The Australian National University
2600 ACT | Canberra | Australia

Australian National University

School of Computing

College of Systems and Society

Weakly Supervised Semantic Segmentation of Histopathology Images

— Honours project (S1/S2 2025)

A thesis submitted for the degree
Bachelor of Advanced Computing (Research and Development)

By:
Felix O'Brien

Supervisor:
Dr. Benjamin Mashford

October 2025

Declaration:

I declare that this work:

- upholds the principles of academic integrity, as defined in the Academic Integrity Rule;
- is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the class summary and/or LMS course site;
- is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

I acknowledge that I am expected to have undertaken Academic Integrity training through the Epigeum Academic Integrity modules prior to submitting an assessment, and so acknowledge that ignorance of the rules around academic integrity cannot be an excuse for any breach.

October (2025), Felix O'Brien

Acknowledgements

I would like to thank my supervisor, Dr. Benjamin Mashford for his consistent support and guidance throughout the duration of this project.

I would also like to thank my mother for tireless support and encouragement through my undergraduate years.

Abstract

Tissue segmentation is a key part of the digital pathology workflow, allowing for the automatic extraction of regions of interest in an image. However, as with most deep learning methods, a significant amount of quality labeled data is required to train the models. This challenge is amplified by the high level of expertise required to annotate the data and the unique characteristics of histopathology images. To address these challenges, this thesis investigates the use of weakly-supervised segmentation approaches as a means to reduce the burden of annotation and thereby increase accessibility to tissue segmentation. Traditional weakly-supervised segmentation approaches can suffer from poor activation as a result of the model favouring the most discriminative features for the classification task. While Class Activation Map (CAM) based approaches often suffer from the problem of partial activation, we propose a novel approach to assist the model in creating more complete activation maps. We apply a pseudo-supervised contrastive loss (PSCL) which improves the activation and convergence of these models. We demonstrate significant increases in performance across domain-standard datasets and achieve particularly strong performance on the BCSS-WSSS dataset with respect to the state-of-the-art.

Table of Contents

Chapter 1

Introduction

Histopathology is the diagnosis of disease through the analysis of tissue samples under a microscope. It is conducted by a medical doctor, known as a pathologist, who examines the tissue sample and provides a diagnosis, often involving the identification of a disease or the extent of a disease. It is still considered the gold standard for the diagnosis of many diseases, the most notable being cancer (Zarella et al., 2018). With the advent of machine learning, and in particular deep learning, there has been a significant increase in the use of machine learning methods for histopathology (Komura et al., 2025). This has been driven by a desire to increase the speed and accuracy of diagnosis, improve consistency of diagnosis and increase the scalability of diagnosis. Whilst various approaches to Computer Aided Diagnosis (CAD) have been developed, assistance in identifying regions of interest in tissue samples closely mimics the human-pathologist's approach to diagnosis. Therefore, semantic segmentation, the identification of regions in an image to be of a particular class, has attracted great interest. Whilst fully supervised semantic segmentation approaches have been shown to produce strong results, they require a significant amount of labelled data, often in the form of per-pixel attributions (Kang et al., 2025). This is in contrast to other tasks, such as classification, which can be performed with only global labels (Ridnik et al., 2021). Given the high level of expertise required, the demand for pathologists on other tasks and the unique characteristics of histopathology images, applying machine learning approaches, particularly those which are segmentation-based, to histopathology is a significant time and financial burden.

The annotation bottleneck is a significant barrier to the development of semantic segmentation methods for histopathology. Some estimates suggest that the annotation process for identifying a tumour region can take up to 5 hours, with over 10,000 clicks (Xu et al., 2022). In recognition of this challenge, significant work has been done to reduce the amount of annotation required for the task. Of particular interest to this work is the use of weak supervision, a supervision context in which reduced annotation is used to train the segmentation task. Applications of such approaches include bounding boxes,

scribbles, point annotations and global labels. In the context of this work, Weakly Supervised Semantic Segmentation (WSSS) refers to the use of global labels which indicate the presence of a certain tissue type in an image to train the segmentation task Han et al. (2021). This significantly reduces the annotation burden of producing sufficient data for training and thereby increases accessibility to machine learning approaches to aid in diagnosis. Some estimates suggest that this could reduce the annotation time by 500x, from 5 hours to 1 minute (Xu et al., 2022). However, weak supervision does not come without drawbacks.

Perhaps the most common implementation of WSSS is through the use of Class Activation Maps (CAMs). CAMs highlight the regions of the feature map which are most discriminative for the classification task by applying the weights of the classification layer to the last feature map of the network, prior to Global Average Pooling (GAP). These CAMs are often then used to generate pseudo-labels for the segmentation task. However, a side effect of weak supervision is that the CAMs suffer from the problem of **partial-activation**, where only the most discriminative regions of the image are activated (Chang et al., 2020; Kang et al., 2025). This is not desirable for the segmentation task as it inherently means that less discriminative regions can be ignored and misclassified, thereby not providing a whole object segmentation. In response to this challenge, significant investment has been made in the development of WSSS approaches, including the use of alternative CAM generation methods such as Grad-CAM (Selvaraju et al., 2019) or prototype methods such as SIPE (Chen et al., 2022a). Despite this, the problem of partial-activation persists and is a significant challenge for the development of WSSS approaches.

As part of a global investigation into the development of WSSS approaches, this work addresses the issue of partial-activation by introducing the Pseudo-Supervised Contrastive Loss (PSCL), a novel loss function that leverages a model's own Class Activation Maps as pseudo-labels to learn a more semantically separable feature space, directly addressing the issue of partial activation. We outline the following contributions:

1. We provide a thorough evaluation of existing weakly-supervised segmentation approaches, including the use of CAMs, prototype methods, contrastive learning and self-supervised learning to the histopathology domain.

2. We propose the Pseudo-Supervised Contrastive Loss (PSCL), a novel loss function that leverages a model's own Class Activation Maps as pseudo-labels to learn a more semantically separable feature space, directly addressing the issue of partial activation.

3. We conduct a thorough evaluation of our method on the domain-standard LUAD-HistoSeg (Bulten et al., 2020) and BCSS-WSSS (Han et al., 2021) datasets, demonstrating significant performance increases compared to baseline approaches.

4. We achieve results competitive with the state-of-the-art, particularly on the BCSS-WSSS dataset, validating the effectiveness of our proposed method in a challenging,

weakly-supervised context.

5. We conduct a thorough evaluation of the wholistic weakly-supervised training approach, examining the motivation for and impact of various components of a Weakly Supervised Semantic Segmentation (WSSS) approach.

We break down this work into the following chapters:

- Chapter 2: Background - This chapter provides a background to the histopathology domain, the digital pathology domain, deep learning and the segmentation task.
- Chapter 3: Related Work - This chapter provides a review of the related work in the field of WSSS, including the use of CAMs, prototype methods, contrastive learning and self-supervised learning.
- Chapter 4: Methods - This chapter provides a detailed description of the methods used in the study, including the implementation of the PSCL approach and the evaluation of the wholistic weakly-supervised training approach.
- Chapter 5: Evaluation - This chapter provides a detailed description of the evaluation metrics used and the results of the study.
- Chapter 6: Conclusion - This chapter provides a conclusion to the study and outlines avenues for future work.

Chapter 2

Background

2.1 Histopathology

2.1.1 Overview

Histopathology is the study of tissue samples under a microscope for the diagnosis of disease and is often regarded as the gold standard for diagnosis of many diseases, included cancer (He et al., 2012). In examining tissue, histopathology lies above cytology (examination of cells) and below radiology (examination of organs and structures) in terms of scale within the body (Gurcan et al., 2009).

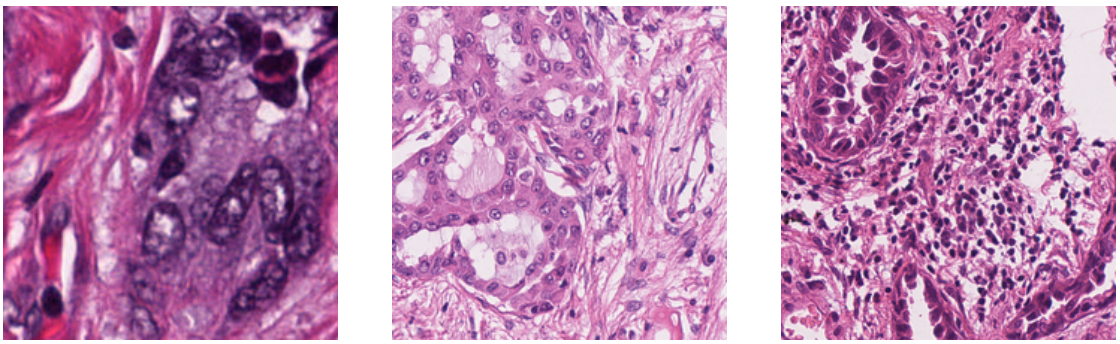

Figure 2.1: Example histopathology patches. (Han et al., 2021)

Histopathology is not simply the identification of disease, but also the assessment of the extent of the disease. This may look like the identification of a tumour region, or the assessment of the severity of a disease such as the grading of a cancer (He et al., 2012). Whilst cancer is perhaps the most well-known disease studied through histopathology, it is routinely used to study other diseases such as:

- **Infectious diseases** such as tuberculosis
- **Inflammatory diseases** such as Crohn's disease
- **Autoimmune diseases** such as coeliac disease
- **Organ-specific diseases** such as endometriosis

Despite the rise in digital pathology, the vast majority of histopathology is still performed manually.

2.1.2 Slide Preparation

In order to prepare a tissue sample for examination on a glass **slide**, there are a number of steps that are required. Moyes (2019) and (Rolls, 2025) outline some key steps:

Fixation Fixation is the process of preserving the tissue sample to prevent against decay via autolysis or putrefaction. Using chemical or physical methods, such as formalin, the tissue is preserved such that it can withstand further processing.

Dehydration Any water in the tissue is removed by a series of increasing concentrations of alcohol.

Clearing The alcohol is removed from the tissue by a solvent such as xylene.

Embedding Embedding is the process of infiltrating the tissue with a solid medium, often paraffin wax. This creates sufficient support to allow the tissue to be sectioned.

Sectioning Sectioning is the process of cutting the tissue into thin slices, such that light can pass through the tissue for examination.

Staining The tissue is stained to highlight particular cell structures or features. The most common stain applied across a broad range of tissues is Hematoxylin and eosin (H&E) stain, which stains the nuclei blue and the cytoplasm pink. (Gurcan et al., 2009)

2.2 Digital Histopathology

In response to the desire to increase accessibility of histopathology, as well as increasingly powerful computer-aided-diagnosis tools in other domains such as radiology, digital histopathology has become increasingly popular (Gurcan et al., 2009). In digital Histopathology, tissue samples are scanned using a microscope to produce a digital image also known as a Whole Slide Image (WSI). WSIs are produced through multiple tiles or lines of tissue that are digitally stitched together (Zarella et al., 2018). Scanning occurs at various magnifications, with scanning at x20 magnification being common for standard viewing, with some applications requiring x40 magnification to resolve more detail. Scanners offer up to x100 magnification.

Given the incredible magnification at which a WSI is scanned, the resulting image is often in the order of gigapixels; a $1mm^2$ area of tissue at x40 magnification is approximately 48 MB to store. (Orchard Software, 2025; Zarella et al., 2018). Compression of WSIs is therefore necessary to make them more manageable, however lossy compression methods such as JPEG can introduce artefacts which can affect the interpretation of the image. As such, a balance must be struck between the size of the WSI and the quality of the image. Even with compression, WSIs often exceed 1GB in size. As such, some images are stored in a pyramid-style format, where multiple downsampled versions of the image are stored, allowing for pre-rendering of images at different magnifications and thereby reducing the time required to render an image (Zarella et al., 2018).

