# L$^3$Seg: Lean Linear Layers for Language-Guided Vision Transformer in Medical Image Segmentation

**Rahul Bhardwaj**[1,*] **Utkarsh Yashwant Tambe**[2] **Debanga Raj Neog**[1]

[1]Mehta Family School of Data Science & Artificial Intelligence,
Indian Institute of Technology Guwahati, Guwahati, India

[2]Department of Data Science & Business Systems,
SRM Institute of Science & Technology, Kattankulathur, India

{r.bhardwaj, dneog}@iitg.ac.in; ty7171@srmist.edu.in

## Abstract

Vision-language models offer strong potential for medical image segmentation by integrating visual data with clinical text. However, these models typically involve large parameter counts and high computational cost, making them impractical for real-time use. This paper presents L$^3$Seg, a lightweight and efficient vision-language segmentation framework. The key component is the Lean Linear Layer (L$^3$), a linear projection that freezes pretrained weights and biases while learning only a small, token-dependent residual, parameterized into two low-rank matrices. Unlike conventional parameter-efficient methods, L$^3$ adapts each token representation with minimal extra computational cost. L$^3$Seg replaces all dense linear projections in the vision encoder and the vision-text fusion module with L$^3$, achieving state-of-the-art segmentation with only 8.2M parameters and 5.1GFLOPs. Experiments demonstrate consistent improvements across X-ray (QaTa-COV19), endoscopy (Kvasir-SEG), and ultrasound (BUSI), even with limited training data and sparse textual input. The source code is available at: https://github.com/bhardwaj-rahul-rb/l3seg

## 1 Introduction

Image segmentation in medical imaging aims to delineate regions of interest such as tumors, which is critical for diagnosis and treatment planning. Deep learning models [2] achieve impressive results but depend on large, expert-labeled datasets which are costly and time-consuming to produce. Clinical text reports, routinely written by clinicians, offer valuable supplementary information without extra labeling effort. Recent vision-language methods [26] utilize these reports to enhance segmentation accuracy and reduce reliance on limited annotations. However, these models often carry large parameter counts and high computational cost, making them impractical for real-time or resource-constrained clinical settings. Further challenges arise from variations in lesion characteristics (scale, texture, contrast, and surrounding tissue) across X-ray, endoscopy, and ultrasound [5]. A practical segmentation framework should therefore generalize across these modalities while keeping parameter count and compute low for clinical deployment.

Transformers concentrate most parameters and FLOPs in repeated dense linear projections inside attention and feed-forward layers [24]. Fine-tuning studies show that over 90% of pretrained weights change only slightly, with a small subset contributing most to improvements [21]. Parameter-efficient

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: The 3rd Workshop on Imageomics: Discovering Biological Knowledge from Images Using AI.

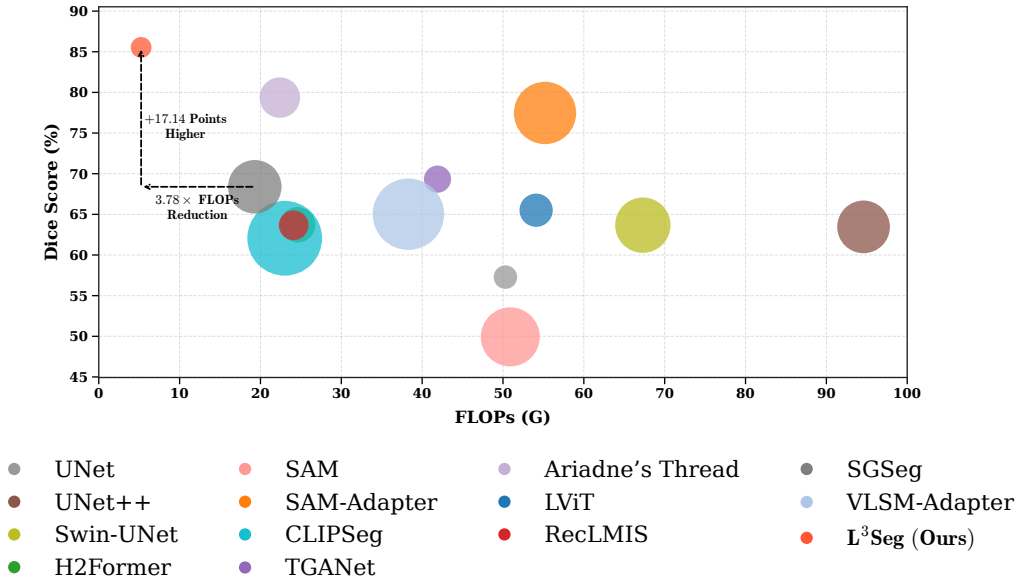



Figure 1: **Comparison with state-of-the-art models on the BUSI dataset.** GFLOPs on the x-axis, Dice (%) on the y-axis, and bubble area encodes parameter count (M). L$^3$Seg attains the best Dice while using the least compute (5.1 GFLOPs), a 3.8-fold reduction in compute compared with SGSeg (19.3 GFLOPs), and an absolute 17.14-point Dice gain on BUSI compared with SGSeg, increasing from 68.39% to 85.53%. Best viewed in color, and further enhance the view by zooming in.

fine-tuning (PEFT) addresses this by freezing most parameters and adding small adapter modules to selected layers [8]. Because the original dense projections remain active, FLOPs largely persist. When adapters are limited to a few locations, their ability to adapt every projection is constrained. Even attaching adapters to all projections offers little compute relief, since the base matrix multiplications still dominate.

This paper introduces L$^3$Seg, a segmentation framework that replaces every dense linear projection with a Lean Linear Layer (L$^3$). Rather than attaching adapter modules next to frozen parameters, L$^3$Seg keeps the pretrained weights and bias fixed and learns a small, token-dependent low-rank residual at each projection. This spreads adaptation across the network and provides dynamic per-token adjustments with minimal overhead. Figure 1 shows that L$^3$Seg achieves the highest Dice on BUSI while substantially reducing compute and parameters compared with prior state-of-the-art unimodal and multimodal methods. The main contributions of this paper are:

- A lightweight vision-language segmentation framework, L$^3$Seg, in which every dense linear projection across the network, including those in the vision encoder and the cross-attention modules, is replaced with L$^3$. This design enables dynamic adaptation across layers and strong segmentation performance on X-ray, endoscopy, and ultrasound with low computational cost and parameter count.

- L$^3$ keeps pretrained weights and biases fixed and learns a small, token-dependent residual parameterized by two low-rank matrices. It reduces trainable parameters per projection from $O(d_{\text{in}}d_{\text{out}})$ to $O(r(d_{\text{in}} + d_{\text{out}}))$, where $d_{\text{in}}$ and $d_{\text{out}}$ are the input and output feature dimensions, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$.

