# Enhancing Segment Anything Model (SAM) for Brain Tumor Image Segmentation

Komei Ryu
Department of Computer Science
Stanford University
komeiryu@stanford.edu

Yi Jing
Department of Statistics
Stanford University
jingi@stanford.edu

June Zheng
Institute for Computational and Mathematical Engineering
Stanford University
yujunz@stanford.edu

## Abstract

*The* ***Segment Anything Model (SAM)*** *[10] is a powerful prompt-able image segmentation model that allows for zero-shot performance transfer. We propose a pipeline that automatically segments brain tumors within MRI images. Given the prompt-able nature of SAM, we utilize* ***Vision Language Models (VLMs)*** *to automate the process of bounding box prompt generation to help SAM focus on specific regions in MRI images. We fine-tune SAM on the 2024 Brain Tumor Segmentation (BraTS) Challenge dataset [4] and perform data augmentation to enhance its robustness against imperfect bounding box prompts. Our contribution is an automated pipeline for brain tumor segmentation that starts with a VLM for bounding box generation and ends with a fine-tuned SAM model for mask prediction. The performance of this pipeline surpasses that of a state-of-the-art model, demonstrating the validity of our pipeline and its contribution. By automating and enhancing this segmentation process, our proposed pipeline could save time and cost for both patients and physicians in real-life scenarios.*

## 1. Introduction

The Segment Anything Model (SAM) [10] is a promptable image segmentation foundation model that allows for zero-shot performance transfer. It supports three types of prompts to help SAM identify objects of interest in an image: points, boxes, and masks. Meanwhile, recent advancements in foundation Vision Language Models (VLMs) have sparked growing interest in applying them to a wide range of downstream and domain-specific tasks.

Automatic brain image image segmentation is an important task with significant real-life implications, for it can help physicians and neurosurgeons diagnose the existence of brain tumors, classify tumor types, and even identify regions to perform surgeries on. An automatic segmentation technique can save ample time for physicians and reduce the monetary costs of patients who suffer from brain tumors. The cost-effectiveness of automatic image segmentation also allows patients to monitor the development of brain tumors by periodically observing changes in sizes and shapes of brain tumors with the help of automatic image segmentation.

We propose a fully automated inference pipeline that utilizes the power of large pre-trained foundation segmentation models and VLMs to automate brain tumor segmentation. Specifically, the pipeline involves feeding a brain MRI image into a VLM or object detection model and asking it to generate a predicted bounding box around the brain tumor region; then, the pipeline uses the MRI image as input and the bounding box as prompt to ask SAM, fine-tuned using augmented data, to predict a brain tumor segmentation mask. Our experiments found that the performance of our proposed pipeline surpasses that of a state-of-the-art medical image segmentation model, highlighting our contribution to the field of brain tumor segmentation.

## 2. Related Work

Brain tumor segmentation is a long-standing challenge in medical image analysis, with numerous methods developed ranging from traditional image processing to recent deep learning approaches. We categorize the related work into

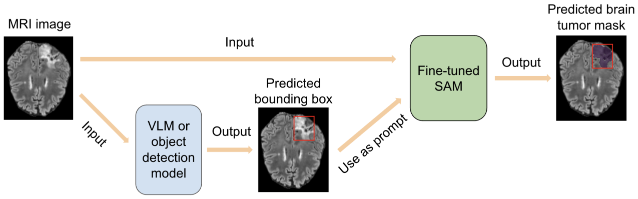


Figure 1. Proposed inference pipeline.

four main groups: (1) classical and atlas-based methods, (2) CNN-based segmentation models, (3) transformer-based segmentation models, and (4) prompt-based and vision-language-guided segmentation.

**Classical and Atlas-Based Methods** Early work on brain tumor segmentation often relied on intensity thresholding, region growing, and atlas-based approaches. Methods like the one proposed by Prastawa et al. (2003) employed statistical atlases to detect abnormalities as outliers from normal brain anatomy [16]. While interpretable, these techniques often struggled with tumor variability in shape and intensity. Similarly, the use of fuzzy clustering methods [22] was sensitive to initialization and noise.

**CNN-Based Segmentation Models** The advent of deep learning brought significant improvements. The U-Net architecture [19] became the de facto baseline due to its encoder-decoder structure and skip connections. Variants like nnU-Net [8] further automated architecture tuning and preprocessing, achieving state-of-the-art results on BraTS benchmarks. DeepMedic [9] explored 3D CNNs and multi-scale processing to better capture context. However, CNN-based models are typically trained in a fully supervised manner, requiring pixel-level labels, and lack flexibility in inference-time guidance.

**Transformer-Based Segmentation Models** Transformers have recently gained traction in medical imaging for their ability to model long-range dependencies. TransUNet [2] and Swin-Unet [1] integrate transformer blocks with CNNs to balance local and global features. These models outperform traditional CNNs in certain tasks, especially when spatial relationships are important. However, they are still fully supervised and require extensive labeled data to generalize well to new domains such as brain MRIs.

**Prompt-Based and Vision-Language-Guided Segmentation** Prompt-based segmentation offers a new paradigm by allowing models to adapt at inference time using guiding inputs like points, boxes, or masks. The Segment Anything Model (SAM) [10] exemplifies this approach, achieving strong generalization on natural images through training on over a billion masks. Despite impressive zero-shot generalization to natural images, its performance on domain-specific data like medical imaging is limited without adaptation. To address this, several studies have explored fine-tuning or adapting SAM for medical domains. MedSAM [14] fine-tunes SAM on CT and MRI data, demonstrating significant improvements in medical segmentation tasks. Zhang et al. [23] provide a comprehensive review of SAM's applications in medical imaging, outlining both its current potential and the challenges in transferring its capabilities to specialized domains. Methods like SAM-Adapter [3] introduce lightweight task-specific adapters that condition SAM on additional domain-specific information, yielding improvements in underperforming scenes such as polyp and lesion detection. Meanwhile, Grounded SAM [17] combines open-set object detectors like Grounding DINO [13] with SAM to enable language-driven segmentation, though its direct use in medical imaging remains an open area for investigation. In parallel, Vision-Language Models (VLMs) such as BLIP-2 [11] and MedCLIP [21] offer a means of generating segmentation prompts automatically from textual or visual cues. Most relevant to our work is Learning to Prompt, introduced by Huang et al. [6], which trains a model to generate optimal prompts (e.g., bounding boxes) to guide SAM for improved segmentation. This automated prompting strategy bridges the gap between general-purpose foundation models and the specificity required in medical imaging.