Another issue with WSIs is they are susceptible to variability between scanners, known as inter-scanner variability. Scanners, hardware and settings can vary significantly across manufacturers and thus affect the quality of the image produced. Whilst human readers can often adjust to variability, computer vision models are not as robust. (Ryu et al., 2025)

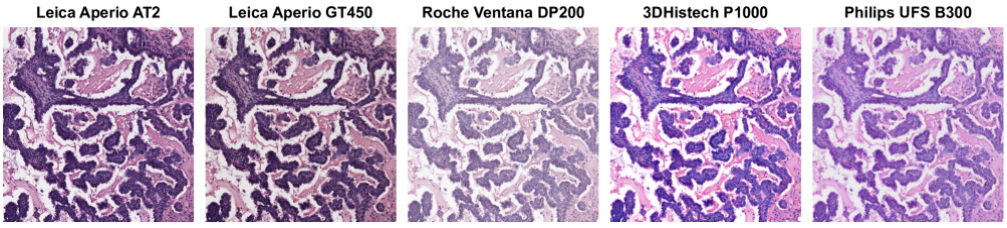

Figure 2.2: Inter-scanner variability across five different scanners. (Ryu et al., 2025)

Given the large size of WSIs, there exist significant challenges in developing annotations that are necessary for training deep learning models. A worst case study by Xu et al. (2022) found that the identification of a tumour region in a WSI required 5 hours of annotation, with over 10,000 clicks. Such a region can consist of up to 8000 vertices. The same study found that the use of weaker annotation strategies for a WSI required around 1 minute per case, a 500x reduction in the time required to annotate a WSI (Xu et al., 2022).

Furthermore, there also exist large challenges concerning the consistency of annotations across different annotators. Identified regions can differ significantly in area (up to 46%) across annotators (Marrón-Esquivel et al., 2023), leading to inconsistent pixel-level annotations. Whilst global annotations can still differ between annotations, there is generally much higher agreement (Cohen's Kappa > 0.9) between annotators than pixel-level annotations (Bulten et al., 2020).

2.2.1 Machine Learning for Histopathology

Whilst a relatively new field in histopathology's long history, the application of machine learning methods for histopathology has seen immense growth in recent years. Komura et al. (2025) find an over 8 fold increase in the number of publications per year between 2018 and 2024.

We can currently categorise the majority of machine learning methods for histopathology into three main categories:

- **Computer-assisted diagnosis (CAD)**
  CAD is the use of ML approaches to assist pathologists in their diagnosis. Examples include image classification and segmentation tasks.
- **Predicting or Discovering Clinicopathological Relationships**
  This includes tasks such as predicting survival or recurrence, based on pathologic features.
- **Virtual Staining**
  Virtual staining is the replacement of the traditional staining process with ML approaches. Typically, this involves using a model such as a General Adversarial

Network (GAN) (Goodfellow et al., 2014) to generate a staining image from a histopathology image.

It is important to underscore there exist a number of unique challenges of applying ML-based CAD approaches to histopathology. Komura et al. (2025) outline the four most common and significant challenges in adapting ML-based CAD approaches to histopathology:

Large Image Sizes As mentioned, WSIs can be gigapixel-sized which imposes memory limits in many applications and typically necessitates division of the image into patches for analysis.

Insufficient labeled data Labelled histopathological images are scarce in nature. One reason already outlined is the time-intensive nature of expert annotation, but additional issues arise given the privacy concerns around medical data. To address these concerns, production of large publicly available datasets and the development of learning approaches to limited labeled data are two avenues which have seen significant interest and which we explore in this work.

Multidimensional analysis Histopathology does not exist in a vacuum. External information such as patient outcomes, reports, additional tissue samples or imaging are all dimensions of data which can assist in the understanding of disease processes but necessitate more complicated analysis techniques.

Domain shifts across institutions The digital histopathological process is susceptible to variation across samples and institutions due to the multiple stages required to prepare a sample, from surgery to scanning. Factors such as time to fixation, chemical concentration and differences in stains can introduce variations which can negatively impact ML outcomes. Coupled with the aforementioned variation across scanners and even annotators, such shifts necessitate approaches that can generalise well.

These challenges provide significant motivation for the development of weakly-supervised approaches to histopathology.

2.3 Metric Learning

Metric learning involves learning distance or similarity metrics between samples in a data space based on the principle that such measures can be used to reveal the underlying structure of the data. It has applications in a range of fields, including computer vision, natural language processing, and recommender systems.

2.3.1 Metric Spaces

Fundamental to metric learning is the concept of a metric space. A metric space provides a notion of distance between points in the space.

Definition 1. A metric space is a set M equipped with a real-valued function $D(a, b)$ defined for all $a, b \in M$ which satisfies the following properties:

1. Positivity: $D(a, b) \geq 0 \forall a, b \in M$
2. Symmetry: $D(a, b) = D(b, a) \forall a, b \in M$
3. Triangle inequality: $D(a, b) \leq D(a, c) + D(c, b) \forall a, b, c \in M$

A common notion of distance is the Mahalanobis distance, which is a measure of the distance between two points in a high-dimensional space.

$$D(a, b) = \sqrt{(a - b)^T M (a - b)} \tag{2.1}$$

Where M is a positive semi-definite matrix. The case when $M = I$ is the Euclidean distance.

An alternative notion of distance popular in the field of metric learning is the cosine distance, which is a measure of the angle between two vectors.

Definition 2. The cosine similarity between two vectors a and b is defined as:

$$D(a, b) = \frac{a \cdot b}{\|a\| \|b\|} \tag{2.2}$$

The cosine similarity is a measure of the angle between two vectors, and is particularly useful for high-dimensional data. Inherently, the cosine similarity varies between -1 and 1, with 1 indicating that the two vectors are identical, and 0 indicating that the two vectors are orthogonal.

2.3.2 Important Algorithms in Metric Learning

K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a non-parametric classification and regression algorithm that makes predictions based on the k closest training examples in the feature space. The algorithm assumes that similar instances are likely to have similar labels.

Formal Definition Given a training dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^d$ are feature vectors and y_i are corresponding labels, and a distance metric $d(\cdot, \cdot)$, the k-nearest neighbours of a query point x_q are defined as:

$$N_k(x_q) = \{x_{(1)}, x_{(2)}, \ldots, x_{(k)}\} \tag{2.3}$$

where $x_{(i)}$ is the i-th nearest neighbour to x_q based on the distance metric d.

Algorithm The KNN algorithm proceeds as follows:

1. **Distance Computation**: For a query point x_q, compute the distance to all training points:
$$d_i = d(x_q, x_i) \quad \forall i \in \{1, 2, \ldots, n\} \tag{2.4}$$

2. **Neighbour Selection**: Select the k training points with smallest distances:
$$N_k(x_q) = \arg \min_{S \subseteq D, |S|=k} \sum_{x_i \in S} d(x_q, x_i) \tag{2.5}$$

3. **Prediction**: For classification, predict the majority class among k neighbours:
$$\hat{y} = \arg\max_c \sum_{x_i \in N_k(x_q)} \mathbb{I}(y_i = c) \tag{2.6}$$
For regression, predict the average of k neighbours:
$$\hat{y} = \frac{1}{k} \sum_{x_i \in N_k(x_q)} y_i \tag{2.7}$$

Properties

- **Lazy Learning**: No explicit training phase; all computation occurs at prediction time
- **Non-parametric**: Makes no assumptions about the underlying data distribution
- **Memory-based**: Requires storing the entire training dataset
- **Sensitivity to k**: Choice of k affects bias-variance tradeoff; smaller k increases variance, larger k increases bias
- **Curse of Dimensionality**: Performance degrades in high-dimensional spaces due to distance concentration

The KNN algorithm is fundamental in metric learning as it directly relies on distance metrics, making it an ideal candidate for evaluation of learned distance functions.

K-Means

K-Means is a clustering algorithm that aims to partition a collection of samples into a fixed number (k) of clusters. The algorithm works by iteratively updating cluster centroids to minimize the within-cluster sum of squares (WCSS).

Formal Definition Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ where each $x_i \in \mathbb{R}^d$, k-means seeks to find k cluster centroids $\mu_1, \mu_2, \ldots, \mu_k$ and cluster assignments $c_1, c_2, \ldots, c_n$ (where $c_i \in \{1, 2, \ldots, k\}$) that minimize the objective function:

$$J = \sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2 \tag{2.8}$$

Algorithm The k-means algorithm proceeds as follows:

1. **Initialization**: Randomly select k initial centroids $\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_k^{(0)}$
2. **Assignment Step**: For each data point x_i, assign it to the nearest centroid:

$$c_i^{(t)} = \arg\min_j \|x_i - \mu_j^{(t)}\|^2 \tag{2.9}$$

3. **Update Step**: Update each centroid to be the mean of all points assigned to it:

$$\mu_j^{(t+1)} = \frac{1}{|C_j^{(t)}|} \sum_{x_i \in C_j^{(t)}} x_i \tag{2.10}$$

   where $C_j^{(t)} = \{x_i : c_i^{(t)} = j\}$ is the set of points assigned to cluster j at iteration t.
4. **Convergence Check**: Repeat steps 2-3 until convergence (no changes in assignments or centroids) or maximum iterations reached.

Properties

- **Convergence**: The algorithm is guaranteed to converge to a local minimum of the objective function
- **Sensitivity to Initialization**: The algorithm is sensitive to initial centroid placement and may converge to different local minima

The k-means algorithm is widely used in metric learning as a baseline clustering method and can be extended to work with learned distance metrics instead of Euclidean distance.

2.4 Deep Learning

Deep Learning (DL) is a subfield of machine learning that makes use of artificial neural networks (ANNs) to model complex relationships between input and output data.

2.4.1 Artificial Neural Networks

Perceptron

The artificial neural network (ANN) is a type of machine learning model inspired by the structure and function of the human brain. The simplest form of an ANN is the perceptron, a single layer neural network with a single output node used to perform binary classification (Rosenblatt, 1958).

Each node in the network is called a neuron and contains a weight and a bias. The weights are used to scale the input to the neuron whilst

$$y = (\sum_{i=1}^{n} w_i x_i) + b \tag{2.11}$$

Where w_i is the weight of the ith input, x_i is the ith input, and b is the bias. The output of the perceptron is then passed through an activation function to produce the final output. In the case of the perceptron, the activation function is 1 if the output is greater than 0, otherwise 0. As a linear model, the perceptron is only able to classify data that is linearly separable, i.e there exists a hyperplane that can separate the two classes.

Activation Functions

In order to model more complex relationships between the input and output data of a perceptron, and ANNs in general, non-linear activation functions are used. Without non-linear activation functions, the perceptron is only able to model linear relationships between the input and output data (Rosenblatt, 1958).

Kruse et al. (2013) provides a comprehensive overview of the most common activation functions used in ANNs, including the sigmoid, softmax, and the ReLU:

- **Sigmoid (Logistic) Activation Function:**

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.12}$$

  The sigmoid function squashes input values to the range $(0, 1)$ and is often used for binary classification tasks.

- **Softmax Activation Function:**

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{2.13}$$

The softmax function is mainly used in the output layer of a classifier to represent multi-class probabilities that sum to 1, where K is the number of classes.

- **Rectified Linear Unit (ReLU) Activation Function:**

$$\text{ReLU}(x) = \max(0, x) \tag{2.14}$$

The ReLU function outputs zero if the input is less than zero, and outputs the input directly otherwise. It is widely used due to its computational simplicity and effectiveness in training deep networks.

Multi-Layer Perceptron

Whilst the perceptron can solve simple problems, it is limited in its inability to handle non-linearly separable data as demonstrated by its inability to solve the XOR problem (Minsky and Papert, 1969). To address this, stacking multiple perceptrons together in a multi-layer perceptron (MLP) allows the model to learn more complex relationships between the input and output data. In an MLP, each layer is connected to the next layer. The first layer is called the **input layer**, the last layer is called the **output layer**, and the layers in between are called **hidden layers**.