- Extensive experiments on QaTa-COV19 (X-ray), Kvasir-SEG (endoscopy), and BUSI (ultrasound) showing consistent improvements over unimodal and multimodal baselines. Ablations show high accuracy with limited training data and with sparse textual guidance, supporting use in clinical settings.

## 2 Related Works

### 2.1 Medical Image Segmentation

Convolutional U-shaped networks, beginning with U-Net [22] [2], laid the groundwork for medical image segmentation by capturing fine spatial detail and broad context. UNet++ [28] architecture deepened this design with nested skip pathways, improving feature fusion across scales. Transformer-based variants such as Swin-UNet [3] adopt a hierarchical Swin backbone with shifted-window self-attention, capturing long-range context more efficiently than full global attention and improving performance on multi-organ and cardiac benchmarks. Hybrid approaches, for example H2Former [9], combine convolutional encoders, Transformer blocks, and multi-scale fusion, improving representation quality while keeping computational cost moderate. Despite these advances, most segmentation models still require large expert-labeled datasets. $L^3$Seg addresses this by pairing images with routinely recorded clinical reports as auxiliary guidance and by replacing dense projections with lean low-rank residual layers, retaining high accuracy even when labeled data are limited.

### 2.2 Language-Guided Segmentation

Early vision-language models extract separate image and text features and fuse them with cross-modal attention [26]. CLIPSeg [18] adds a Transformer decoder on top of CLIP [20] to produce dense masks from natural-language or image prompts. TGANet [23] introduces text-guided attention that injects polyp-size cues, boosting colonoscopy segmentation accuracy. Ariadne's Thread [27] improves lung-infection masks by feeding spatially descriptive prompts into the decoder. LViT [15] combines convolutional features with vision transformer tokens and augments them with medical text to compensate for scarce labels. RecLMIS [11] aligns the two modalities through cross-modal reconstruction, letting visual patches and clinical words predict one another. SGSeg [25] generates localization-aware reports during training and then segments without text at test time. While effective, these systems rely on large backbones or multiple fusion blocks, leading to high parameter counts and FLOP. $L^3$Seg removes this overhead by implementing every dense linear projection as $L^3$, keeping pretrained weights and biases fixed and learning only a low-rank, token-dependent residual. This enables token-wise adaptation with fewer parameters and lower compute, yielding strong segmentation across modalities.

### 2.3 Efficient Adaptation of Pretrained Models

Adapter methods update only small task modules while keeping a large backbone frozen. SAM-Adapter [4] adds visual prompts to the Segment Anything Model (SAM) [14], adjusting only a small share of parameters. VLSM-Adapter [7] follows a similar idea for vision-language segmentation by inserting compact Transformer adapters into both image and text streams. Low-rank updates (LoRA) [10] and related residual schemes [8] reduce the number of trainable parameters for continual or domain-specific medical segmentation, but the dense projections of the base model still run, so FLOPs largely persist. $L^3$Seg adapts at the projection level. Every dense linear projection in the vision encoder and the language-guided vision decoder is implemented as $L^3$ that keeps the pretrained weights and biases fixed and learns a small, token-dependent low-rank residual. This removes separate adapter blocks, preserves pretrained representations, and reduces trainable parameters throughout the network. The additional compute is confined to the low-rank branches, yielding lower GFLOPs than adapter-based and large-backbone baselines in our experiments.

## 3 Method

Figure 2 shows an overview of $L^3$Seg, which is organised into three parts. (i) $L^3$ replace every trainable dense linear projection with a frozen weights and bias plus a low-rank, input-adaptive residual. This substitution keeps the original pre-training intact while greatly reducing the number of learnable parameters. (ii) Dual-Path Encoders, consisting of a Swin-V2-Tiny [17] branch for images and a frozen ClinicalBERT [16] branch for text, produce multi-scale visual features and linguistic embeddings. (iii) Language-Guided Vision Decoder (LGVD) processes the visual hierarchy in three stages, fusing image and text tokens through Gated $L^3$ Cross-Attention ($GL^3CA$) and progressively upsampling the tokens to the final resolution. The following sections describe each of these components in detail.

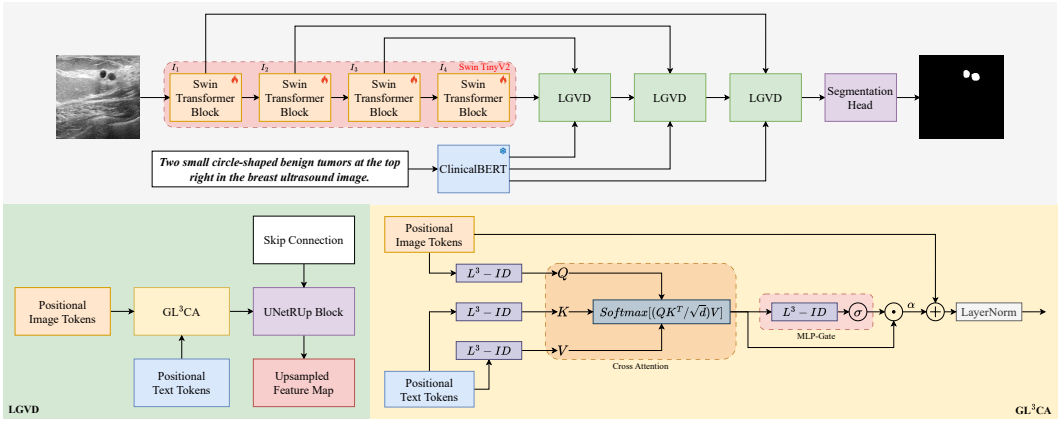



Figure 2: **Overview of the L$^3$Seg pipeline/framework.** A Swin-V2 Tiny image encoder produce multi-scale visual features and a frozen ClinicalBERT text encoder produce contextual embeddings. Every dense linear projection is replaced by its L$^3$ counterpart, which keeps pretrained weights fixed and adds a learnable low-rank residual. Three cascaded LGVD stages fuse image and text tokens via GL$^3$CA and progressively upsample through skip connections to yield the final segmentation.

### 3.1 Lean Linear Layer (L$^3$)

Dense linear projections account for most of the parameter count and FLOPs of modern Transformers [24]. Let an input tensor of token embeddings be

$$x \in \mathbb{R}^{B\times N\times d_{\text{in}}}, \tag{1}$$

where $B$ is the batch size, $N$ the number of tokens, and $d_{\text{in}}$ the input feature dimension. A standard linear layer applies a weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$ and bias $b_0 \in \mathbb{R}^{d_{\text{out}}}$ to produce

$$y = x\,W_0^{\mathsf{T}} + b_0, \tag{2}$$

where $d_{\text{out}}$ is the output dimension.

