 'gScoreCAM' might not be the most suitable CAM-based method for use with ViT models. A potential future direction could involve incorporating HilaCAM, a CAM-based approach designed specifically for ViT architectures, into the inference pipeline to assess performance with ViT models.

Additionally, the results from the gScoreCAM pipeline were found to be highly sensitive to the Gaussian filter settings used to generate bounding box prompts for SAM2. Due to limited time and computational resources, we were unable to experiment with different filter settings. Future work could explore how varying these settings impacts pipeline performance.

References

[1] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[2] Peijie Chen, Qi Li, Saad Biaz, Trung Bui, and Anh Nguyen. gscorecam: What objects is clip looking at? In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2022.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pages 770–778, 2016.

[6] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below.

[7] 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.

[8] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[10] Taha Koleilat, Hojat Asgariandehkordi, Hassan Rivaz, and Yiming Xiao. Medclip-sam: Bridging text and image towards universal medical image segmentation. *arXiv preprint arXiv:2403.20253*, 2024.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[13] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024.

[14] Nikhil Pandey. Chest xrays masks and labels. Accessed 12-22-2024.

[15] 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.

[16] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[17] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[18] Samuel Stevens, Jiaman Wu, Matthew J Thompson, Elizabeth G Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila M Dahdul, Charles Stewart, Tanya Berger-Wolf, et al. Bioclip: A vision foundation model for the tree of life. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19412–19424, 2024.

[19] Charles Sutton and Andrew McCallum. An introduction to conditional random fields, 2010.

[20] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[21] Junchi Wang and Lei Ke. Llm-seg: Bridging image segmentation and large language model reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1765–1774, 2024.

[22] Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. Medclip: Contrastive learning from unpaired medical images and text. *arXiv preprint arXiv:2210.10163*, 2022.

[23] Jiayuan Zhu, Yunli Qi, and Junde Wu. Medical sam 2: Segment medical images as video via segment anything model 2. *arXiv preprint arXiv:2408.00874*, 2024.