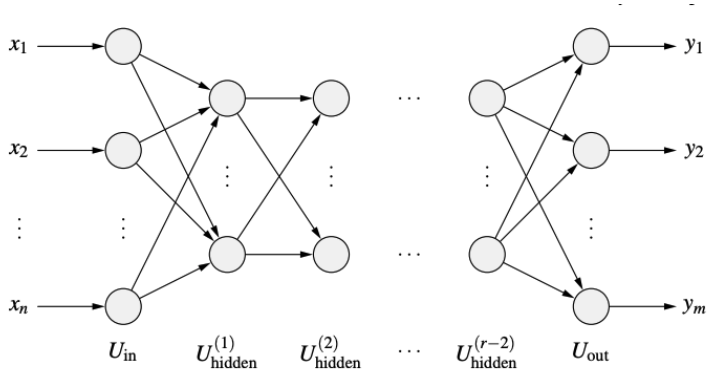

Figure 2.3: A simple Multi-Layer Perceptron (MLP). Each neuron in one layer is typically fully connected to every neuron in the next layer. Adapted from (Kruse et al., 2013).

Generally as the number of hidden layers increases, so too does the ability of the model to model complex relationships between the input and output data: more hidden layers may enable the same approximation quality with significant fewer neurons (Kruse et al., 2013). Importantly, with sufficient hidden layers, ANNs are universal approximators, meaning that they can model any continuous function to any degree of accuracy (Hornik et al., 1989). As such, if a task can be modelled as a continuous function, it is possible to model it with an ANN. This makes ANNs uniquely powerful for a wide range of tasks.

2.4.2 Gradient-Based Learning

Designing or building an ANN is only the first step in applying machine learning to a task. We need to train the model on the task at hand. In order to train a neural network, we must produce an error measure which measures the performance of the model on the desired task. This is typically referred to as a **loss function**.

Loss Functions

Broadly, a loss function measures the difference between the predicted output of the model and the true output. Generally, we define the loss function in terms of the model parameters θ, which typically measures how well a model is able to fit the data (Goodfellow et al., 2016).

$$L(\theta) = \sum_i l(f_\theta(x_i), y_i) + R(\theta)$$

where $f_\theta(x_i)$ is the model's prediction for the ith input, y_i is the true output for the ith input, and $R(\theta)$ is a regularisation term.

We optimise the loss function with respect to the model parameters θ.

$$\theta^* = \arg\min_\theta L(\theta)$$

As an example, a simple loss function is the mean-squared error (MSE), which measures the average squared difference between the true and predicted outputs (Kruse et al., 2013).

$$\text{MSE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.15}$$

Common loss functions for classification tasks, such as the pixel-level segmentation explored in this work, are the cross-entropy loss and the binary cross-entropy loss. Both maximise the similarity between the data's probability distribution and the model's probability distribution (Kruse et al., 2013).

$$\text{Cross-entropy}(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \tag{2.16}$$

$$\text{Binary Cross-entropy}(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{2.17}$$

Backpropagation

Backpropagation is an efficient algorithm for computing the gradient of the loss function with respect to the weights of the model first proposed by Rumelhart et al. (1986). This algorithm sees the application of the chain rule to compute the gradient of the loss function with respect to the weights at each layer of the model. Consider the case of a single layer perceptron with a sigmoid activation function.

$$\hat{y} = \sigma(w^T x + b)$$

where w is the weight vector, x is the input vector, and b is the bias. We can compute the gradient of the loss in terms of the weights by applying the chain rule.

$$\begin{aligned}\frac{\partial L}{\partial w} &= \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w} \\ &= \frac{\partial L}{\partial \hat{y}} \left(\sigma(w^T x + b)(1 - \sigma(w^T x + b))x\right)\end{aligned}$$

By computing the gradient of the loss function with respect to each parameter in the model we can perform **gradient descent** on the model parameters to minimise the loss function.

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \tag{2.18}$$

Where θ_t is the parameters of the model at time t, η is the learning rate, and $\nabla L(\theta_t)$ is the gradient of the loss function with respect to the weights of the model at time t.

Stochastic Gradient Descent

(Goodfellow et al., 2016) outline perhaps the most common algorithm used to train a model is that of stochastic gradient descent (SGD) which updates the weights of the model in the direction of the negative gradient of the loss function with respect to a selection of data, rather than the entire dataset. This selection of data is often referred to as a **minibatch**. SGD employs a **learning rate** parameter, which is a hyperparameter that controls the step size by which the weights are updated. The learning rate is typically chosen to be small enough to ensure that the model converges to a global minimum, but large enough to ensure that the model is able to learn the data. A simple SGD update rule is given by:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla L(f(x^{(i)}; \theta_t), y^{(i)}) \tag{2.19}$$

where m is the batch size, $x^{(i)}$ and $y^{(i)}$ are the ith input and label in the minibatch, and the gradient is averaged over the minibatch.

Momentum is a technique to smooth the gradient descent by introducing a moving average of past gradients into the gradient update (Goodfellow et al., 2016).

The learning rate does not necessarily have to be fixed during the training process; a **scheduler** is a function that is used to alter the learning rate of the model over time. An example of a scheduler is a simple scheduler that linearly decreases the learning rate over time.

$$\eta(t) = \eta_0 \left(1 - \frac{t}{T}\right) \tag{2.20}$$

Where η_0 is the initial learning rate, t is the current epoch, and T is the total number of epochs.

In this work we make use of a Polynomial Decay scheduler, which is a scheduler that decays the learning rate according to a polynomial function.

$$\eta(t) = \eta_0 \left(1 - \frac{t}{T}\right)^p \tag{2.21}$$

Where η_0 is the initial learning rate, t is the current epoch, T is the total number of epochs, and p is the power of the polynomial.

2.4.3 Empirical Techniques

In practice, there are a range of empirical techniques that are used to improve the performance of a model.

Data Augmentation

Data augmentation is a technique used to artificially increase the size of a dataset and thereby increase the ability of a model to generalise by applying transformations to the data. It increases the amount of unique data that the model is exposed to and can act as a regularisation technique, preventing the model from memorising the training data. In the case of image data, this often looks like apply transformations to the image such as cropping, rotation, scaling, and flipping.

Optimisers

Optimisers are algorithms used to alter how the gradient of the loss function is applied to the weights of the model. There are many other optimisers, such as Adam (Kingma and Ba, 2017), and AdamW (Loshchilov and Hutter, 2019) which are significantly more complex with respect to the learning rate schedule.

Pre-training

Pre-training is a technique used to improve model performance by training on a large, often more general, dataset before fine-tuning on a smaller, more task-specific dataset.

This has been demonstrated to significantly improve performance when compared to random initialisation of the model weights (Ridnik et al., 2021).

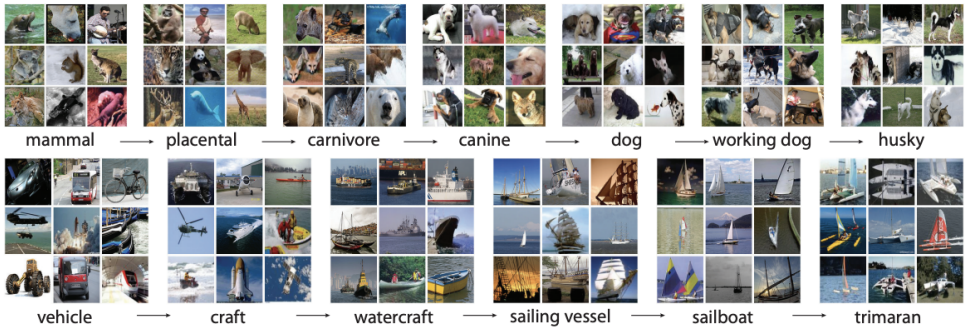

Figure 2.4: Depiction of the ImageNet dataset. Source: (Deng et al., 2009a).

In the computer vision domain, perhaps the most common pre-training dataset is ImageNet (Deng et al., 2009a). ImageNet is a dataset of over 1.2 million images categorised into 1000 classes. It is a large, general dataset that is often used to pre-train models for the classification task. Another popular pre-training dataset is ImageNet-21k (Ridnik et al., 2021), which is a larger version of ImageNet with 21,841 classes.

2.4.4 Supervision in Deep Learning

Returning to the notion of loss functions, we can now consider the use of labelled data to train a model. The **supervision** of a learning task refers to the amount and type of labelled data that is available to the model. Across the literature there are two broad contexts of supervision (Goodfellow et al., 2016):

- **Fully Supervised**: The model is trained on a dataset with fully labelled data.
- **Unsupervised**: The model is trained on a dataset with no labelled data.

As an example, in a fully-supervised setting labels might consist of the class of the image, the bounding box of the object in the image, or the segmentation mask of the object in the image. In an unsupervised setting, the goal is to learn the distribution of the data. For example, clustering data using K-means or other probabilistic models is an example of an unsupervised context.

We can further clarify these contexts by considering the context in between fully supervised and unsupervised. This is known as **semi-supervised** learning (SSL) (Chapelle et al., 2005), where the model has access to some supervision but may not necessarily have access to all the labels or the labels may not be directly relevant to the task at hand. Most relevant to this work, the latter context, where the labels are not directly relevant to the task at hand, is known in the literature as **weakly-supervised** learning (WSL) (Kang et al., 2025; Han et al., 2021).

2.4.5 Classification Metrics

Classification is one of the most fundamental tasks in machine learning, and as such, there are a range of metrics used to evaluate the performance of a classification model. Each of the following metrics evaluates different aspects of the performance of a classification model.

Accuracy Accuracy is the most basic metric for evaluating the performance of a classification model. It is defined as the number of correctly classified samples divided by the total number of samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.22}$$

Where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

Precision Precision is the ratio of true positives to the total number of samples classified as positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.23}$$

Recall Recall is the ratio of true positives to the total number of samples that are actually positive.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.24}$$

F1 Score The F1 score is the harmonic mean of precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.25}$$

2.4.6 Convolutional Neural Networks

A convolutional neural network (CNN) is a type of ANN which incorporates **convolutional layers** alongside traditional fully connected layers (i.e MLPs) that are particularly useful for image data. In CNNs, the convolutional layers are used to extract features from the image, which are then passed through to the MLPs to produce the final output. CNNs have long been the dominant architecture for image processing tasks, and are still widely used in a range of applications.

Convolution

We start by exploring a convolution as an example of the mathematical operation that is used in CNNs. A convolution is a mathematical operation involving two functions, f and g, to produce a third function, h. Goodfellow et al. (2016) defines a convolution as:

A convolution is a mathematical operation involving two functions, f and g, to produce a third function, defined as:

$$h(x) = \int f(t)g(x-t)dt \tag{2.26}$$

Where $f(t)$ is the input, $g(x-t)$ is the **kernel**, and $h(x)$ is often referred to as the **feature map**.

In the context of CNNs, as the image is discrete and two-dimensional and thus the filter is a small matrix of weights that are applied to the input image in a sliding window manner and the kernel is two-dimensional.

$$h(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)g(x-i,y-j) \tag{2.27}$$

Where m and n are the dimensions of the filter (Goodfellow et al., 2016).

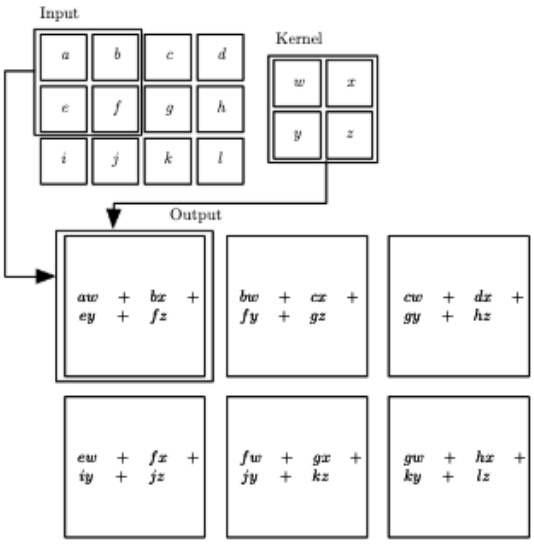

Figure 2.5: Example of convolution operation using a filter (kernel) sliding over an image. The output is a feature map. (Goodfellow et al., 2016)

Convolutions can be performed with different strides, which is the step size at which the filter is applied to the input image. Convolutions may also be dilated, which is the process of adding spacing between the filter weights in order to create a larger effective filter whilst maintaining the same number of parameters. Convolutional layers are layers where the parameters correspond to kernels, rather than a linear set of weights. In this way, the model can learn parameters that correspond with spatial information in an

image, to extract image features and characteristics. These features are then used by the fully connected layers to produce the final output.