Rather than updating or replacing $(W_0, b_0)$, the L$^3$ freezes them and learns only a lightweight, input-conditioned residual. Formally, L$^3$ computes

$$y = \big(1+\gamma(x)\big)\;\odot\;\big(x\,W_0^{\mathsf{T}} + b_0\big) + \beta(x), \tag{3}$$

where "$\odot$" denotes element-wise multiplication. The modulation terms are given by low-rank factorizations:

$$\gamma(x) = x\,A_g\,B_g, \quad \beta(x) = x\,A_b\,B_b, \tag{4}$$

with

$$A_g, A_b \in \mathbb{R}^{d_{\text{in}}\times r}, \quad B_g, B_b \in \mathbb{R}^{r\times d_{\text{out}}}, \tag{5}$$

and $r \ll \min(d_{\text{in}}, d_{\text{out}})$. Only these four small matrices $\{A_g, B_g, A_b, B_b\}$ are updated, reducing the per-layer trainable parameter count from $d_{\text{in}} \times d_{\text{out}}$ to $2r\,(d_{\text{in}} + d_{\text{out}})$. Because $(W_0, b_0)$ remains in the forward path, gradients still flow through the frozen pretrained kernel, preserving its prior knowledge while the low-rank residual adapts the layer to the target domain with minimal overhead. See Algorithm 1 for a PyTorch-style pseudocode implementation of L$^3$.

### 3.2 Dual-Path Encoders

#### 3.2.1 Image Encoder

Swin-V2-Tiny [17] serves as the visual backbone, extracting multi-scale feature maps from an input image

$$I \in \mathbb{R}^{H\times W\times 3}, \tag{6}$$

where $H$ and $W$ denote height and width. The model processes $I$ through four hierarchical Transformer stages ($s = 1, \ldots, 4$), where each stage halves the spatial resolution and doubles the channel

**Algorithm 1** Lean Linear Layer ($\mathbf{L}^3$)

**Input:** $x \in \mathbb{R}^{B \times N \times d_{\text{in}}}$
**Frozen base:** $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, $b_0 \in \mathbb{R}^{d_{\text{out}}}$
$B$: batch size, $N$: number of tokens, $d$: feature dimension
**class** LEANLINEARLAYER(*Module*):
  **def** __init__(self, $d_{\text{in}}$, $d_{\text{out}}$, $r$, $W_0$, $b_0$):
    super().__init__()
    self.$W_0 \leftarrow W_0$ $[d_{\text{out}}, d_{\text{in}}]$, frozen
    self.$b_0 \leftarrow b_0$ $[d_{\text{out}}]$, frozen
    # trainable low-rank factors
    **Notation:** $\mathcal{N}(0, 10^{-3})_{m \times n} = \texttt{randn}(m, n) \times 10^{-3}$
    **Notation:** $\mathbf{0}_{m \times n} = \texttt{zeros}(m, n)$
    self.$A_g \leftarrow \mathcal{N}(0, 10^{-3})^{d_{\text{in}} \times r}$
    self.$B_g \leftarrow \mathbf{0}_{r \times d_{\text{out}}}$
    self.$A_b \leftarrow \mathcal{N}(0, 10^{-3})^{d_{\text{in}} \times r}$
    self.$B_b \leftarrow \mathbf{0}_{r \times d_{\text{out}}}$
  **def** forward(self, $x$):
    # 1. frozen baseline
    $y_0 \leftarrow xW_0^\top + b_0$ $[B, N, d_{\text{out}}]$
    # 2. low-rank scale and shift
    $\gamma \leftarrow (xA_g)B_g$ $[B, N, d_{\text{out}}]$
    $\beta \leftarrow (xA_b)B_b$ $[B, N, d_{\text{out}}]$
    # 3. scaled–offset fusion
    **return** $(1 + \gamma) \odot y_0 + \beta$

width. This produces

$$I_1 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 96}, \quad I_2 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 192}, \\ I_3 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 384}, \quad I_4 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 768}. \tag{7}$$

Early stages preserve fine spatial detail, while later stages capture broader context. These multi-scale feature maps are forwarded to the decoder as skip connections, guiding the progressive up-sampling.

Within each Swin block, every dense linear projection is replaced by $\mathrm{L}^3$ in its pre-trained variant $\mathrm{L}^3$-PT. Here, the original weight and bias $(W_0, b_0)$ are initialized from the ImageNet checkpoint and then held fixed; only the low-rank residuals $\gamma(x)$ and $\beta(x)$ (Eqs. 3–4) are learned. This strategy retains the backbone's pretrained knowledge while introducing minimal additional parameters.

For fusion with text, each feature map $I_s$ is reshaped into token form:

$$v_s = \text{reshape}(I_s) \in \mathbb{R}^{B \times N_s \times C_s}, \tag{8}$$

where $N_s = H_s W_s$ and $C_s \in \{96, 192, 384, 768\}$. These token sequences $\{v_1, \ldots, v_4\}$ are then passed to the Language-Guided Vision Decoder for cross-modal integration and up-sampling.

### 3.2.2 Text Encoder

ClinicalBERT [16], pretrained on millions of medical reports and radiology notes, converts each tokenized report

$$T \in \mathbb{Z}^{B \times L}, \tag{9}$$

where $B$ denotes batch size and $L$ the sequence length, into a sequence of contextual embeddings

$$h^{(\ell)} \in \mathbb{R}^{B \times L \times d_{\text{emb}}}, \quad \ell = 1, \ldots, 12, \tag{10}$$

with embedding dimension $d_{\text{emb}} = 768$. All ClinicalBERT parameters remain frozen to retain its domain-specific priors. The final-layer embeddings

$$E = h^{(12)} \in \mathbb{R}^{B \times L \times d_{\text{emb}}}, \tag{11}$$

are used directly as token-level features in the $\mathrm{GL}^3$CA block of LGVD.