Our project builds on these insights by exploring both fine-tuning and prompt automation strategies to enhance SAM's performance in brain tumor segmentation, using MRI data from the BraTS 2024 challenge [4].

## 3. Methods

### 3.1. Baseline

As our baseline, we use the nnU-Net [8] model pre-trained on brain tumor datasets from previous BraTS challenges. We use the publicly available checkpoint without further fine-tuning. Unlike SAM, nnU-Net is a fully supervised segmentation model that does not take any external prompts like points or boxes as input. Instead, it directly maps MRI slices to dense segmentation masks using pixel-wise supervision. To ensure a fair comparison with SAM which operates 2D RGB images, we extract the same MRI slice used for SAM and convert it to a 3-channel format before feeding it to nnU-Net. This ensures both models receive consistent input data despite their differing architectures and supervision paradigms, providing a strong reference point for evaluating how well prompt-based models like SAM perform under low-supervision or zero-shot settings.

### 3.2. Segment Anything Model

The Segment Anything Model (SAM) is a state-of-the-art prompt-able image segmentation foundation model that allows zero-shot performance transfer. SAM was pre-trained on a dataset that consists of 11M diverse and high-resolution images along with 1.1B segmentation masks [10]. The model architecture contains three components: an image encoder, a prompt encoder, and a mask decoder. The image encoder is a Masked-Auto-Encoding pre-trained Vision Transformer. In additional to accepting images as inputs, SAM accepts two sets of prompts: sparse (points, boxes, text) and dense (masks). For the sparse prompts, points and boxes are represented by positional encodings and their learned embeddings, along with text embeddings generated from CLIP. Dense mask prompts are embedded with convolution networks. However, even though text prompt is mentioned in the original paper, the implemented model does not accept free-form text as a prompt, as it is more of a proof-of-concept feature. The mask decoder turns the combination of an input image embedding, prompt embeddings, and an output token into a segmentation mask. It utilizes a modification of the Transformer decoder block by including prompt self-attention and cross-attention between image and prompt embeddings. For ambiguous prompts that can correspond to multiple objects in the image, SAM outputs 3 different masks along with their confidence scores. We can also ask the model to output the mask with the highest confidence score during evaluation/training for ambiguous prompts. However, we use the single-mask mode of SAM in our experiments, which outputs the single-mask prediction with the highest confidence score, since we believe that users would normally want a single predicted mask for convenience, and our experiments should reflect performance applicable to real-world settings.

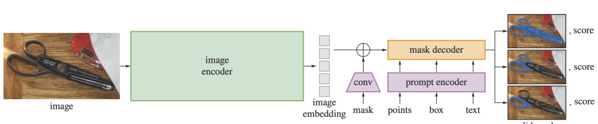


Figure 2. Segment Anything Model Architecture [10]

### 3.3. Prompting the Segment Anything Model

Given the prompt-able feature of the **Segment Anything Model**, we wanted to utilize **Vision Language Models** and/or **Object Detection Models** to help SAM perform the brain tumor segmentation task. We hypothesized that providing correct points or boxes as prompts to SAM can help SAM correctly focus on specific regions to identify the objects or regions of interest, which in our case are the brain tumor regions, and generate correct segmentation masks. However, users often do not have the correct point or box prompts when using SAM in real life. Therefore, we want to evaluate the performance of using a Vision Language Model and/or Object Detection Model to automate the process of prompt-generation for brain tumor segmentation.

To identify the best type of prompt to supply to SAM for our task, we evaluated and compared the performance of the Segment Anything Model when feeding in three different types of prompts generated from the ground-truth brain tumor masks: center point, 20 randomly sampled points, and bounding box of the ground truth tumor mask. The precise definition and the process of constructing these prompts are described in Section 4. In addition to helping us identify the best type of prompt to use, this evaluation establishes our baseline performance by helping us better assess the zero-shot performance of SAM, without any fine-tuning, on brain tumor segmentation.

Based on the evaluation results A, we can draw the following conclusions. First, using only the center point of the ground-truth mask does not really help SAM focus on the desired region. The model often just segment out the entire brain in the MRI. Given the irregular shapes of brain tumors, the center of the tumor region may not lie in the tumor itself, as shown in Figure 8. This can also create ambiguities when asking the Vision Language Model to locate the center of a possible tumor region. Second, when using randomly sampled points within the ground-truth tumor mask, SAM often generates mask predictions that are not consistent and do not form an enclosed region, as shown in Figure 6. Overall, ground-truth bounding-box prompts lead to better zero-shot performance, as they can help SAM better focus on a specific area. Moreover, the irregular shapes of brain tumors hinder SAM's zero-shot performance, as the model often detects the "smoothened" region, as shown in Figure 7. This suggests that further fine-tuning SAM to the BraTS dataset is necessary as it could help the model gain a better understanding of the various shapes of brain tumors.