Pooling

Another important operation in CNNs is that of pooling. Pooling is a technique used to reduce the spatial resolution of feature maps. There are two main types of pooling: **max pooling** and **average pooling**. Average pooling takes the average value of the area of the feature map that is being pooled. Max pooling takes the maximum value of the area of the feature map that is being pooled.

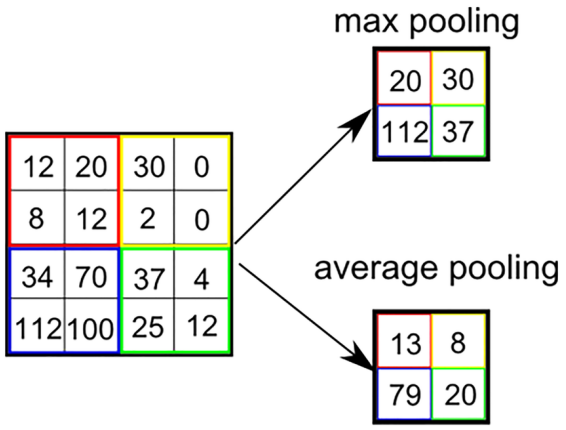

Figure 2.6: Max and average pooling (Buckner, 2019)

Global average pooling (GAP) and global max pooling (GMP) are special cases of pooling where the entire feature map is pooled. In particular, GAP is traditionally used after the final convolutional layer within a CNN to produce a fixed-length vector representation of the input image from which a linear (classification) layer can be applied to produce the final output.

Residual Networks (ResNets)

Residual Networks (ResNets) proposed by He et al. (2015) are a fundamental CNN architecture that are used across a range of tasks and are often used as backbone models for segmentation tasks (Kang et al., 2025; Chang et al., 2020; Cheng et al., 2022). The key component of the ResNet architecture is the **residual block**, which is a block that is used to skip the forward pass of the network and add the input to the output. The residual block increases stability of large deep networks by mitigating the problem of vanishing gradients. Figure 2.7 illustrates the residual block.

He et al. (2015) introduce various standard depths, denoted by the number of layers in the network, including ResNet18, ResNet34, ResNet50 and ResNet101.

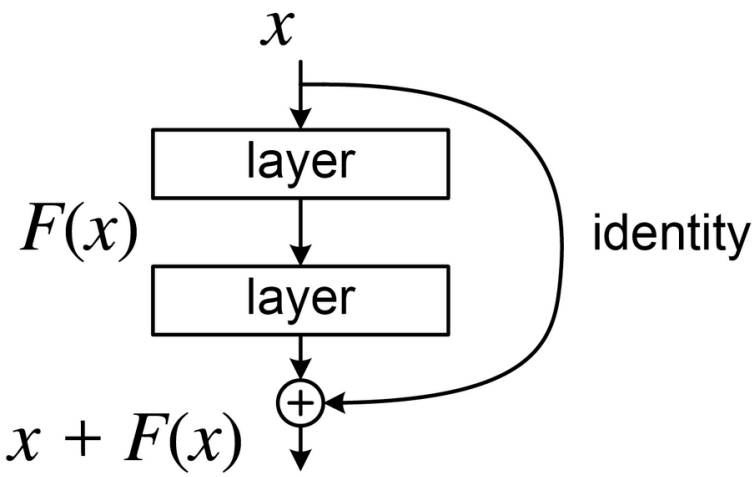

Figure 2.7: Illustration of the residual block. (He et al., 2015)

2.4.7 Transformers

First introduced by Vaswani et al. (2023), the Transformer is a neural network architecture that is designed to process sequential data in parallel in response to the limitations of recurrent neural networks (RNNs). The key component of the Transformer is the **self-attention mechanism**, which allows the model to attend to different parts of the input sequence in parallel.

Transformer Architecture

The transformer architecture consists of two main components: the encoder and the decoder. The encoder is responsible for encoding the input sequence into a sequence of feature vectors, and the decoder is responsible for decoding the feature vectors into a sequence of output tokens. The encoder and decoder are both composed of a stack of identical layers known as **transformer blocks**, which are themselves composed of a self-attention mechanism and a feed-forward network (MLP). This architecture is shown in Figure 2.8.

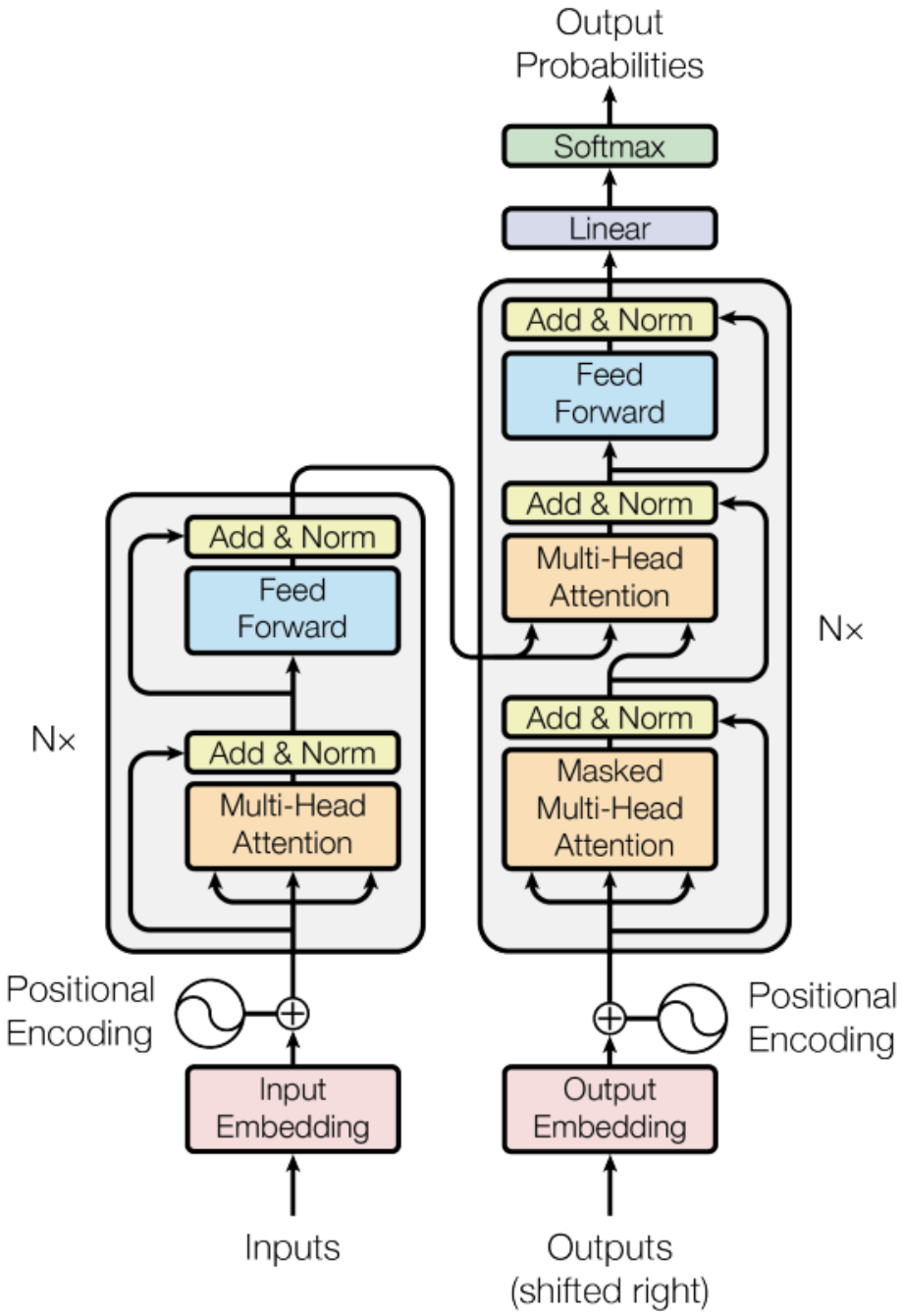

Figure 2.8: Illustration of the transformer architecture. (Vaswani et al., 2023)

Attention Mechanism

The key component of the transformer is the **self-attention mechanism**, which allows the model to attend to different parts of the input sequence in parallel. Vaswani et al. (2023) introduces the self-attention mechanism and its application to the transformer architecture. The attention mechanism consists of three main parameters, the **query**, **key**, and **value** matrices. The query and key matrices are used to compute the attention weights, and the value matrix is used to compute the weighted sum of the value vectors. The attention weights are computed by taking the dot product of the query and key matrices, and then applying a softmax function to the result.

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.28}$$

Where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the dimension of the key and query matrices. The Q K and V matrices are computed by multiplying the input sequence by three learned matrices, W^Q, W^K, and W^V.

The attention mechanism is applied to the input sequence in parallel, and the output is a weighted sum of the value vectors, where the weights are determined by the similarity between the query and key vectors. Importantly, attention is applied globally: each token in the input sequence is attended to by every other token in the sequence. Multi-head attention introduces multiple attention heads to the mechanism, each with its own set of query, key, and value matrices. The output of the multi-head attention is the concatenation of the outputs of the individual attention heads. This allows the model to attend to different parts of the input sequence in parallel, and is particularly effective for capturing global context.

Vision Transformers

Introduced by Dosovitskiy et al. (2021), Vision Transformers (ViTs) adapt the transformer architecture to the image domain, largely by treating an image as a sequence of tokens, akin to the use of the transformer in natural language processing. The ViT architecture consists of a transformer encoder with a classification head appended to the end. Images are split into a grid of patches, and each patch is passed through a linear projection layer to produce a token. The full architecture is Figure 2.9.

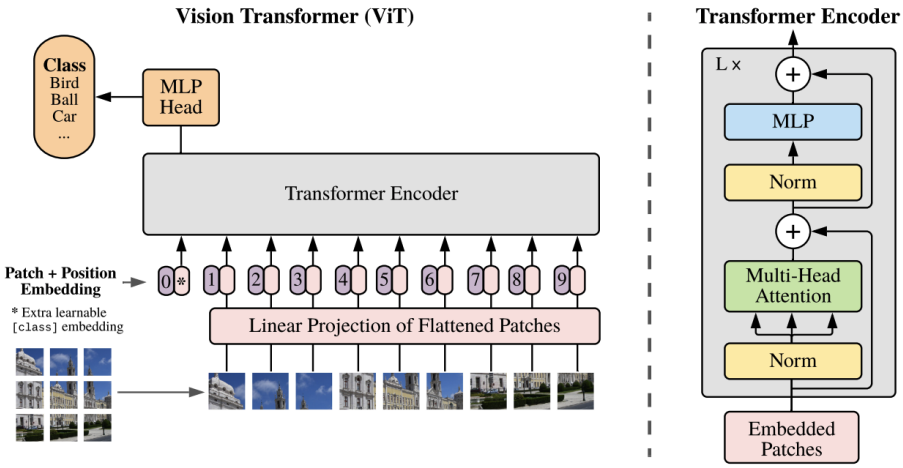

Figure 2.9: Illustration of the Vision Transformer architecture. (Dosovitskiy et al., 2021)

ViTs have been shown to approach or exceed the performance of state-of-the-art CNNs on a range of tasks, including image classification. Although, it is important to note that they generally require a greater amount of data to train in order to produce such results when compared to non-transformer based models (Dosovitskiy et al., 2021; Liu et al., 2021).

Inherently, converting an image into a sequence of tokens removes the spatial information from the image, and as such, position encodings, fixed or learned, are added to the patch embeddings. The broad benefit of ViTs is that they consider the global receptive field of the image from the start by means of the self-attention mechanism. This is in contrast to CNNs, which are inherently local, and require multiple layers in order to capture the global context of the image.