### 3.3 Language-Guided Vision Decoder

The decoder restores spatial resolution over three stages by fusing visual tokens with text embeddings through GL³ CA. At each stage $s$:

1. **Positional Encoding.** The flattened image tokens $v_s \in \mathbb{R}^{B\times N_s\times C_s}$ and the text embeddings $E \in \mathbb{R}^{B\times L\times d_{\text{emb}}}$ are enhanced with separate sinusoidal positional encodings, producing

$$v_s' = \mathrm{PE}_{\text{vis}}(v_s), \quad E' = \mathrm{PE}_{\text{txt}}(E). \tag{12}$$

2. **Gated L³ Cross-Attention (GL³ CA).** Queries $Q$, keys $K$, and values $V$ are obtained by applying the identity-initialized Lean Linear Layer (L³-ID) to $v_s'$ and $E'$:

$$\begin{aligned} &Q = \Phi_{\text{ID}}(v_s'), \quad K = \Phi_{\text{ID}}(E'), \\ &V = \Phi_{\text{ID}}(E'), \end{aligned} \tag{13}$$

   where,

$$\Phi_{\text{ID}}(x) = (1+\gamma(x)) \odot x + \beta(x) \tag{14}$$

   begins as the exact identity ($\Phi_{\text{ID}}(x) = x$) and learns only the low-rank residuals $\gamma, \beta$. Standard scaled dot-product attention then produces fused features:

$$\begin{aligned} &A = \mathrm{softmax}\big(QK^{\mathsf{T}}/\sqrt{C_s}\big) \in \mathbb{R}^{B\times N_s\times L}, \\ &F = A\,V \in \mathbb{R}^{B\times N_s\times C_s}. \end{aligned} \tag{15}$$

3. **MLP-Gate.** An L³-ID projection followed by a sigmoid yields a per-channel gate:

$$\begin{aligned} &g = \sigma\big(\Phi_{\text{ID}}^{\text{gate}}(F)\big) \in (0,1)^{B\times N_s\times C_s}, \\ &\widetilde{F} = F \odot g. \end{aligned} \tag{16}$$

4. **Residual Merge & Normalization.** The gated features are scaled by a learnable scalar $\alpha$, added back to $v_s'$, and normalized:

$$U_s = \mathrm{LayerNorm}\big(v_s' + \alpha\,\widetilde{F}\big) \in \mathbb{R}^{B\times N_s\times C_s}. \tag{17}$$

Each $U_s$ is reshaped to $\mathbb{R}^{B\times C_s\times H_s\times W_s}$ and, together with the corresponding encoder skip connection, passed through a UNETR-Up block to double spatial resolution. After three stages (recovering $H/32 \to H/16 \to H/8 \to H/4$), a final $1\times 1$ convolution and sigmoid produce the segmentation mask.

## 4 Experiments and Results

### 4.1 Datasets

To evaluate generalization across imaging modalities, experiments were conducted on three publicly available medical datasets: QaTa-COV19 (chest X-ray) [6], Kvasir-SEG (gastrointestinal endoscopy) [12], and BUSI (breast ultrasound) [1]. The QaTa-COV19 dataset comprises 9258 chest X-ray images labeled for COVID-19 infection, with textual notes obtained from LViT [15]. Following the split defined in [15], 7145 images were used for training and 2113 for testing. Kvasir-SEG contains 1000 endoscopic images of gastrointestinal polyps, with textual annotations sourced from [19]. A random 80/20 train/test split was applied, allocating 800 images for training and 200 for testing. The BUSI dataset includes 780 ultrasound images of breast tumors, with clinical notes derived from [19], and was partitioned identically to Kvasir-SEG, resulting in 623 training images and 157 test images. An example BUSI annotation is: *"Two small circle-shaped benign tumors at the top right in the breast ultrasound image."* (see Figure 2).

Table 1: **Quantitative comparison on QaTa-COV19 (X-ray), Kvasir-SEG (endoscopy), and BUSI (ultrasound).** Results are grouped by unimodal (×) and multimodal (✓). Each method is reported with Params (*M*), FLOPs (*G*), and Dice/mIoU (%). Backbone tags: CNN$^{\diamond}$, SAM$^{\P}$, hybrid$^{\dagger}$. Best and second best are **bold** and underlined.

| Method | Venue | Text | Params (M) ↓ | FLOPs (G) ↓ | QaTa-COV19 (XRay) | | Kvasir-SEG (Endoscopy) | | BUSI (Ultrasound) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ |
| UNet$^{\diamond}$ [22] | MICCAI'15 | × | 14.8 | 50.3 | 79.02 | 69.46 | 81.83 | 74.60 | 57.28 | 49.19 |
| UNet++$^{\diamond}$ [28] | IEEE TMI'19 | × | 74.5 | 94.6 | 79.62 | 70.25 | 82.10 | 74.43 | 63.46 | 56.59 |
| Swin-UNet$^{\dagger}$ [3] | ECCV'22 | × | 82.3 | 67.3 | 78.07 | 68.34 | 85.90 | 77.56 | 63.67 | 55.54 |
| H2Former$^{\dagger}$ [9] | IEEE TMI'23 | × | 33.7 | 24.6 | 77.86 | 68.35 | 80.03 | 72.23 | 63.72 | 56.71 |
| SAM$^{\P}$ [14] | ICCV'23 | × | 93.6 | 50.9 | 71.85 | 56.06 | 77.83 | 70.72 | 49.93 | 33.27 |
| SAM-Adapter$^{\P}$ [4] | ICCV'23 | × | 104.3 | 55.2 | 84.76 | 73.55 | 83.42 | 71.55 | 77.47 | 63.22 |
| ViLT$^{\dagger}$ [13] | ICML'21 | ✓ | 113.1 | 8.1 | 79.63 | 70.12 | 62.44 | 45.39 | 63.51 | 46.53 |
| CLIPSeg$^{\dagger}$ [18] | CVPR'22 | ✓ | 150.0 | 23.0 | 78.92 | 71.55 | 83.71 | 76.02 | 62.06 | 57.91 |
| TGANet$^{\diamond}$ [23] | MICCAI'22 | ✓ | 19.8 | 41.9 | 79.87 | 70.75 | 89.51 | 82.49 | 69.33 | 62.32 |
| Ariadne's Thread$^{\dagger}$ [27] | MICCAI'23 | ✓ | 44.0 | 22.4 | 89.78 | 81.45 | 87.61 | 77.95 | 79.36 | 65.78 |
| LViT$^{\dagger}$ [15] | IEEE TMI'23 | ✓ | 29.7 | 54.1 | 83.66 | 75.11 | 88.62 | 81.90 | 65.51 | 58.73 |
| RecLMIS$^{\dagger}$ [11] | IEEE TMI'24 | ✓ | 23.7 | 24.1 | 85.22 | 77.00 | 85.78 | 78.76 | 63.66 | 55.96 |
| SGSeg$^{\dagger}$ [25] | MICCAI'24 | ✓ | 76.9 | 19.3 | 87.41 | 77.85 | 86.99 | 77.27 | 68.39 | 63.68 |
| VLSM-Adapter$^{\dagger}$ [7] | MICCAI'24 | ✓ | 136.9 | 38.3 | 79.98 | 76.69 | 82.34 | 74.91 | 65.02 | 57.20 |
| **L$^3$Seg (Ours)$^{\dagger}$** | | ✓ | **8.2** | **5.1** | **90.98** | **83.46** | **90.10** | **82.67** | **85.53** | **74.72** |