The performance difference when using different types of prompts is supported by the quantitative results based on evaluation metrics described in Section 5.1. Table 1 exhibits the DSC and HD95 values for the output masks predicted by SAM given each of the three types of ground-truth generated prompts. We can see that using bounding boxes as prompts to SAM led to the highest DSC and lowest HD95, indicating that it has the best performance. Based on these evaluation results for using ground-truth generated prompts, we decided to use bounding box as our chosen type of prompt that we will ask a Vision Language Model to generate, given its better performance. We will also use bounding boxes as prompts to fine-tune SAM.

| | DSC | HD95 |
|---|---|---|
| center point as prompt | 0.3612 | 54.34 pixels |
| 20 random points as prompt | 0.5102 | 51.39 pixels |
| bounding box as prompt | 0.7623 | 10.36 pixels |

Table 1. Evaluation Results of SAM Given Ground-Truth Generated Prompts

### 3.4. Bounding Box Selection

We experimented with both open-source zero-shot multi-model Vision Language Models (VLM) and zero-shot Object Detection Models for generating the bounding boxes of brain tumors, including Qwen2-VL-7B-Instruct by Alibaba Rsearch [20], MedGemma by Google [5], Blip-Vqa by Salesforce [12], and GroundingDino by IDEA-Research [13]. We found that **MedGemma** has the best performance on zero-shot brain tumor detection.

MedGemma is a Gemma 3 variant that is trained on medical text and/or images for comprehension tasks. MedGemma comes in two variants: a 4B multimodal version and a 27B text-only version. We used the 4B multimodal version for the bounding box generation.

MedGemma 4B utilizes a SigLIP image encoder that has been pre-trained on a variety of medical data, such as chest X-rays, dermatology images, ophthalmology images, and histopathology slides. Its LLM component is pre-trained on a set of medical data, such as radiology images, histopathology patches, ophthalmology images, dermatology images, and medical text.

The better zero-shot performance on MedGemma is expected as it was specifically trained on medical image and text data. See Appendix Section B for the prompt we used for the MedGemma model.

In contrast, Blip-Vqa and GroundingDINO, despite being a strong open-set object detector in natural image domains, failed to detect tumors in our MRI datasets. This is likely due to a domain mismatch and the models' lack of exposure to grayscale medical imaging. These results suggest that current zero-shot object detection methods do not generalize well to brain tumor detection tasks, especially in the medical domain. Therefore, we excluded these models from further experimentation.

In additional to the open-source models, we also experimented with using one-shot example, such as giving an example MRI image with the brain tumor drawn with its bounding box, to ask ChatGPT to generate bounding box for the brain tumor. See Appendix Section C for the one-shot example as well as the prompt used for ChatGPT.

Worth to note, we also experimented with one-shot examples when using the open-source VLM models. However, we noticed that one-shot examples tend to cause worst behaviors. We suspect it is due to the models' smaller parameter size and training data size vs. ChatGPT, which make them harder to interpret prompts that they have not seen before during training.

### 3.5. SAM Fine-Tuning

While SAM is a foundation model trained on a wide variety of images and segmentation masks, it is not specialized in tasks that are highly domain-specific, such as our brain tumor segmentation task. Therefore, we hypothesize that fine-tuning SAM on a brain tumor segmentation dataset would improve SAM's performance on our task. We fine-tune on the `facebook/sam-vit-base` pre-trained SAM model available on Hugging Face [7]. This model has 91M parameters [10], an adequate model size given our computational constraints. We start from the fine-tuning code written by Neils Rogge [18] and implement heavy modifications to preprocess data from our dataset, optimize running time and memory efficiency using methods like mixed precision training, collect and visualize metrics, and establish hyperparameter tuning pipeline, early stopping mechanism, and inference pipeline. We conduct full-scale fine-tuning to the mask decoder part of SAM and freeze the image and prompt encoders. SAM's image encoder is already well trained to extract important features from images, and the prompt encoder does not need to be changed since our bounding box prompts should be encoded the same way as any other bounding box, so it is reasonable to only fine-tune on the mask decoder to help SAM adapt to the brain tumor segmentation task while saving compute and preventing overfitting to our relatively small dataset.

During training, each resized and normalized MRI slice is fed to SAM as input, along with one of its associated ground-truth or augmented bounding boxes as the prompt. The output is a predicted tumor mask and the resized ground-truth segmentation mask is used as supervision. We use the `DiceCELoss` from MONAI [15], which combines Dice loss and binary cross-entropy loss to balance region overlap and pixel-wise accuracy. The binary cross-entropy (BCE) loss for each pixel is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \quad (1)$$

where $y_i \in \{0,1\}$ is the ground-truth label for pixel $i$, $\hat{y}_i \in [0,1]$ is the predicted probability, and $N$ is the total number of pixels.

The Dice Similarity Cofficient (DSC) is explained in Section 5.1, which quantifies the spatial overlap between the predicted tumor region and the ground-truth mask, with a higher score indicating better agreement. The Dice loss is then:

$$\mathcal{L}_{\text{Dice}} = 1 - \text{DSC} \quad (2)$$

The final training loss is a weighted sum of Dice loss and BCE loss:

$$\mathcal{L}_{\text{DiceCE}} = \lambda_{\text{Dice}} \cdot \mathcal{L}_{\text{Dice}} + \lambda_{\text{BCE}} \cdot \mathcal{L}_{\text{BCE}} \quad (3)$$

### 3.6. Inference Procedures

Figure 1 illustrates our proposed inference pipeline. This pipeline feeds an MRI image into a VLM or object detection model and asks the model to generate a bounding box that captures the tumor region in the image. Then, the pipeline resizes the bounding box generated by the VLM or object detection model into the size accepted by SAM, then feed it as a prompt, along with the image input, into the fine-tuned SAM to obtain a predicted mask of the brain tumor region in the image as output. We run (1) zero-shot inference on the MedGemma VLM and (2) one-shot inference on ChatGPT, but they can be replaced by any other VLM or object detection model, demonstrating the flexibility of our pipeline. This pipeline ensures that everything can be automated without human efforts such as manually creating bounding boxes as prompts. We evaluate on the performance of our proposed pipeline and compare it against our baseline performance. Evaluation results, along with ablation studies showing the importance of fine-tuning in improving performance, are described in Section 5. The evaluation results demonstrate that our pipeline is able to achieve superior performance compared to that of the baseline state-of-the-art model while requiring no human effort to provide bounding boxes.