The standard ViT architecture maintains a single, constant feature resolution throughout all its layers which makes it less suitable for the semantic segmentation task. To address this, Hierarchical Vision Transformers (HViTs), such as the Swin Transformer (Liu et al., 2021) or HIPT (Chen et al., 2022b) use a hierarchical architecture to capture both local and global information.

Swin Transformers Despite the success of ViTs in the image domain, they can suffer performance degradation in low-data settings (Caron et al., 2021) and miss out on local features that CNNs are adept at capturing (Li et al., 2023). In response to such difficulties, various modifications to the ViT architecture to address these issues have been proposed. The Swin Transformer, is a modified Vision Transformer (ViT) that uses shifted window self-attention, similar to convolutional layers, to introduce hierarchical structure to the ViT (Liu et al., 2021). As the layers progress, the spatial resolution of

the feature maps is reduced whilst the embedding dimension is increased. This is shown in Figure 2.10.

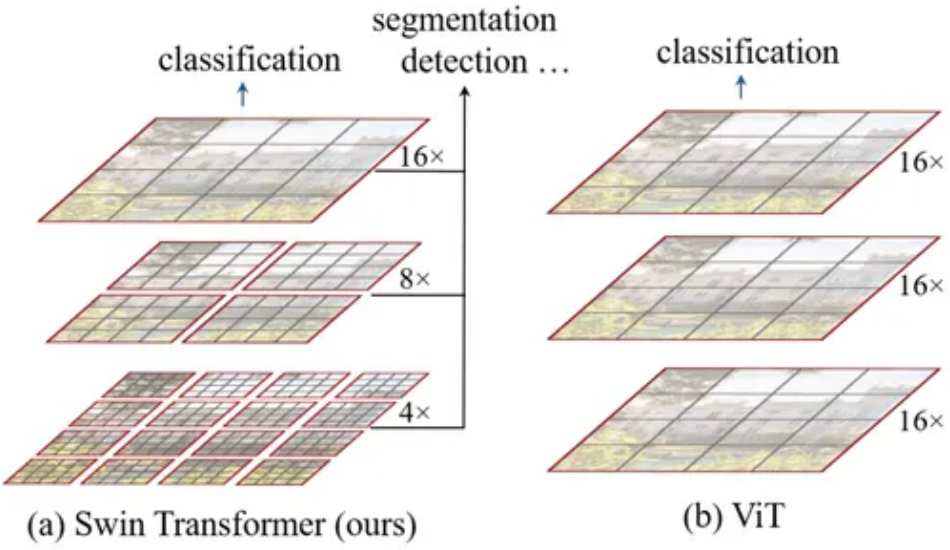

Figure 2.10: Illustration of the Swin Transformer architecture, showing the hierarchical design and shifted window self-attention mechanism. (Liu et al., 2021)

Swin Transformers of all sizes outperform Resnet101 (He et al., 2015) when used as the backbone model for semantic segmentation (Liu et al., 2021). Furthermore, they do so with significantly fewer parameters. The hierarchical nature of the Swin Transfomer also means they can often be used a drop-in replacement for CNNs in segmentation tasks such as the MaskFormer (Cheng et al., 2022).

ConvNeXt (Liu et al., 2022) introduce the ConvNeXt family of models which "modernify" the ResNet architecture by replacing ResNet components with their transformer counterparts, moving towards the structure of the Swin Transformer. Changes include: introduction of a patch embedding layer, larger kernel sizes and depth-wise convolutions which mimic self-attention. Whilst this approach has been demonstrated to outperform the Swin Transformer on the ImageNet dataset in some contexts, it has not been as popular as a backbone model for semantic segmentation.

2.4.8 Deep Metric Learning

Building from the idea of metric learning, deep metric learning applies deep learning techniques to learn a metric space from a set of data. Much of traditional deep learning principles apply, with changes being focused on the task at hand. This is primarily achieved by the use of loss functions which aim to ensure certain properties of the embedding space are satisfied.

Loss Functions

Given that the separability of an embedding space is not a target of the loss functions, only indirectly influenced, the loss functions used in deep metric learning instead aim to ensure certain properties of the embedding space are satisfied. Generally, loss functions used in deep metric learning are based on the principle of contrasting similar and dissimilar pairs of embeddings. With respect to a loss function, we define a positive pair as a pair of samples of the same class, and a negative pair as a pair of samples of different classes. Another common notion is that of an anchor, which is a sample used as the point from which the distance to the positive and negative samples is computed. Mohan et al. (2023) provide a comprehensive overview of three of the most common loss functions used in deep metric learning: the contrastive loss, the triplet loss, and the n-pair loss.

Contrastive Loss A foundational loss function applied to the representation of pairs is the contrastive loss, which aims to manipulate the representation of samples, such that samples of the same class have increased similarity (or decreased distance), whilst samples of different classes are further apart (or increased distance) (Hadsell et al., 2006).

$$L(y, x_1, x_2) = \frac{1}{2}(1-y)\left(\mathbf{f}_1 - \mathbf{f}_2\right)^2 + \frac{1}{2}y\left(\mathbf{f}_1 - \mathbf{f}_2\right)^2 \tag{2.29}$$

Where y is the label of the pair, x_1 and x_2 are the embeddings of the two samples, and $\mathbf{f}_1$ and $\mathbf{f}_2$ are the features of the two samples.

Triplet Loss Another standard loss function for deep metric learning is the triplet loss, which extends the contrastive loss to include a third sample, the anchor, which is used as the point from which the distance to the positive and negative samples is computed.

$$L(y, x_a, x_p, x_n) = \max(0, d(x_a, x_p) - d(x_a, x_n) + \alpha) \tag{2.30}$$

Where $d(x_a, x_p)$ is the distance between the anchor and the positive sample, $d(x_a, x_n)$ is the distance between the anchor and the negative sample, and α is the margin.

This can be thought of as an extension of the contrastive loss, where the anchor-positive similarity is maximised whilst the anchor-negative similarity is minimised by a margin α.

N-pair Loss One limitation that may be immediately evident in the contrastive and triplet loss formulations is that they only consider a single positive and negative pair for each sample. This doesn't necessarily represent the ultimate goal of deep metric learning, which is to separate all samples into their respective classes. Additionally, triplet and

contrastive losses can be susceptible to slow convergence (Mohan et al., 2023). The n-pair loss extends this to include multiple positive and negative pairs for each sample.

$$L = \frac{1}{B} \sum_{i=1}^{B} \log \left(1 + \sum_{\substack{j=1 \\ j \neq i}}^{B} \exp \left(d(x_a^i, x_p^j) - d(x_a^i, x_p^i) \right) \right) \tag{2.31}$$

Where B is the batch size, x_a^i is the i-th anchor, x_p^i is the positive sample corresponding to the i-th anchor.

2.4.9 Foundation Models

Given the significant advances in empirical techniques, hardware and data availability, the training of large-scale deep learning models has become more feasible. This has led to the rise of **foundation models**, which are large-scale deep learning models trained on large datasets which are often used as a starting point for downstream tasks. Models may be fine-tuned with training data specific to the downstream task or used without further training for "in-context learning" such as prompting, as described by Schneider (2022).

Typically, foundation models are trained in a self-supervised manner, such as by using contrastive learning, to both reduce the amount of training data required and to improve the generalisation of the model (Caron et al., 2021; Vorontsov et al., 2024; Chen et al., 2022b).

2.5 Segmentation

Segmentation is the process of partitioning an image into regions. It can be broadly categorise into three main types: semantic, instance, and panoptic.

- **Semantic Segmentation**: Each pixel in an image is assigned a class label, grouping together all pixels belonging to the same class without distinguishing between different instances of the same class.

- **Instance Segmentation**: Each pixel is assigned a class label as well as an instance label, which differentiates between different instances of the same class.

- **Panoptic Segmentation**: Combines both semantic and instance segmentation by assigning a unique label to every pixel, indicating both the class and the specific instance (if applicable).

(**a**) Image (**b**) Semantic segmentation

(**c**) Instance segmentation (**d**) Panoptic segmentation

Figure 2.11: Segmentation types (Buhl, 2024)

2.5.1 Traditional Segmentation Methods

Prior to the advent of deep learning, there existed a range of traditional segmentation methods. These

Thresholding

Thresholding is a simple method for segmentation, using a grayscale image and a threshold value to binarise the image.

$$I_{i,j} = \begin{cases} 1 & \text{if } I_{i,j} > T \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

where $I_{i,j}$ is the intensity of the pixel at position (i, j), and T is the threshold value.

Morphological Operations

Morphological operations are operations applied to binary images. They are used to remove noise and small objects from the image, and to enhance the boundaries of objects. A morphological operation requires a structuring element, or kernel, which outlines the neighbourhood over which the operation is applied. There are two fundamental morphological operations: dilation and erosion. Dilation adds pixels to the boundaries of the object, while erosion removes pixels from the boundaries of the object.

2.5.2 Deep Learning Based Segmentation Methods

With the advent of deep learing, traditional segmentation methods were largely replaced by deep learning based methods.

Fully Convolutional Networks

Much of the deep-learning semantic segmentation methods of today can be attributed to the development of the Fully Convolutional Network (FCN) framework (Long et al., 2015). Rather than using fully connected layers as is appropriate for image classification, FCNs make use of convolution layers to directly produce feature maps for each pixel (or pixel group) in the image. Combined with upsampling through reverse convolutions, FCNs are able to produce a segmentation mask for the entire image.

The FCN framework has been widely adopted in the encoders of encoder-decoder architectures. (Csurka et al., 2023; Buhl, 2024)

Encoder-Decoder Architectures

Drawing from the NLP domain, the encoder-decoder architecture is a popular architecture for semantic segmentation. As its name suggests, it is a two part architecture consisting of an encoder and a decoder. The encoder is responsible for transforming the

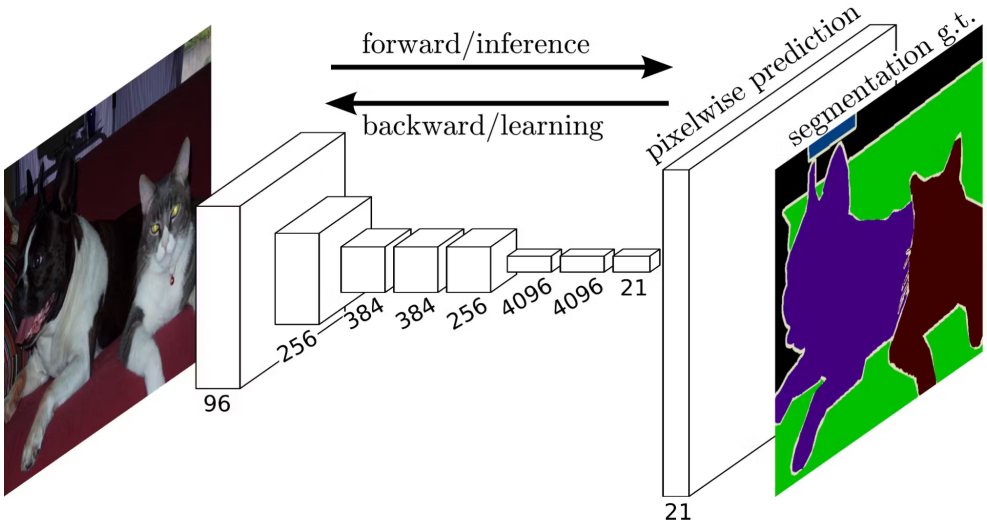

Figure 2.12: Fully Convolutional Network (Buhl, 2024)

input image into latent features, and the decoder is response for transforming the latent features into the final segmentation mask. Perhaps the most notable influential encoder-decoder architecture and of particular relevance to this work is the U-Net (Ronneberger et al., 2015), The key-innovation of the U-Net is the introduction of skip connections which allow output from the encoder layers to be concatentated with the input to the decoder layers. This allows the decoder to have access to higher resolution feature maps, rather than the lower resolution latent feature maps. Figure 2.13 illustrates the U-Net architecture.

The U-Net has become a ubiquitous architecture for semantic segmentation, particularly in the field of biomedical image segmentation (Berman et al., 2021; Zeng et al., 2025a)

2.5.3 Evaluation Metrics

Given the unique nature of segmentation tasks, there are a range of metrics which can evaluate different aspects of the performance of a segmentation model.