### 4.2 Implementation Details

All experiments were implemented in PyTorch and executed on an NVIDIA A100 SXM4 GPU with 40GB memory. The training process employed a cosine annealing learning rate schedule, decaying from an initial rate of $3 \times 10^{-4}$ to $1 \times 10^{-6}$. Input images and corresponding masks were resized to 224×224 pixels, with a 10% probability of applying random zoom as data augmentation. A batch size of 32 was used, and the model was optimized using the AdamW optimizer with a compound Dice and Cross-Entropy loss function. Training proceeded for 200 epochs, with early stopping triggered after 50 consecutive epochs without improvement in validation performance. Evaluation metrics included Dice score and mean Intersection over Union (mIoU), where Dice was emphasized due to its sensitivity to small target structures.

### 4.3 Performance comparison with existing methods

L$^3$Seg was evaluated against unimodal and multimodal baselines under identical data splits and training settings, as shown in Table 1. Methods are organized by backbone as CNN-based, SAM-based, and hybrid CNN-Transformer. Among unimodal methods, the hybrid CNN–Transformer Swin-UNet [3] (82.3 M, 67.3 GFLOPs) improves long-range context and outperforms the CNN-based U-Net [22] on Kvasir-SEG and BUSI; these gains come with higher parameter counts and FLOPs. Adapter-based systems present a different trade-off. SAM-Adapter [4] improves over SAM [14] but adds 10.7 M parameters (104.3 M vs. 93.6 M) and additional compute (55.2 GFLOPs vs. 50.9 GFLOPs). For multimodal hybrids, TGANet [23] (41.9 GFLOPs) is strong on Kvasir-SEG, whereas LViT [15] (54.1 GFLOPs) performs better on QaTa-COV19. CLIPSeg [18], built on CLIP [20], maps text prompts to pixel-level masks and uses 150.0 M parameters, making it relatively heavy. Adapter-based multimodal hybrid VLSM-Adapter [7] increase model size and compute (136.9 M, 38.3 GFLOPs). In contrast, L$^3$Seg achieves Dice 90.98% and mIoU 83.46% on QaTa-COV19 [6], Dice 90.10% and mIoU 82.67% on Kvasir-SEG [12], and Dice 85.53% and mIoU 74.72% on BUSI [1] with 8.2 M parameters and 5.1 GFLOPs. Consistently higher scores across all three datasets indicate effective handling of modality-specific challenges, including variation in scale, texture, contrast, and surrounding tissue typical of X-ray, endoscopy, and ultrasound. Qualitative results in Figure 3 show that L$^3$Seg delineates lesions more accurately across modalities than state-of-the-art baselines, reducing both over-segmentation and under-segmentation.

## 5 Ablation Studies

Three ablation studies were conducted to evaluate different aspects of our method: **(i)** replacing (L$^3$) with Standard Linear Layer, **(ii)** varying the amount of training data, and **(iii)** varying the granularity of text prompts.

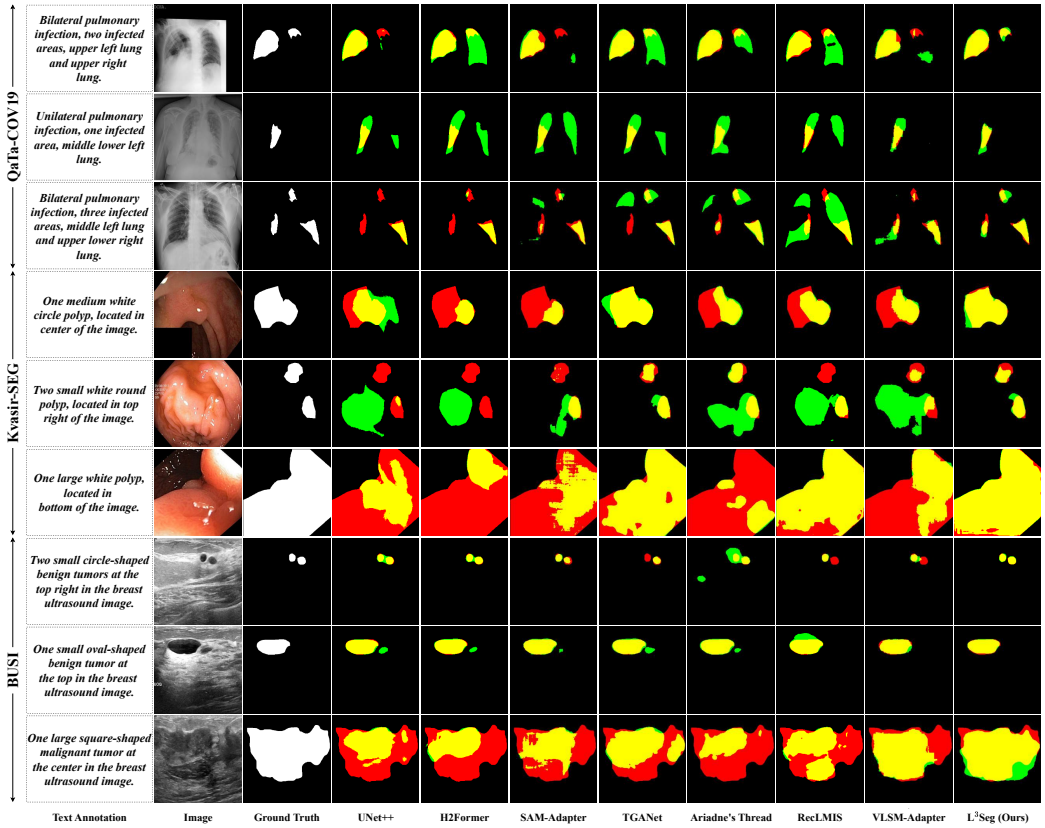



Figure 3: **Segmentation Visualizations on QaTa-COV19, Kvasir-SEG and BUSI dataset.** The overlays show true positives (correctly segmented regions) in yellow, false negatives (under-segmented areas) in red, and false positives (over-segmented areas) in green.

Table 2: **Impact across different projection layers**. (*) frozen Swin-V2 Tiny; (‡) trainable Swin-V2 Tiny.