### 3.7. Bounding Box Data Augmentation

We adopt a data augmentation technique such that for each MRI image in dataset, in addition to the ground-truth derived bounding box, we create 6 more bounding boxes by adding random noise to each of the edge coordinates of the ground-truth bounding box. We hypothesize that using bounding boxes with noises as prompts during fine-tuning would help SAM generalize to situations where the bounding boxes are imperfectly constructed, which would be beneficial to our inference pipeline since bounding boxes generated by VLMs are almost always imperfect. Therefore, using these augmented bounding boxes as prompts during training should force the model to be more robust to poorly constructed bounding boxes by relying less on bounding boxes for brain tumor segmentation. Moreover, this augmentation procedure effectively increases the training set size by 7 times, which should the risk of overfitting and help the model generalize better.

## 4. Dataset and Features

Our experiments are conducted on the brain tumor dataset provided by the 2024 Brain Tumor Segmentation (BraTS) Challenge [4]. As described in the challenge paper, this dataset consists of multi-institutional, multi-modal MRI scans from approximately 2,200 post-operative glioma patients. For each case, four imaging modalities are provided: T1-weighted (T1), T1-weighted contrast-enhanced (T1c), T2-weighted (T2), and T2-FLAIR, along with expert-annotated segmentation masks delineating tumor sub-regions.

In the released data, we find a total of 1,809 cases. The validation set (188 cases) does not include ground-truth labels, as the challenge results are not yet finalized. Therefore, we exclude it and instead split the training set into 1,350 training and 271 test cases (approximately a 5:1 ratio), which we use for our own evaluation.

In our work, we specifically use the T2-FLAIR modality as input for segmentation due to its effectiveness in visualizing tumor-related edema. All annotated tumor sub-regions (enhancing tumor, tumor core, and edema) are merged into a single binary tumor region. We do not differentiate between subtypes of tumor tissue.

Since SAM operates on 2D RGB images and the original dataset consists of 3D volumetric scans, we extract representative 2D slices to serve as input to both VLMs and SAM. Specifically, we extract the 2D axial slice with the largest tumor area from each 3D scan. Each slice is normalized and converted to RGB for compatibility with SAM, resulting in a dataset of 1,621 labeled 2D images at 218 by 182 resolution. To comform with SAM's input and target size requirements, we resize each input image and bounding box to 1024 by 1024 resolution and resize each target mask to 256 by 256 resolution during training and validation. Both normalization and resizing are done through the SamProcessor class from HuggingFace [7].

To assess the performance of SAM on brain tumor image segmentation when accurate prompts are provided, we derive from the ground-truth segmentation masks in the dataset and generate different types of prompts for each image:

- Center point prompt: geometric center point of the ground-truth tumor area.
- Random points prompt: randomly sampled points from the ground-truth tumor area.
- Ground-truth bounding box prompt: minimal axis-aligned rectangle enclosing the tumor area.
- Noised bounding box prompt: a perturbed version of the ground-truth box, where each side is randomly expanded by a value sampled uniformly from 0 to a maximum perturbation level. We experiment with maximum values of 5, 10, 15, 20, 30, 40 pixels.

Note that noised bounding box prompts are used as the bounding box prompts when fine-tuning SAM. Since we create 6 noise bounding boxes in addition to the ground-truth bounding box prompt, we form a training set of 8505 examples and a validation set of 945 examples after train-validation split by pairing each image with each of the 7 bounding boxes.

## 5. Experiments/Results/Discussion

### 5.1. Evaluation Metrics

We employ both quantitative and qualitative metrics to comprehensively assess model performance. Quantitatively, we use the Dice Similarity Coefficient (DSC), the

95th Percentile Hausdorff Distance (HD95), and the Intersection over Union (IoU) to evaluate the overlap and spatial accuracy between predicted segmentations and ground truth masks. These metrics are standard in medical image segmentation.

**Dice Similarity Coefficient (DSC)** quantifies the overlap between a predicted segmentation region $P$ and the corresponding ground truth region $G$, and is defined as:

$$\mathrm{DSC}(P,G) = \frac{2|P \cap G|}{|P| + |G|} \tag{4}$$

**Intersection over Union (IoU)**, also known as the Jaccard Index, is another measure of set similarity and is given by:

$$\mathrm{IoU}(P,G) = \frac{|P \cap G|}{|P \cup G|} \tag{5}$$

**95th Percentile Hausdorff Distance (HD95)** measures the spatial discrepancy between the boundaries of $P$ and $G$. It is defined as:

$$\mathrm{HD}_{95}(P,G) = \max \left\{ \begin{array}{l} \mathrm{Percentile}_{95} \left( \min\limits_{g \in G} d(p,g) \right)_{p \in P}, \\ \mathrm{Percentile}_{95} \left( \min\limits_{p \in P} d(g,p) \right)_{g \in G} \end{array} \right\} \tag{6}$$

where $d(a,b)$ denotes the Euclidean distance between points $a$ and $b$.

In addition to these quantitative metrics, we conduct qualitative evaluations by visualizing segmentation outputs across representative cases to assess the model's ability to localize and delineate tumor regions with high fidelity.

### 5.2. Bounding Box Evaluations

We evaluate the accuracies of the bounding boxes generated by both MedGemma and ChatGPT and compare them against the ground-truth bounding boxes in the test set by calculating the IoU scores between the two regions. We can see that ChatGPT does have a meaningfully higher performance in terms of bounding box generation compared to MedGemma for brain turmors. SAM predictions with MedGemma can show us the robustness of SAM when given lower quality bounding boxes.

| **IoU** | 25% | 50% | 75% | Max | Mean | Std Dev |
|---|---|---|---|---|---|---|
| **MedGemma** | 0.13 | 0.21 | 0.31 | 0.73 | 0.23 | 0.15 |
| **ChatGPT** | 0.09 | 0.30 | 0.60 | 0.99 | 0.36 | 0.31 |