Intersection over Union (IoU)

The Intersection over Union (IoU) is a common metric for evaluating the performance of semantic segmentation models. It is calculated as the intersection of the predicted and ground truth masks divided by the union of the predicted and ground truth masks.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{2.33}$$

The mean IoU (mIoU) is simply an average of the IoU across all classes. The frequency

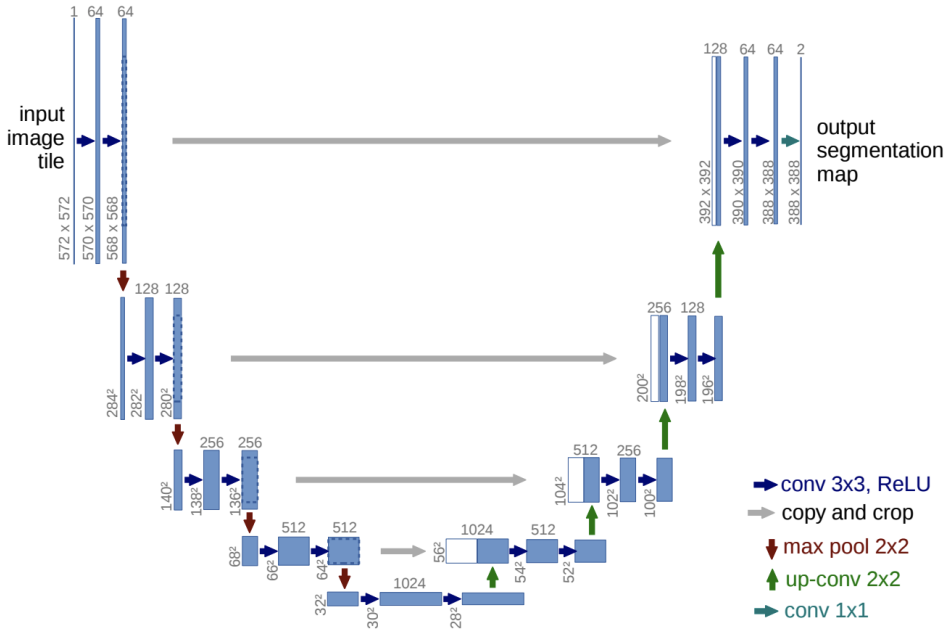

Figure 2.13: The U-Net architecture. (Ronneberger et al., 2015)

weighted IoU (fwIoU) is a weighted average of the IoU across all classes, weighted by the frequency of the class in the dataset (i.e the number of pixels).

Dice Score

The Dice Score is a common metric for evaluating the performance of semantic segmentation models. It is calculated as twice the intersection of the predicted and ground truth masks divided by the sum of the predicted and ground truth masks.

$$\text{Dice} = \frac{2 \times \text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{2.34}$$

Mean Average Precision (mAP)

The Mean Average Precision (mAP) is a common metric for evaluating the performance of semantic segmentation models. It is calculated as the average of the precisions at different recall levels.

Pixel Accuracy

The Pixel Accuracy is a common metric for evaluating the performance of semantic segmentation models. It is calculated as the number of correctly classified pixels divided by the total pixels.

Chapter 3

Related Work

3.1 Contrastive and Self-Supervised Learning

Focused on the underlying representation of data in a latent space, contrastive and self-supervised learning approaches have been shown to be effective in learning representations of data without additional annotation, a significant advantage for the histopathology domain and other low-data settings. Many successful methods for weakly-supervised semantic segmentation, both in and out of the histopathology domain, make use of self-supervised learning and contrastive learning (Zeng et al., 2025a; Tang et al., 2025).

3.1.1 Supervised Contrastive Learning

In the supervised domain, much of the literature has focused on the choice of contrastive loss function. Expanding basic contrastive losses which only consider pairs or triplets of embeddings to the supervised domain has seen success. The supervised contrastive loss (Khosla et al., 2021) considers multiple positives and negatives, rather than only a single positive and multiple negatives as seen in basic contrastive losses. This allows it to better leverage available label information, as evidenced in its increased performance on the ImageNet dataset (Khosla et al., 2021). An additional benefit is that its structure naturally performs hard negative mining, which avoids hyperparameter tuning that may otherwise be required.

3.1.2 Self-Supervised Contrastive Learning

Self-Supervised Learning has emerged as a powerful approach for learning meaningful representations of data without or with minimal annotation. The core idea across the literature is to generate similar representations for augmented views of the same image, and different representations for different images. The augmentations inherent to the

contrastive learning framework have the additional benefit of making models robust to inter-scanner variability and stain variability. (Komura et al., 2025)

Such an approach is formulated by SimCLR (Chen et al., 2020), MoCo (He et al., 2015), and DINO (Caron et al., 2021) in the creation of positive pairs by means of image augmentation and negative pairs as other images in a batch. A contrastive loss function is then applied to the representations, aiming to maximise the similarity between the embeddings of the same image, and minimise the similarity between the embeddings of different images. The contrastive loss in question is often the NT-Xent loss function (Chen et al., 2020):

$$l_i = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq i]} \exp(s_{i,k}/\tau)} \tag{3.1}$$

Where $s_{i,j}$ is the similarity between the embeddings of the two views of the same image, and τ is a temperature parameter.

Whilst the general principle of the above approaches is similar, there exist a number of differences. MoCo (He et al., 2015) introduces a momentum-based encoder updating strategy and a memory bank to generate unsupervised representations of data. DINO (Caron et al., 2021) introduces a linear projection head to the encoder, and a decoder to generate the representations.

As opposed to global representations, more granular approaches have demonstrated success, particularly in the medical domain. Given the structural properties of medical images, superpixel based approaches have demonstrated particular success. SuperCL (Zeng et al., 2025b) builds from the SimCLR framework by introducing a superpixel-guided contrastive approach prior to supervised training demonstrating state-of-the-art results across 8 medical image datasets. Multi-level Asymmetric Contrastive Learning (Zeng et al., 2025c) uses a similar approach but performs feature-level contrastive learning across features of differing scales in addition to instance-level contrastive learning.

It is worth noting that most of the aforementioned contrastive learning approaches are used as a pre-training step for later downstream tasks, particularly in the medical domain (Zeng et al., 2025b), rather than as part of, or to enhance the segmentation task itself. This is in contrast to approaches such as Local Contrastive Loss with Pseudo-Label based Self-Training (Chaitanya et al., 2021) which uses pseudo-labels produced by a minimally trained segmentation model on a limited dataset in combination with a contrastive loss to improve the performance of the segmentation model. What differentiates those approaches to this work is the separation of the supervised and self-supervised training stages. Here, we join the two stages in this work to form a more complete and robust approach.

When working with pseudo-labels filtering and / or thresholding of features prior to the application of a contrastive loss is common in task-specific approaches. Chaitanya et al. (2021) threshold the Dice Similarity Coefficient (DSC) of pseudo-labels, PBIP (Tang

et al., 2025) uses adaptive thresholding during the pseudo-label generation process, and MaskContrast (Gansbeke et al., 2021) uses a confidence threshold to filter foreground from background features.

Regardless of the approach, contrastive based self-supervised learning provides a strong foundation for both pre-training and weakly-supervised training. Whilst most of the work outlined relate to the natural and medical image domain, there have also been successful implementations of common contrastive-learning approaches in the histopathology domain; CTransPath (Wang et al., 2022a) highlights this with improved results across multiple downstream tasks by means of contrastive learning.

3.2 Segmentation Approaches

3.2.1 Network Architectures

As outlined in the background, most deep-learning semantic segmentation methods employ an encoder-decoder architecture. The choice of both the encoder and decoder are of significant importance to the performance of the model. With the success of FCNs, Resnets were the dominant backbone model for segmentation tasks for many years (Chen et al., 2018) (Ronneberger et al., 2015). Recently, transformers-based backbones have been shown to outperform their CNN counterparts, with the Swin Transformer (Liu et al., 2021) being particularly successful and now favoured in many segmentation tasks. We choose to make use of a Swin Transformer backbone for our encoder.

To better counter the spatial information loss inherent with the reducing feature resolution of the encoder, atrous decoders such as the DeepLab family of models, have been shown to be effective, particularly in the context of semantic segmentation (Chen et al., 2018). Decoders such as DeepLabv3+ (Chen et al., 2018) or that of Mask2Former (Cheng et al., 2021) have been shown to outperform standard upsampling decoders in the context of semantic segmentation. We choose to make use of the Mask2Former pixel-level decoder in this work.

3.2.2 Label-Efficient Methods

As previously discussed, the cost of obtaining pixel-level annotations for segmentation is high, and the need for label-efficient methods is clear. In order to address this challenge, various forms of less precise supervision have been used.

Generally, across all computer-vision domains, image-level labels are the most accessible and easiest to obtain, but lack the spatial information required for fully supervised semantic segmentation (Li et al., 2023; Tang et al., 2025). This provides the motivation for the use of bounding boxes, scribbles, and point annotations, which are more informative than image-level labels, but still less expensive to obtain than pixel-level annotations. Bounding Boxes, highlight regions of interest, often in the form of rectangular image coordinates, are simpler to create than pixel-level annotations, and are more informa-

tive than image-level labels as they contain meaningful spatial information (Gansbeke et al., 2021). Although, these are less suitable for the histopathology domain, as they do not provide well-defined borders of tissue regions, such as in the case of tumours, which are often required. Other approaches make use of scribbles, (Lin et al., 2016) and point annotations, (Bearman et al., 2016), which are more informative than image-level labels, but still less expensive to obtain than pixel-level annotations. The broad use of alternative forms of annotation underscore the limiting and often prohibitive cost of obtaining pixel-level annotations and the need for label-efficient methods.

3.2.3 Foundation Models

The rise of foundation models has also touched upon the segmentation task, with many models demonstrating strong few and zero-shot performance whilst also being able to be fine-tuned to the segmentation task.

General purpose foundation models such as CLIP (Radford et al., 2021), DINO (Caron et al., 2021) and SAM (Kirillov et al., 2023) have been shown to have strong performance in few and zero-shot tasks (Zhou et al., 2024). Notably, all of these models are transformer-based and the emergence of segmentation ability often comes through modifications to the attention mechanism. The emergent segmentation properties are strengthened by additional contextual information provided through promptable models; models such as SAM receive additional context such as weak labels, or example segmentations which with aid of a prompt encoder lead to more refined masks.

3.2.4 Weakly-Supervised Semantic Segmentation Approaches

The dominant approach for weakly-supervised semantic segmentation is the use of Class Activation Maps (CAMs) to generate pseudo-labels for the segmentation task. Introduced by Zhou et al. (2015), CAMs apply the linear classifier weights to the last feature map of the network, prior to GAP, to reveal the importance of each pixel in the feature map for the classification task. Importantly, Zhou et al. (2015) demonstrate that a network is capable of localizing discriminative image regions for a variety of tasks, despite only being trained on the global classification task.