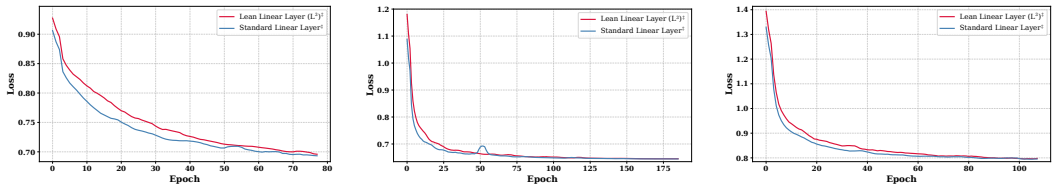
| Layer usage | Params (M) ↓ | FLOPs (G) ↓ | QaTa-COV19 | | Kvasir-SEG | | BUSI | |
|---|---|---|---|---|---|---|---|---|
| | | | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ |
| Standard Linear Layer* | 13.8 | 9.9 | 89.50 | 81.65 | 87.74 | 78.15 | 82.53 | 72.21 |
| Standard Linear Layer‡ | 41.4 | 9.9 | 90.64 | 82.89 | 88.18 | 80.48 | 84.42 | 73.05 |
| **Lean Linear Layer ($L^3$) (Ours)‡** | **8.2** | **5.1** | **90.98** | **83.46** | **90.10** | **82.67** | **85.53** | **74.72** |

## 5.1 Effect of Projection Layer

This section examines the impact of different projection layers on segmentation performance. Table 2 compares three configurations: the Standard Linear Layer with a frozen Swin-V2 Tiny encoder (*), the same layer with a trainable encoder (‡), and the proposed $L^3$ with a trainable encoder (‡). While enabling encoder fine-tuning improves performance over the frozen baseline, replacing the standard layer with the $L^3$ design yields the highest Dice and mIoU scores across all datasets, despite a significant reduction in parameter count from 41.4M to just 8.2M and FLOPs from 9.9G to 5.1G. To assess optimization behavior, Figure 4 reports training loss curves for the standard linear and $L^3$ variants (both with a trainable Swin-V2-Tiny encoder). The trajectories are similar across datasets, suggesting comparable learning dynamics; gains with $L^3$ stem from the projection parameterization rather than training instability or schedule differences.

## 5.2 Varying Text Input

This study examines how text prompt granularity affects segmentation by splitting each annotation into Part A (for example, unilateral or bilateral infection and the number of infected areas) and Part B (positional details). As shown in Table 3, three settings are evaluated: No Text, Part “A”, and Part




(a) QaTa-COV19 (XRay) [6] (b) Kvasir-SEG (Endoscopy) [12] (c) BUSI (Ultrasound) [1]

Figure 4: **Training loss curves for L$^3$Seg.** The curves for both the Standard Linear layer and the Lean Linear layer (L$^3$), each with a trainable Swin-V2 Tiny encoder ($^‡$), exhibit similar patterns, suggesting that both configurations share comparable learning dynamics.

Table 3: **Impact with Varying Text Inputs.**

| Text Usage | QaTa-COV19 | | Kvasir-SEG | | BUSI | |
|---|---|---|---|---|---|---|
| | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ |
| No | 87.62 | 81.06 | 86.05 | 75.51 | 82.08 | 69.61 |
| Part A | 89.73 | 82.03 | 88.69 | 79.68 | 83.47 | 71.63 |
| **Part A + B** | **90.98** | **83.46** | **90.10** | **82.67** | **85.53** | **74.72** |

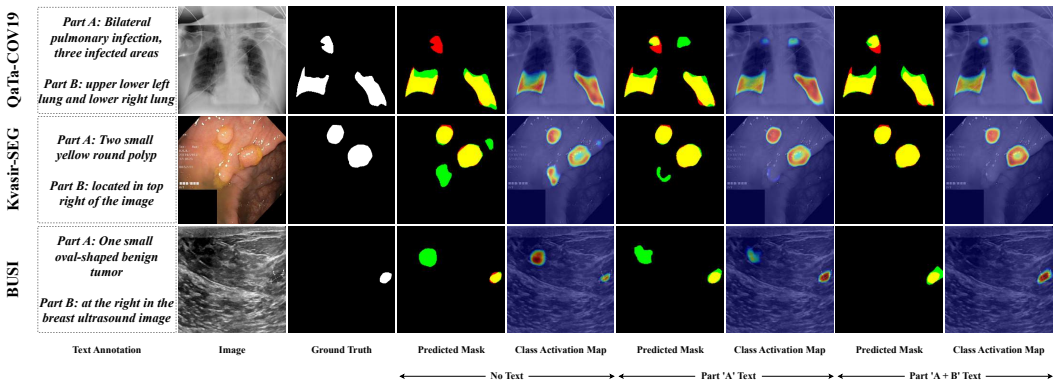



Figure 5: **Segmentation Visualizations with Varying Text Inputs.** Each row presents results using three levels of textual input: No Text, Partial text (Part 'A'), and Complete text (Part 'A + B'). For each setting, both the predicted segmentation mask and the corresponding Class Activation Map are shown. Results are presented for QaTa-COV19 [6] (top row), Kvasir-SEG [12] (middle row) and BUSI [1] (bottom row). Overlays show true positives (correctly segmented regions) in yellow, false negatives (under-segmented areas) in red, and false positives (over-segmented areas) in green. Best viewed in color, and further enhance the view by zooming in.

"A+B". Across QaTa-COV19, Kvasir-SEG, and BUSI, richer textual descriptions, in particular spatial cues, improve accuracy. Even without text, L$^3$Seg is competitive with strong baselines in Table 1, and performance increases when text is provided. Part "A+B" yields the best Dice and mIoU, followed by Part "A", with both surpassing the no-text case. Figure 5 shows the same trend, with the full-text configuration (Part 'A + B') producing more precise masks than partial or absent prompts, and the corresponding activation maps stronger and more localized, reflecting better focus on relevant regions as textual detail increases.