Table 2. IoU statistics summary for MedGemma and ChatGPT

### 5.3. SAM Fine-Tuning Hyperparameter Selection

We established a hyperparameter tuning pipeline when fine-tuning SAM. We tried different values for learning rates, weight decay, the number of epochs, and the choice of optimizer (e.g., Adam vs. AdamW). Because of computational constraints, running a full scale grid search on all training and validation data is too expensive. Thus, we hand-picked several combinations of hyperparameter values on a coarse scale, where the hand-picking process is dynamically guided by the performance of previous hyperparameter value combinations. We used a random subset of the training and validation sets for training and validation. Specifically, we used a training set size of 1701 and a validation set size of 343 for hyperparameter tuning. We run the model on the training subset using each hand-picked combination of hyperparameter values and choices, and evaluate the model's performance on the validation subset after each epoch. We record the best validation set performance across epochs during fine-tuning, and pick the hyperparameter combination that yields the lowest validation DiceCELoss. In the end, we found that a learning rate of $1e{-}4$, a weight decay of $1e{-}5$, a number of training epochs of 10, and AdamW as the optimizer yields the lowest loss on the validation subset. We did not perform cross-validation due to heavy computational constraints.

### 5.4. SAM Fine-Tuning using Best Hyperparameters

We use the aforementioned best combination of hyperparameters to do a final training on the entire training set, which consists of 8505 examples, except that we decrease the number of epochs to 5 due to computational constraints. We believe that this should not hurt the performance too much since we observed that validation loss does not decrease much after 5 epochs when training on the training subset, as we can see in Figure 18 (note that the number of epochs in the plot is 0-indexed). During the final training run on the entire training set, we measured the loss on the entire validation set, which consists of 945 examples, after each training epoch. Figure 3 shows the training loss and validation loss over the number of training epochs (note that the number of epochs in the plot is 0-indexed). The lowest validation loss, which occurs after the last epoch, is $0.2661$, and the lowest training loss is $0.2275$. The plot shows that both training and validation losses decrease over time, suggesting the effectiveness of training.

Although training and validation loss curves seem to be diverging, the validation loss curve of the full training run is still decreasing. Moreover, although a lower learning rate and a higher weight decay can mitigate overfitting, we observed that this setup does not lead to a lower training or validation loss. Therefore, we believe that using our current choices of hyperparameters is reasonable regardless of whether the loss curves seem to diverge. For example, when using a learning rate of $1e-6$ and weight decay of $1e-3$, although the loss curves do not diverge, as shown in Figure 19, validation loss remains high compared to our chosen hyperparameters despite more training epochs.

### 5.5. Ablation Study on Data Augmentation

The fine-tuned model and hyperparameter selection process described in Section 5.3 and Section 5.4 uses a training dataset that consists of examples with augmented bounding boxes as prompts to SAM. We would like to investigate whether the augmented examples can truly improve performance. Therefore, we fine-tune on another instance of SAM but only uses the ground-truth bounding boxes as prompts during training and validation, so each each image in the dataset only pairs with one bounding box. Due to computational constraints, we used the same combination of hyperparameters as in Section 5.4, although it would be better if we could conduct a separate hyperparameter tuning process for this ablation study, since the optimal set of hyperparameter will likely be different, although probably not to a great extent since the training images remain the same.


Figure 3. Training and validation loss over 0-indexed number of epochs during the final training run using all training and validation data.

### 5.6. Test Set Segmentation Performance

We evaluate the performance of SAM after fine-tuning on the test set for the brain tumor segmentation task and compare the results with the results of SAM without fine-tuning. We evaluate their performance on three types of bounding boxes: those derived from ground-truth mask, those generated by MedGemma, and those generated by ChatGPT. While the latter two demonstrate the performance of our proposed inference pipeline by providing VLM-generated prompts to SAM, evaluating on using ground truth bounding boxes demonstrates the full potential of SAM when given optimal bounding boxes, which could act as an upper bound to SAM's performance. We also compare SAM's performance with nnU-Net, which serves as our baseline.

| | DSC | HD95 (in pixels) | IoU |
|---|---|---|---|
| w/o FT & GT bbox | 0.7624 | 10.36 | 0.6413 |
| w/o FT & MedGemma bbox | 0.1855 | 55.44 | 0.1172 |
| w/o FT & ChatGPT bbox | 0.4210 | 39.70 | 0.3202 |
| w/ FT & GT bbox | 0.7404 | 10.44 | 0.6055 |
| w/ FT & MedGemma bbox | 0.2707 | 43.17 | 0.1805 |
| w/ FT & ChatGPT bbox | 0.4403 | 32.90 | 0.3258 |
| w/ small-FT [1] & GT bbox | 0.7689 | 8.97 | 0.6381 |
| w/ small-FT & MedGemma bbox | 0.2555 | 52.38 | 0.1598 |
| w/ small-FT & ChatGPT bbox | 0.4344 | 36.65 | 0.3197 |
| nn-UNet | 0.25 | 37.5530 | 0.1959 |

Table 3. Results.

Table 3 is a compilation of the performance of our baseline and SAM under different settings. Comparing the performance of SAM without fine-tuning, with fine-tuning, and with fine-tuning but without data augmentation when given bounding boxes generated by MedGemma and ChatGPT, we can see that SAM with fine-tuning performed the best by having the highest DSC and IoU and lowest HD95 scores. This shows that fine-tuning, especially with data augmentation, helps SAM become robust to low-quality bounding boxes. The difference is especially significant with bounding boxes generated by MedGemma, which have even lower qualities than those generated by ChatGPT. Although SAM with fine-tuning but without data augmentation performs slightly worse, it is still better than without fine-tuning, showing the benefit of fine-tuning regardless of data augmentation.

Moreover, we see that combining fine-tuned SAM with ChatGPT for bounding box generation, our pipeline yields a better performance than our baseline, illustrating that our pipeline can surpass a state-of-the-art model while being fully automated, which highlights the contribution of our study.