There have been various attempts at improvements to the CAM method, such as Grad-CAM (Selvaraju et al., 2019), which removes the need for pooling and the linear classifier, and instead uses the gradient of the logits with respect to the features to generate the CAMs. Other attempts follow the same general approach, but use different methods to generate the CAMs such as HiResCAM (Draelos and Carin, 2021) or EigenCAM (Muhammad and Yeasin, 2020). The downside of these methods is that they are computationally expensive

A common issue with most CAM approaches is partial activation, where only the most discriminative regions of the image are activated. This is because the classification objective doesn't necessarily require recognition of the entire object for a given class

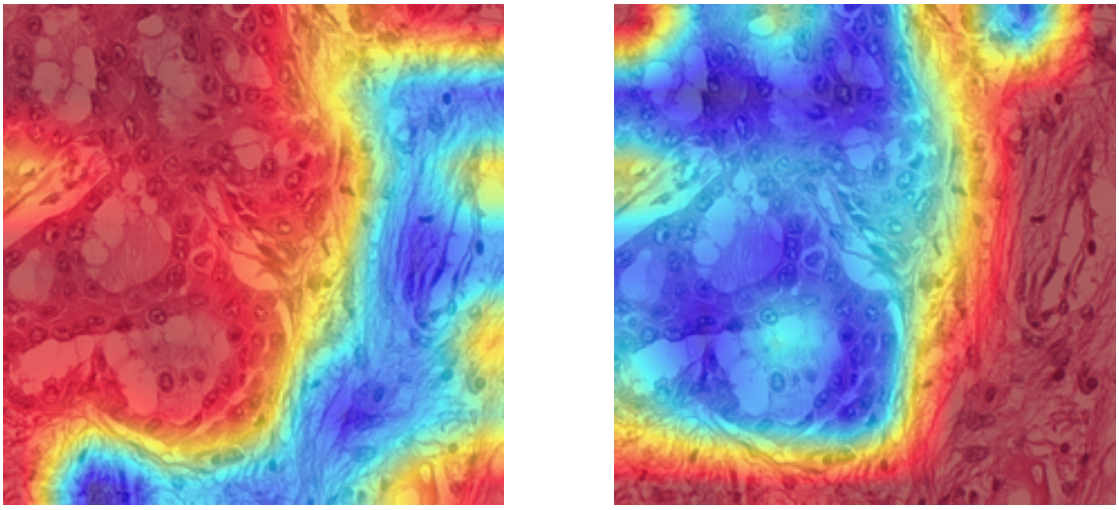

Figure 3.1: Example CAMs for Tumor-associated Stroma (left) and Tumor Epithelial regions (right).

(Kang et al., 2025). This issue provides the motivation for the use of additional methods, to either enforce the activation of the entire object by the classification objective, or to attempt to broaden the activation of the object by the discriminative regions. In the subsections below, we outline various approaches to address the issue of partial activation and / or pseudo-label refinement.

Prototype-based Approaches

Prototype-based approaches make use of a set of class-specific prototypes to generate the pseudo-labels for a segmentation task. Simple prototype networks learn a non-linear transformation of the input data from which the mean across a given class can be taken to create a prototype Liu et al. (2024b). The prototypes can then be applied to the feature maps of the network via a similarity metric, such as cosine similarity, to generate the pseudo-labels for a segmentation task.

More complex prototype-based approaches, such as Self-Supervised Image-Specific Prototype Exploration (SIPE) (Chen et al., 2022a) directly learn the prototypes in a self-supervised manner, by training both the classification and segmentation tasks. SIPE uses the CAMs generated from the classification task to select features, generated in addition to the classification features, which best align with the class-specific regions of the image to generate the prototypes. The segmentation masks generated from the prototypes are then applied in a self-supervised manner, minimising the absolute difference between the generated masks and the CAMs. This approach has been shown to help address the issue of partial activation outlined above.

One benefit of the use of prototypes is that the similarity metric used to generate the pseudo-labels can be tailored to the specific task at hand. This can include rewarding

foreground-background dissimilarity such that the less-disciminative regions of CAMs are artificially activated to help address the issue of partial activation (Kang et al., 2025), (Ahmadi and Kasaei, 2024) (Chen et al., 2022a). The creation of unbiased activation maps (UAMs) (Kang et al., 2025) is a specific example of this, where the creation of the pseudo-labels is guided by the foreground-background dissimilarity.

Attention-based Approaches

With the advent of the ViT, attention-based approaches have grown in popularity for addressing the issue of partial activation. The unchanging size of the attention maps across transformer blocks allows for their aggregation into a single feature map which considers features across various scales. This is a particular advantage when compared to CNN-based approaches which generally only consider the highest level features located in the last layer of the network. This has been demonstrated to increase the quality of the CAMs and thereby the performance of the segmentation task. One such model which makes use of this approach is TransCAM (Li et al., 2023) which combines traditional CNN-based CAMs with the average of the attention maps to generate improved CAMs in a Conformer network, producing state-of-the-art results on segmentation tasks. The SWT-Former (Ahmadi and Kasaei, 2024) combines SIPE (Chen et al., 2022a) with the Swin Transformer (Liu et al., 2021) to generate segmentation masks that compete with or better state-of-the-art methods on the PASCAL VOC 2012 dataset.

Post-Processing Methods

In order to approve the quality of segmentation masks, the literature highlights various post-processing methods are often used. Perhaps the most popular, the use of Dense Conditional-Radiance Fields (DenseCRF) (Krähenbühl and Koltun, 2012) has been shown to be effective in improving the quality of segmentation masks by incorporating the spatial information of the image into the masks. Affinity-based methods, such as AffinityNet (Ahn and Kwak, 2018), make use of the affinity between pixels to help refine the pseudo-labels for a segmentation task.

Inherently, these methods rely on the quality of the initial pseudo-labels, and as such, prioritising the quality of the initial pseudo-labels is key. Furthermore, Kang et al. (2025) demonstrate that such methods which have demonstrated strong performance on the natural image domain can degrade the performance of pseudo-labels when applied to the histopathology domain. As such, we focus on producing pseudo-labels which are as high quality as possible, rather than using post-processing methods in this work.

3.3 Weakly Supervised Semantic Segmentation Approaches for Histopathology

Given the specialised nature of the histopathology domain, there are various approaches specific to the domain. Generally, the core methodology is to train a backbone model

on the classification task using a Multi-Label Soft Margin Loss (MLSM) and then use the feature maps from the backbone to generate the pseudo-labels for the segmentation task (Han et al., 2021), (Tang et al., 2025), (Chan et al., 2019).

The point of difference between approaches lies in the various methods used to generate the pseudo-labels for the segmentation task. Earlier approaches such as HistoSegNet (Chan et al., 2019) primarily used CAM-based methods with standard segmentation post-processing. These methods were susceptible to noise and over-activation of the most discriminative regions and performance was thus limited. To address these issues, more complex methods often integrate the feature maps from multiple layers of the backbone to create more powerful hierarchical features. Multi-Layer Pseudo-Supervision (MLPS) (Han et al., 2021) and PBIP (Tang et al., 2025) are examples of this within the Histopathology domain.

Prototype-based approaches to generating pseudo-labels have also seen success in histopathology tasks. By generating prototypes corresponding to tissue types, these methods use the similarity between the prototypes and the feature maps to generate the pseudo-labels for the segmentation task (Kang et al., 2025). It is worth nothing that these approaches often use an additional intermediary step, such as K-Means in the case of Kang et al. (2025) to refine the feature maps. This differs from the approach outlined in this work which treats CAM refinement as a part of the training process.

The rise of multi-modal approaches across machine learning, thanks to models such as CLIP (Radford et al., 2021), with increased performance has invited application to the Histopathology domain. Through the injection of text and label annotations using MedCLIP (Wang et al., 2022b) and ClinicalBERT (Huang et al., 2020) respectively and an attention mechanism, TPRO demonstates previously state-of-the-art performance on the LUAD-HistoSeg and BCSS-WSSS datasets (Zhang et al., 2023). Their approach differs from that which is outlined in this work through the introduction of two additional encoders and an additional text-prompting mechanism.

The strongest performing approaches such as Kang et al. (2025) and Han et al. (2021) often then introduce the additional step of self-supervised learning by using the pseudo-labels as the target masks for a fully supervised training approach. Most approaches use the DeepLabv3+ decoder (Chen et al., 2018) as the decoder. Whilst a standard CrossEntropy or Dice loss are often used, certain approaches such as (Kang et al., 2025) attempt to consider the uncertainty of the pseudo-labels to improve the performance of the model by introducing the confidence of the CAMs into the loss calculation. We explore the use of this Noise Reduced Loss in this work.

Kang et al. (2025), Zhang et al. (2023) and Han et al. (2021) all evaluate the performance of their approaches on the LUAD-HistoSeg and BCSS-WSSS datasets, providing motivation for the use of these datasets in this work. Additionally, Kang et al. (2025) compare the performance of their approach with various other methods, including results of both CAMs, and the additional fully supervised training approach stage to which we make reference. For convenience, these results can be found in Appendix A

3.4 Foundation Models for Histopathology

The strong performance and broad applicability of foundation models has seen them used across wide-ranging applications. With various, performance proven models such as CLIP (Radford et al., 2021), DINO (Caron et al., 2021), DINOv2 (Oquab et al., 2023), and SAM (Gurcan et al., 2009), now available, there is a growing interest across the literature in their applicability to the Histopathology domain. The application of such models highlight intrinsic difficulties of the histopathology domain, reveal techniques for extracting strong empirical performance from foundation models, whilst underscoring the need for specialist models and approaches remains.

In recognition of the strength of CLIP and other text-image alignment models as well as the intrinsic link between reports and images that exist in the histopathology domain, CLIP-inspired models such as MedCLIP (Wang et al., 2022b) have garnered significant interest in the Histopathology domain. As already outlined, there exist three main problems that make these models somewhat unsuitable for the Histopathology domain. Firstly, the cost of obtaining an appropriate amount of data is high, given the necessity of obtaining both image and text annotations. Secondly, the histopathology domain lacks much of the publicly available text annotations that are available for other medical images such as radiology. Thirdly, histopathological images are often of a significantly increased size when compared to other medical images such as radiology, which can make the use of these models more difficult (Gurcan et al., 2009).

In recognition of these challenges, image-only models such as Virchow (Vorontsov et al., 2024) and HIPT (Chen et al., 2022b) have been developed. Whilst these models are not specifically designed for the segmentation task, they provide a foundation for the development of specialist models for the histopathology domain. Such models are large (632 million and 10,000,000 parameters respectively) and require significant amounts of data to train. Virchow is trained on 1.5 million hematoxylin and eosin stained WSIs whilst HIPT is trained on 10,000 WSIs in addition to 408,218 4096x4096 patches and 104 million 256x256 patches. Following the trend in foundation models, both models are transformer-based and are trained using the DINO algorithm (Caron et al., 2021) highlighting the affinity between transformers and self-supervised learning. Of interest are their approaches to the large image sizes inherent to the histopathology domain. Unsurprisingly, both models make use of patch-based training, although HIPT extends this approach by training on patches of varying sizes to create a hierarchical representation of the image. Unfortunately, neither model has been applied to segmentation tasks, although HIPT provides some segmentation ability through visualisation of the self-attention maps of the model which can assist in interpreting classification results.

Whilst the above approaches adapt foundation models to the segmentation task, SAM provides a foundation for zero-shot histopathology segmentation with few modifications such as PathoSAM (Griebel et al., 2025), and WSI-SAM (Liu et al., 2024a). Such models demonstrate strong performance on their respective segmentation tasks in a zero-shot context, although they still fall behind specialist state-of-the-art models. To attempt to

overcome this, the production of industry specific foundation models for segmentation such as MedSAM (Ma et al., 2024) have been show to consistently outperform state-of the-art segmentation foundation models whilst achieving similar performance to specialist models. Despite the strong performance of these models, they generally fall behind specialist state-of-the-art models, particularly in the histopathology domain as demonstrated by Kang et al. (2025). Therefore, motivation for furthering the development and performance of specialist models, such as that which is explored in this work, remains.

Chapter 4

Weakly Supervised Semantic Segmentation for Histopathology.

This chapter contains the explored approaches of our work across varying levels of supervision for the segmentation task. We first start by foregrounding the experimental setup, including the model structure and various other elements. We then outline the training approaches we explore across the different levels of supervision. The results of the experiments are presented in Chapter 5.

4.1 Model Architecture

In this section, we outline the building blocks of the model architecture used in our experiments.

4.1.1 Encoder Backbone

The primary backbone for the model is the Swin Transformer Liu et al. (2021) with a window size of 7×7 and a patch size of 4×4. We make use of the Imagenet-22k pre-trained model for which the weights can be found here. Feature maps can be extracted from various layers of the encoder. In this work, we extract feature maps from the first, second, third, and fourth blocks of the Swin Transformer of resolutions $(56 \times 56, 28 \times 28, 14 \times 14, 7 \times 7)$.

To produce classification logits, for the weak supervision tasks, we append a final projection head to the decoder, in the form of a 1×1 convolutional layer to produce the class logits for each feature. This is further outlined in Section 4.3.