### 5.3 Training Data Size

This study examines how training data size influences segmentation performance. Table 4 compares the unimodal SAM-Adapter [4] and the multimodal VLSM-Adapter [7], both trained on 100% of the available data, with L$^3$Seg trained using 25%, 50%, 75%, and 100% of the training samples. L$^3$Seg surpasses SAM-Adapter even with only 25% of the training data on QaTa-COV19, and 50% on Kvasir-SEG and BUSI. It also outperforms VLSM-Adapter with just 25% of the data on QaTa-COV19 and BUSI, and 75% on Kvasir-SEG. These results highlight the effectiveness of the proposed L$^3$

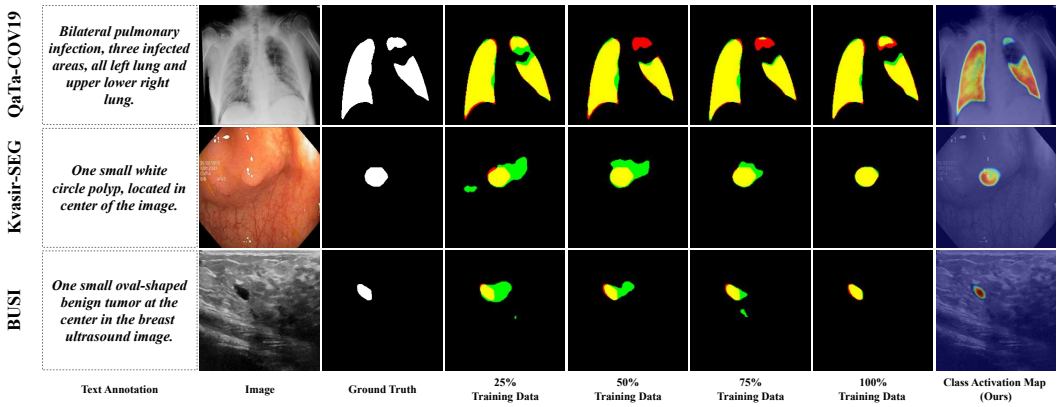



Figure 6: **Segmentation Visualizations of Training Data Size.** Each row compares the L$^3$Seg outputs when trained on 25%, 50%, 75%, or 100% of training data. The final column shows the Class Activation Map corresponding only to the model trained with 100% of training data. The top row shows QaTa-COV19 [6], middle row presents Kvasir-SEG [12] and the bottom row showcases BUSI [1]. Overlays display true positives (correctly segmented regions) in yellow, false negatives (under-segmented areas) in red, and false positives (over-segmented areas) in green. Best viewed in color, and further enhance the view by zooming in.

Table 4: **Impact of Training Data Size.**

| **Data Usage** | **QaTa-COV19** | | **Kvasir-SEG** | | **BUSI** | |
|---|---|---|---|---|---|---|
| | **Dice(%)** ↑ | **mIoU(%)** ↑ | **Dice(%)** ↑ | **mIoU(%)** ↑ | **Dice(%)** ↑ | **mIoU(%)** ↑ |
| SAM-Adapter [4] (100% Training) | 84.76 | 73.55 | 83.42 | 71.55 | 77.47 | 63.22 |
| VLSM-Adapter [7] (100% Training) | 79.98 | 76.69 | 82.34 | 74.91 | 65.02 | 57.20 |
| L$^3$Seg (25% Training) | 86.15 | 77.43 | 83.06 | 72.50 | 77.29 | 62.98 |
| L$^3$Seg (50% Training) | 87.10 | 80.98 | 84.99 | 73.90 | 82.05 | 69.57 |
| L$^3$Seg (75% Training) | 89.59 | 81.80 | 87.96 | 78.50 | 83.61 | 71.83 |
| **L$^3$Seg (100% Training)** | **90.98** | **83.46** | **90.10** | **82.67** | **85.53** | **74.72** |

design and underscore the robustness of text-driven segmentation in data-limited scenarios. Figure 6 provides a qualitative view, showing that L$^3$Seg maintains high segmentation quality even with reduced supervision.

## 6 Conclusion and Future Work

This work presents L$^3$Seg, a compact and effective vision–language segmentation framework. By replacing every dense projection in the encoder and decoder with L$^3$, which freezes pretrained weights and biases and learns a small input-adaptive residual, L$^3$Seg achieves strong accuracy with 8.2 M parameters and 5.1 GFLOPs. Evaluation across three medical imaging tasks (X-ray, endoscopy, ultrasound) shows consistent gains in Dice and mIoU over leading unimodal and multimodal models, and performance remains strong when training labels are scarce due to guidance from clinical text and per-token adaptation. However, gains assume access to pretrained encoder weights (e.g., ImageNet); without such pretraining (training from scratch or on a mismatched modality), benefits can diminish. Performance also depends on text quality and phrasing, where vague or conflicting descriptions can lower boundary accuracy, and very thin or low-contrast targets may still be missed, especially when precise location cues are absent. Future work will scale the L$^3$ design to larger vision–language networks and self-supervised encoders, extend it to 3D imaging and video, and explore applications to classification and detection without architectural changes.

## References


[1] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in Brief*, 28:104863, 2020.

[2] Reza Azad, Ehsan Khodapanah Aghdam, Amelie Rauland, Yiwei Jia, Atlas Haddadi Avval, Afshin Bozorgpour, Sanaz Karimijafarbigloo, Joseph Paul Cohen, Ehsan Adeli, and Dorit Merhof. Medical image segmentation review: The success of u-net. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)*, 2024.

[3] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *17th European Conference on Computer Vision (ECCV)*, pages 205–218. Springer, 2022.

[4] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *19th IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3367–3375, 2023.

[5] Pierre-Henri Conze, Gustavo Andrade-Miranda, Vivek Kumar Singh, Vincent Jaouen, and Dimitris Visvikis. Current and emerging trends in medical image segmentation with deep learning. *IEEE Transactions on Radiation and Plasma Medical Sciences (IEEE TRPMS)*, 7(6):545–569, 2023.

[6] Aysen Degerli, Serkan Kiranyaz, Muhammad EH Chowdhury, and Moncef Gabbouj. Osegnet: Operational segmentation network for covid-19 detection using chest x-ray images. In *29th IEEE International Conference on Image Processing (ICIP)*, pages 2306–2310. IEEE, 2022.

[7] Manish Dhakal, Rabin Adhikari, Safal Thapaliya, and Bishesh Khanal. Vlsm-adapter: Finetuning vision-language segmentation efficiently with lightweight blocks. In *27th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 712–722. Springer, 2024.

[8] Raman Dutt, Linus Ericsson, Pedro Sanchez, Sotirios A. Tsaftaris, and Timothy Hospedales. Parameter-efficient fine-tuning for medical image analysis: The missed opportunity. In *7th International Conference on Medical Imaging with Deep Learning (MIDL)*, 2024.

[9] Along He, Kai Wang, Tao Li, Chengkun Du, Shuang Xia, and Huazhu Fu. H2Former: An Efficient Hierarchical Hybrid Transformer for Medical Image Segmentation. *IEEE Transactions on Medical Imaging (IEEE TMI)*, 42(9):2763–2775, 2023.