Interestingly, when using ground-truth bounding boxes as prompts, SAM without fine-tuning performs better than SAM with fine-tuning, although the difference is not significant, and is still worse than SAM with fine-tuning but without data augmentation. This shows that SAM is powerful enough at brain tumor segmentation tasks as long as high-quality bounding boxes are given, suggesting the power of large pre-trained foundation segmentation models. However, fine-tuning can still improve performance if we only use ground-truth bounding boxes during training. One possible explanation for why SAM with fine-tuning and data augmentation performs worse is that the augmented data help SAM generalize to low-quality bounding boxes at the expense of worse performance when given high-quality bounding boxes, since SAM cannot rely as much on the quality of bounding boxes to segment brain tumor regions.

The performance gap between using ground-truth bounding boxes and VLM generated bounding boxes indicates that generating high-quality bounding boxes remains a difficult task even for state-of-the-art VLMs or VLMs

[1] small-FT refers to the model fine-tuned on the smaller dataset without Bounding Box Data Augmentation

specifically trained on medical data. This suggests that improving VLM's performance on brain tumor bounding box generation could be a promising research direction.

### 5.7. Model Segmentation Visualization

Below shows the performance of zero-shot and fine-tuned SAM when given the MedGemma generated boxes as the prompt. We can observe that after supervised fine-tuning on the BraTS dataset, SAM is more robust to bounding boxes of lower quality 1214. Without fine-tuning, SAM has the tendency to generate masks that do not form a coherent shape when given bad bounding boxes. Moreover, we have included segmentation visualizations of zero-shot/fine-tuned SAM across all the different models in the Appendix D. ChatGPT generally has higher quality bounding boxes, which is reasonable given its much larger parameter size and the use of one-shot prompting.

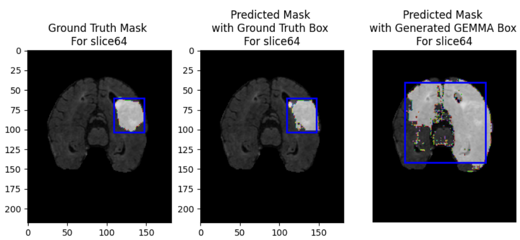


Figure 4. MedGamma BBox Performance with Zero-Shot SAM

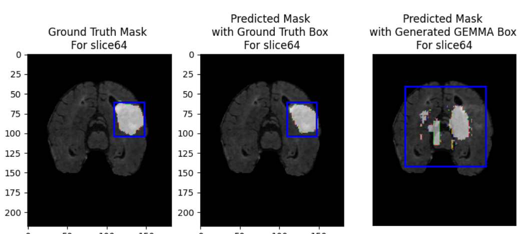


Figure 5. MedGamma BBox Performance with FT SAM

## 6. Conclusion/Future Work

### 6.1. Conclusion

Our proposed inference pipeline involves asking a VLM to generate bounding boxes for brain tumor regions when given MRI images, then feeding the original images, along with the generated bounding boxes as prompts, into the SAM model fine-tuned with data augmentation, to produce final predicted segmentation masks. This pipeline would be beneficial in real-life situations when a ground-truth bounding box is hard to obtain. Our evaluation results show that the performance of our proposed pipeline surpasses the performance of a state-of-the-art medical image segmentation model, indicating our contribution to the field of brain tumor segmentation. Furthermore, we find that augmenting ground-truth bounding boxes with noise and using those as prompting when fine-tuning SAM could help SAM become more robust to imperfect bounding boxes generated by different VLMs, although the overall performance is bottlenecked by the quality of the bounding boxes generated by VLMs, which suggests the inherent difficulty of brain tumor bounding box generation task. The improvement in the performance of our pipeline compared to state-of-the-art model performance has the potential of leading to real-world impacts, since even a small amount of performance and accuracy increase could save additional lives by helping physicians diagnose and monitor the tumor development progress of patients.

### 6.2. Future Work

Our fine-tuning method relies on augmenting ground-truth bounding boxes with noise to help SAM become robust to bounding boxes generated by VLMs, which tend to have lower qualities. It would also be reasonable to directly use bounding boxes generated by VLMs as input prompts using training. We hypothesize that this should lead to even better performance, especially when the training set is large, since this creates a harder task for SAM which eventually could guide SAM become comfortable with bounding boxes generated by a particular VLM. However, one potential downside of fine-tuning SAM on outputs of a particular VLM could mean worse generalization if users want to switch to using another VLM. Moreover, one could investigate whether test-time compute can lead to better performance by asking a VLM to generate multiple bounding boxes at inference time, feed each as a prompt into SAM, and obtain a final predicted segmentation mask by taking majority votes pixel-wise. Another future research direction is to explore fine-tuning a 3D foundation segmentation model on brain tumor segmentation tasks using MRI images, since MRI images are often in 3D, and utilizing 3D data could help physicians diagnose with a greater accuracy. Lastly, if given more compute resources, more exhaustive hyperparameter tuning and training for longer epochs would likely further enhance model performance.

## 7. Contribution

- June Zheng:
  - VLM and object detection model experimentation
  - MedGemma bounding box generation
  - SAM model test-time inference pipeline
  - Performance evaluation and visualization Pipeline
  - Final paper writing and analysis
- Yi Jing:
  - Data processing and ground-truth-based prompts generation
  - Performance evaluation pipeline
  - Baseline model performance evaluation
  - VLM and object detection model experimentation
  - ChatGPT bounding box generation
  - Final paper writing and analysis
- Komei Ryu:
  - Fine-tuning SAM with and without data augmentation
  - Hyperparameter tuning for fine-tuning SAM
  - Final paper writing and analysis

## References


[1] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022.

[2] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[3] T. Chen, L. Zhu, C. Deng, R. Cao, Y. Wang, S. Zhang, Z. Li, L. Sun, Y. Zang, and P. Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023.

[4] M. C. de Verdier, R. Saluja, L. Gagnon, D. LaBella, U. Baid, N. H. Tahon, M. Foltyn-Dumitru, J. Zhang, M. Alafif, S. Baig, K. Chang, G. D'Anna, L. Deptula, D. Gupta, M. A. Haider, A. Hussain, M. Iv, M. Kontzialis, P. Manning, F. Moodi, T. Nunes, A. Simon, N. Sollmann, D. Vu, M. Adewole, J. Albrecht, U. Anazodo, R. Chai, V. Chung, S. Faghani, K. Farahani, A. F. Kazerooni, E. Iglesias, F. Kofler, H. Li, M. G. Linguraru, B. Menze, A. W. Moawad, Y. Velichko, B. Wiestler, T. Altes, P. Basavasagar, M. Bendszus, G. Brugnara, J. Cho, Y. Dhemesh, B. K. K. Fields, F. Garrett, J. Gass, L. Hadjiiski, J. Hattangadi-Gluth, C. Hess, J. L. Houk, E. Isufi, L. J. Layfield, G. Mastorakos, J. Mongan, P. Nedelec, U. Nguyen, S. Oliva, M. W. Pease, A. Rastogi, J. Sinclair, R. X. Smith, L. P. Sugrue, J. Thacker, I. Vidic, J. Villanueva-Meyer, N. S. White, M. Aboian, G. M. Conte, A. Dale, M. R. Sabuncu, T. M. Seibert, B. Weinberg, A. Abayazeed, R. Huang, S. Turk, A. M. Rauschecker, N. Farid, P. Vollmuth, A. Nada, S. Bakas, E. Calabrese, and J. D. Rudie. The 2024 brain tumor segmentation (brats) challenge: Glioma segmentation on post-treatment mri, 2024.

[5] Google. Medgemma hugging face.

[6] J. Huang, K. Jiang, J. Zhang, H. Qiu, L. Lu, S. Lu, and E. Xing. Learning to prompt segment anything models. *arXiv preprint arXiv:2401.04651*, 2024.

[7] Hugging Face. SAM. `https://huggingface.co/docs/transformers/en/model_doc/sam`, 2025. Accessed: 2025-06-04.

[8] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[9] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, L. Rolland, D. Gustafson, C.-Y. Xiao, S. Whitehead, V. Cevher, P. Dollár, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[11] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[12] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[13] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023.

[14] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024.

[15] MONAI Consortium. DiceCELoss. `https://docs.monai.io/en/stable/losses.html#monai.losses.DiceCELoss`. Accessed: 2025-06-04.

[16] M. Prastawa, E. Bullitt, S. Ho, and G. Gerig. A brain tumor segmentation framework based on outlier detection. *Medical image analysis*, 8(3):275–283, 2004.

[17] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[18] N. Rogge. Tutorials, 2025.

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Ger-*

*many, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[20] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[21] Z. Wang, Z. Wu, D. Agarwal, and J. Sun. Medclip: Contrastive learning from unpaired medical images and text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2022, page 3876, 2022.

[22] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.

[23] Y. Zhang, Z. Shen, and R. Jiao. Segment anything model for medical image segmentation: Current applications and future directions. *Computers in Biology and Medicine*, page 108238, 2024.

# A. Predicted Masks of SAM Given Ground-truth Generated Prompts

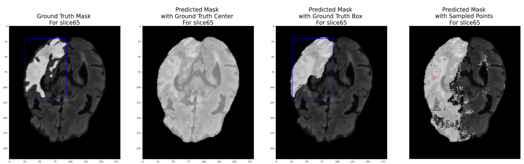


Figure 6. Slice 65

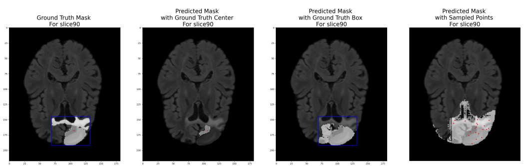


Figure 7. Slice 90

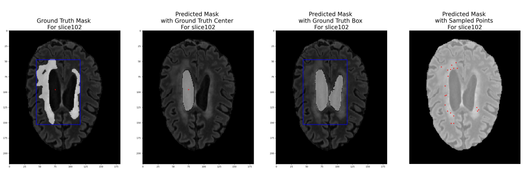


Figure 8. Slice 102

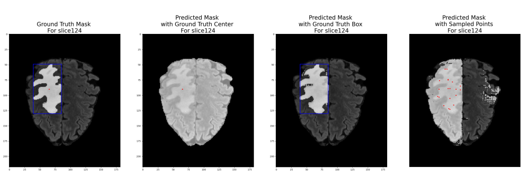


Figure 9. Slice 124

## B. Zero-Shot Prompt for MedGemma

```
Examine the brain MRI image. Your task is to identify the tumor region and provide
bounding box coordinates for it.

Instructions:
1. Carefully examine the image to detect visual patterns consistent with brain tumor
regions (e.g., asymmetry, hyperintensities).
2. Return the top left and bottom right coordinates of the bounding box surrounding the
tumor.
3. Double check that the coordinates of the brain tumor bounding box is in the MRI image.

Image dimensions:
- Width = 182 pixels
- Height = 218 pixels
- Coordinates must be within the range: (x = 0 to 182, y = 0 to 218)
- The upper left corner of the image has coordinate = (0, 0)

Please format your response as follows:
top left corner of the bounding box has coordinate = (x0, y0)
bottom right corner of the bounding box has coordinate = (x1, y1)

Think step by step. Explain your reasoning in detail. Describe the location of the brain
tumor relative to the MRI image. Give the final answer in bounding box coordinates.
```

## C. One-Shot Prompt for ChatGPT

To generate bounding box annotations for brain tumors in MRI images using ChatGPT, we used a one-shot prompting strategy. Specifically, we first showed an example image where the brain tumor region is highlighted in white and surrounded by a blue bounding box. The following prompt was then used to guide the model:

- **Message 1 (Reference Example):**

  *You are a neuropathologist that does brain tumor annotation. Here is an example MRI with the brain tumor region highlighted white and a blue bounding box around the brain tumor.*