[10] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *10th International Conference on Learning Representations (ICLR)*, 2022.

[11] Xiaoshuang Huang, Hongxiang Li, Meng Cao, Long Chen, Chenyu You, and Dong An. Cross-modal conditioned reconstruction for language-guided medical image segmentation. *IEEE Transactions on Medical Imaging (IEEE TMI)*, 2024.

[12] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *25th International Conference on MultiMedia Modeling (MMM)*, pages 451–462. Springer, 2019.

[13] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *38th International Conference on Machine Learning (ICML)*, pages 5583–5594. PMLR, 2021.

[14] 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 *19th IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023.

[15] Zihan Li, Yunxiang Li, Qingde Li, Puyang Wang, Dazhou Guo, Le Lu, Dakai Jin, You Zhang, and Qingqi Hong. Lvit: Language meets vision transformer in medical image segmentation. *IEEE Transactions on Medical Imaging (IEEE TMI)*, 43(1):96–107, 2023.

[16] Xiaohong Liu, Hao Liu, Guoxing Yang, Zeyu Jiang, Shuguang Cui, Zhaoze Zhang, Huan Wang, Liyuan Tao, Yongchang Sun, Zhu Song, et al. A generalist medical language model for disease diagnosis assistance. *Nature Medicine*, pages 1–11, 2025.

[17] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *39th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12009–12019, 2022.

[18] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *39th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, 2022.

[19] Kanchan Poudel, Manish Dhakal, Prasiddha Bhandari, Rabin Adhikari, Safal Thapaliya, and Bishesh Khanal. Exploring transfer learning in medical image segmentation using vision-language models. In *7th International Conference on Medical Imaging with Deep Learning (MIDL)*, 2024.

[20] 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 *38th International Conference on Machine Learning (ICML)*, pages 8748–8763. PmLR, 2021.

[21] Evani Radiya-Dixit and Xin Wang. How fine can fine-tuning be? learning efficient language models. In *23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2435–2443. PMLR, 2020.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *18th International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[23] Nikhil Kumar Tomar, Debesh Jha, Ulas Bagci, and Sharib Ali. Tganet: Text-guided attention for improved polyp segmentation. In *25th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 151–160. Springer, 2022.

[24] Haiyang Wang, Yue Fan, Muhammad Ferjad Naeem, Yongqin Xian, Jan Eric Lenssen, Liwei Wang, Federico Tombari, and Bernt Schiele. Tokenformer: Rethinking transformer scaling with tokenized model parameters. In *13th International Conference on Learning Representations (ICLR)*, 2025.

[25] Shuchang Ye, Mingyuan Meng, Mingjian Li, Dagan Feng, and Jinman Kim. Enabling text-free inference in language-guided segmentation of chest x-rays via self-guidance. In *27th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 242–252. Springer, 2024.

[26] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)*, 2024.

[27] Yi Zhong, Mengqiu Xu, Kongming Liang, Kaixin Chen, and Ming Wu. Ariadne's thread: Using text prompts to improve segmentation of infected areas from chest x-ray images. In *26th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 724–733. Springer, 2023.

[28] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging (IEEE TMI)*, 39(6):1856–1867, 2019.

## A Effect of Projection Layer

Figure 7 shows corresponding qualitative results, where the $L^3$ configuration produces more accurate and refined segmentation masks. These findings highlight the effectiveness of the proposed projection strategy in maintaining high segmentation accuracy while substantially reducing model complexity.

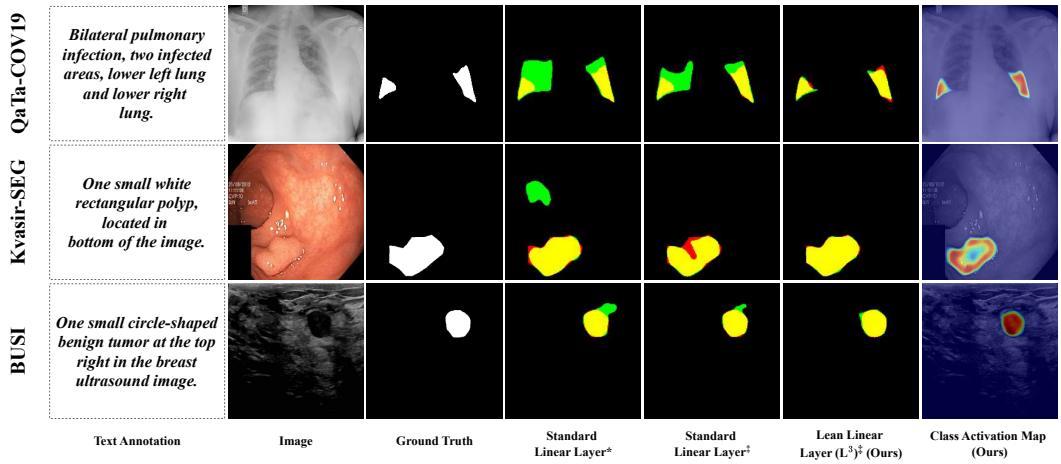



Figure 7: **Segmentation Visualizations across Different Projection Layers.** Each row presents results using three configurations: a standard linear layer with a frozen Swin-V2 Tiny encoder ($^*$), the same linear layer with a trainable encoder ($^\ddagger$), and the proposed $L^3$ with a trainable encoder. The final column displays the Class Activation Map generated using the $L^3$ configuration. Results are shown for QaTa-COV19 [6] (top row), Kvasir-SEG [12] (middle row), and BUSI [1] (bottom row). Overlays show true positives (yellow), false negatives (red), and false positives (green). Best viewed in color; zoom in for details.

## B Effect of PEFT

This section contrasts Lean Linear Layers ($L^3$) with LoRA-based fine-tuning. $L^3$ is applied in both encoder and decoder, reducing trainable parameters while improving accuracy. As shown in Table 5, $L^3$ outperforms the LoRA setting on all datasets, with the largest margin on Kvasir-SEG. The gains hold across datasets with different imaging modalities and text prompts, indicating better adaptability.

Table 5: **Impact of $L^3$ vs. LoRA.**

| Method | Params (M) ↓ | QaTa-COV19 | | Kvasir-SEG | | BUSI | |
|---|---|---|---|---|---|---|---|
| | | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ | Dice(%) ↑ | mIoU(%) ↑ |
| LoRA (applied to projection head) | 13.2 | 90.28 | 82.28 | 86.76 | 76.61 | 83.18 | 72.68 |
| **Lean Linear Layer ($L^3$)** | **8.2** | **90.98** | **83.46** | **90.10** | **82.67** | **85.53** | **74.72** |