  *[Attached: Example MRI image with tumor and ground truth bounding box]*

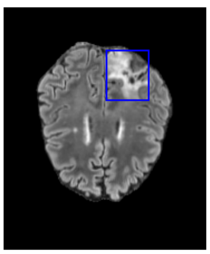

Figure 10. Reference Example

- **Message 2 (Query Image):**

  *This MRI image does not have the blue bounding box. Can you generate the bounding box around the brain tumor and provide the coordinates?*

  *[Attached: Unannotated MRI image for annotation]*

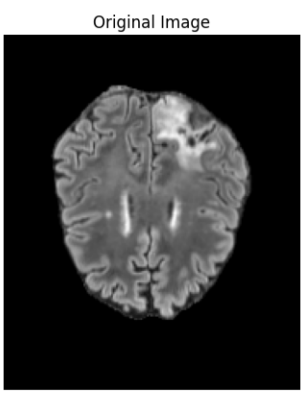


Figure 11. Query Image

This prompt sequence was designed to demonstrate the expected output format using a visual reference and then request bounding box coordinates on a new image. The model returns the bounding box as `(xmin, ymin, xmax, ymax)` coordinates that enclose the visible tumor region.

## D. Model Performance Visualizations

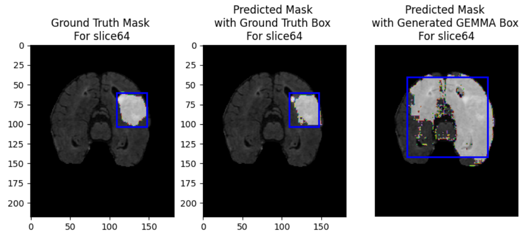


Figure 12. MedGamma BBox Performance with Zero-Shot SAM

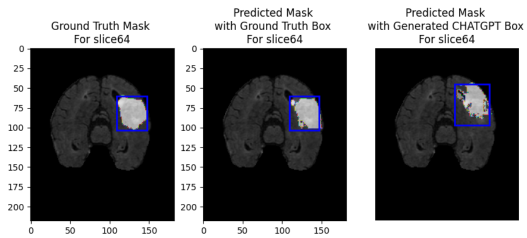


Figure 13. ChatGPT BBox Performance with Zero-Shot SAM

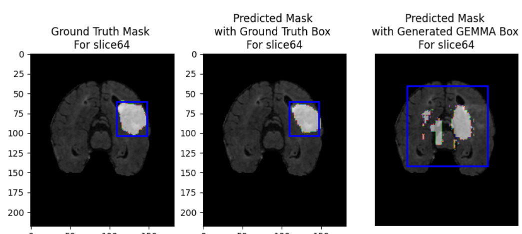


Figure 14. MedGamma BBox Performance with FT SAM

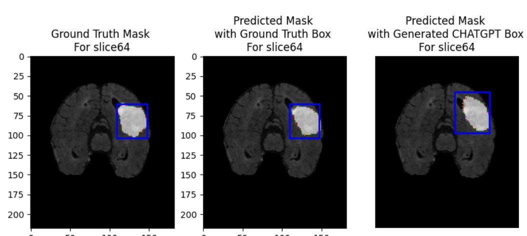


Figure 15. ChatGPT BBox Performance with FT SAM

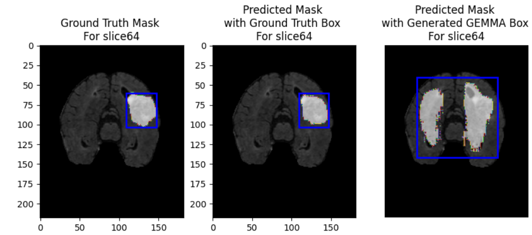


Figure 16. MedGamma BBox Performance with small-FT SAM

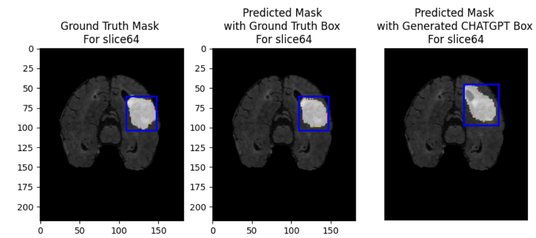


Figure 17. hatGPT BBox Performance with small-FT SAM

## E. Additional Loss Curve Plots Showing Hyperparameter Selection Observations

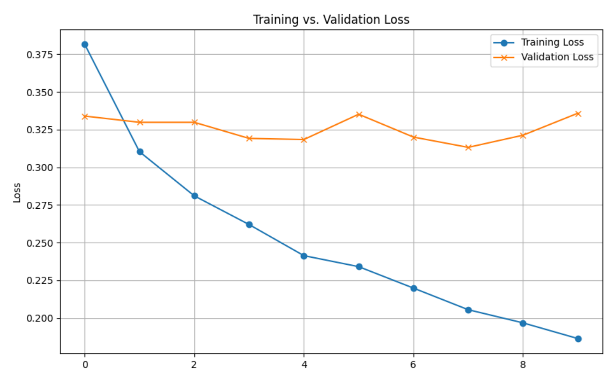


Figure 18. Training and validation loss over 0-indexed number of epochs during the training run of the best combination of hyperparameters on the subsets of training and validation sets.

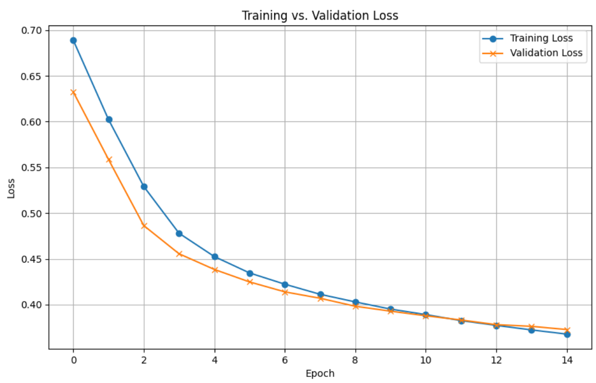


Figure 19. Training and validation loss over 0-indexed number of epochs during the training run using a small learning rate on the subsets of training and validation sets.