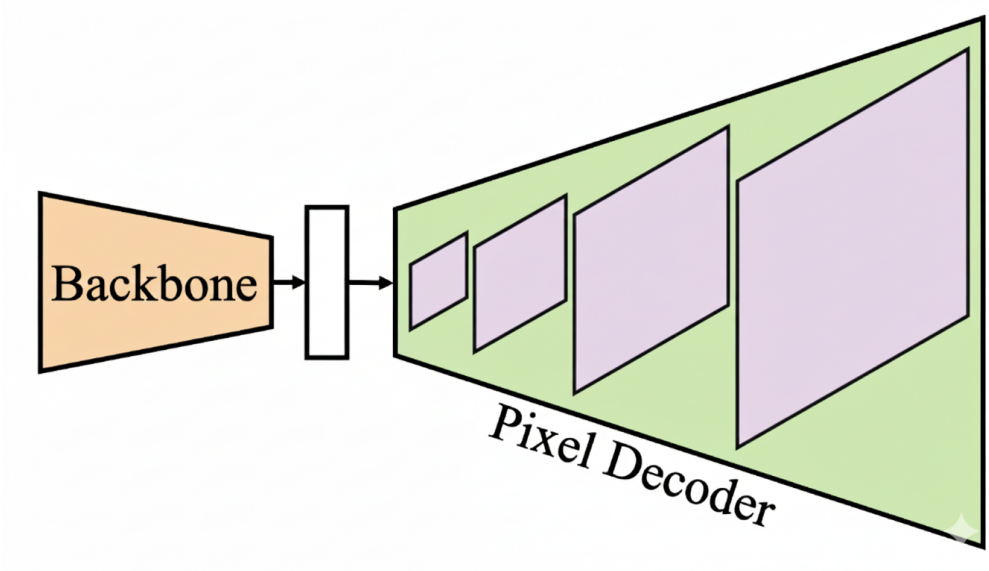

Figure 4.1: Overview of the Mask2Former pixel-level module used as the model structure in our experiments.

4.1.2 Encoder-Decoder

Inspired by Cheng et al. (2022), we use the pixel-level module of the Mask2Former model, which consists of the encoder backbone and a Feature Pyramid Network (FPN) (Lin et al., 2017) based decoder. For simplicity, we make use of the `transformers` python library (Wolf et al., 2020) to load the Mask2Former model. As the pixel-level module of the Mask2Former produces pixel embeddings, we append a final projection head to the decoder, in the form of a 1×1 convolutional layer to produce the class logits for each feature. We use an embedding dimensions of 256.

4.1.3 Augmentations

We perform standard data augmentations to the images throughout the training process. This includes random horizontal and vertical flips and normalisation. Additional transformations performed as part of the unsupervised pre-training process are outlined in Appendix A.3.

4.2 Unsupervised Training

As is common in the literature, we start by perform an unsupervised pre-training on a large dataset of unannotated images. The goal is this stage is to learn, through a pre-text task, a rich feature space that can later be fine-tuned for the segmentation task downstream. As shown in Fig. 4.2, we make use of data augmentations to generate a diverse set of views of the same image.

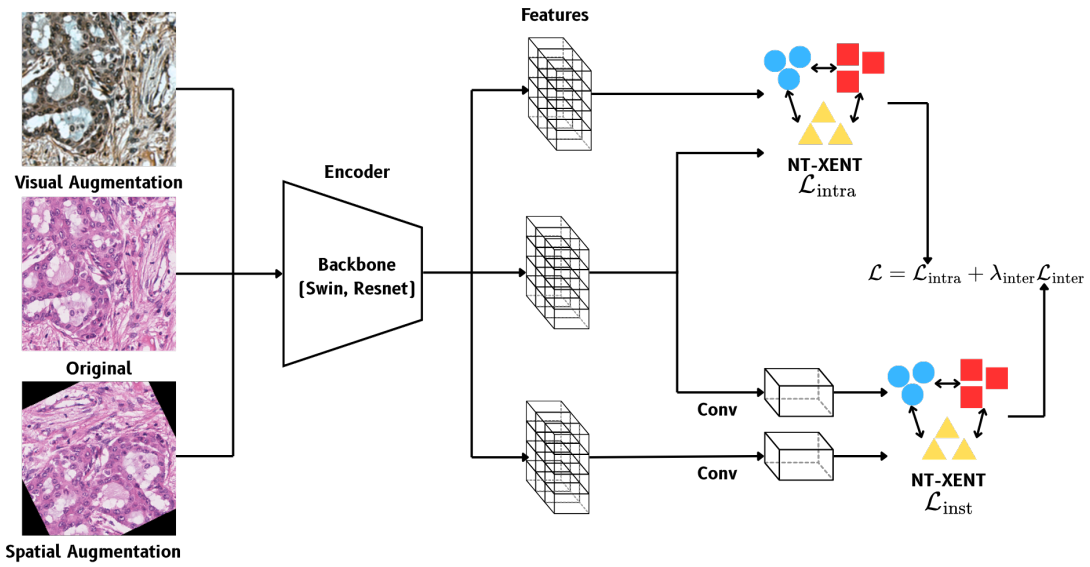

Figure 4.2: Overview of the unsupervised training approach for the downstream segmentation task.

Following Zeng et al. (2025a), we use a contrastive loss based approach to pre-train our model. As is common in unsupervised pre-training, we make use of data augmentations to generate a diverse set of views of the same image. In particular, for each image, we generate two additional views. The first view is a visual transformation of the original image, with no spatial transformations applied. The second view is generated by applying a random spatial transformation to the original image.

Use the augmented views, we can define a contrastive loss with the goal of pushing the embeddings of the augmented views of the same image closer together, while pushing the embeddings of the augmented views of different images apart. This can be broken down into two components: An inter-image contrastive loss and an intra-image contrastive loss. For both the inter-image and intra-image contrastive losses we make use of the NT-Xent loss function 3.1.2. (Zeng et al., 2025a).

Combining the inter-image and intra-image contrastive losses, we can define the total contrastive loss as:

$$\mathcal{L}_{total} = \mathcal{L}_{intra} + \lambda_{\text{inter}}\mathcal{L}_{inter} \tag{4.1}$$

Where λ_{inter} is a weighting factor for the inter-image contrastive loss. We set λ_{inter} to 0.5 as recommended by Zeng et al. (2025a).

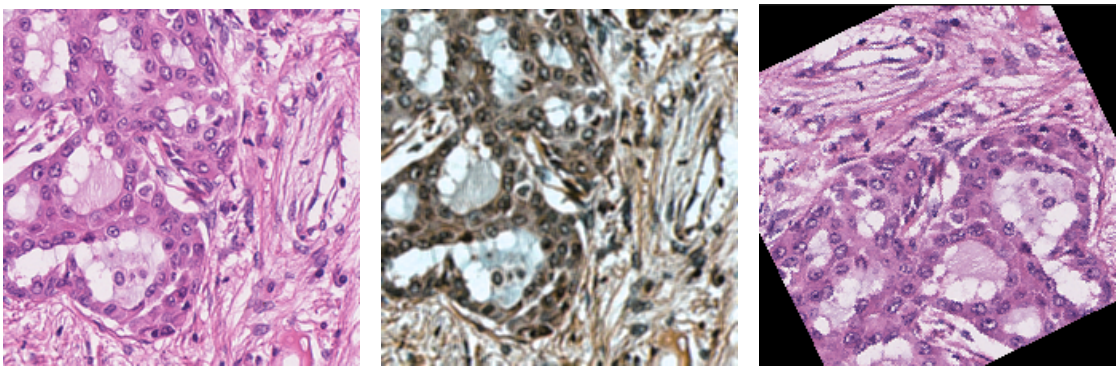

Figure 4.3: Examples of augmentations used in unsupervised contrastive pre-training: (left) original image, (middle) visual-only augmentation, and (right) spatial augmentation.

4.3 Weakly Supervised Training (Stage 1)

In this section, we outline the approach for training the model on the classification task in line with the literature. We outline the classification objective, the CAM generation process (including various modifications to the CAM method), as well as our Pseudo-Supervised Contrastive Loss (PSCL) approach. The process in its entirety can be seen in 4.4.

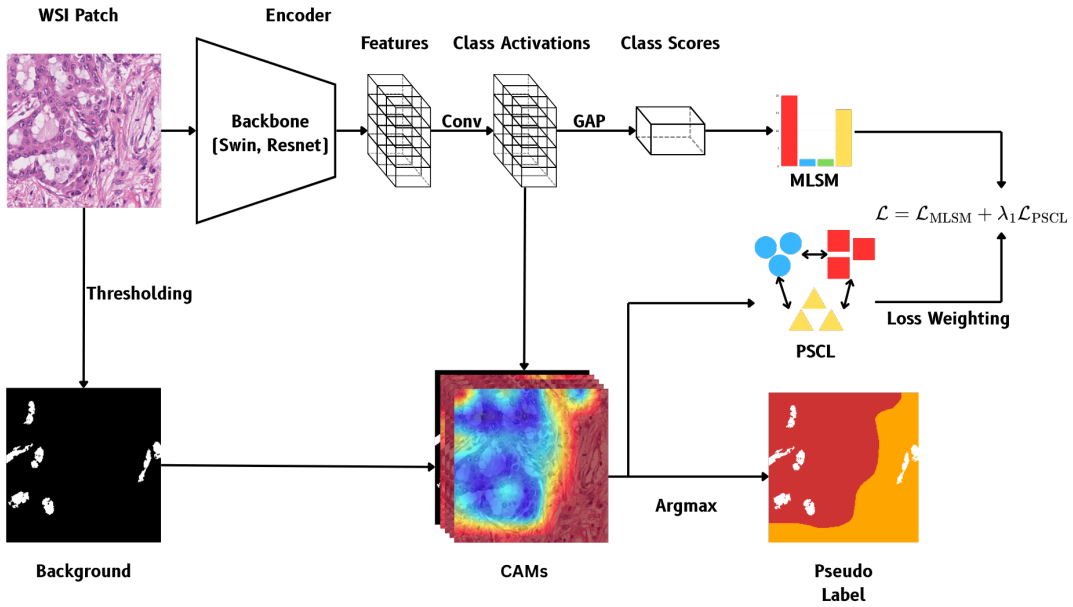

Figure 4.4: Overview of the weakly-supervised training approach for the segmentation task.

Our total loss function is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{mlsm} + \lambda_{\text{pscl}}\mathcal{L}_{pscl} \tag{4.2}$$

where λ_{pscl} is a weighting factor for the PSCL loss. We conduct ablation studies on the weighting of the PSCL loss and find that the performance of the model is maximised when the lambda is set to 0.1.

4.3.1 Classification

We start the weakly-supervised training by training the model on the patch-level labels. We make use of Swin Transformer Resnet38 encoder backbones. We append a $1x1$ convolutional layer to the output of the last layer of the encoder with n output channels where n is the number of classes. To train for the classification task, we make use of the Multi-Label Soft Margin Loss (MLSM) loss function, applying GAP to the output of the $1x1$ convolutional layer to produce a n dimensional vector.

$$L_{\text{mlsm}} = -\frac{1}{N}\sum_{i=1}^{N} y[i]\log(\frac{1}{1+e^{-x[i]}}) + (1-y[i])\log(\frac{\exp(-x[i])}{1+e^{-x[i]}})$$

where $y[i]$ is the one-hot encoded label for the ith class, and $x[i]$ is the class score for the ith class.

4.3.2 CAM Generation

Similar to Han et al. (2021), we use the CAM method to generate the useful feature maps for the segmentation task. For a given class i, we have that the CAM at a is defined as:

$$\begin{aligned} CAM_i(x) &= \frac{ReLU(A_i)}{\max \text{ReLU}(A_i)} \\ A_i &= w_i^T \mathbf{f} \end{aligned} \tag{4.3}$$

where $\mathbf{f}$ is the feature map for class i, and w_i is the weight vector from the $1x1$ convolutional layer.

Background Seeding

One issue with the CAM method is that identification of the background class is not an objective to which the model can be trained as it is not included in the labels. Various methods have been proposed to address this issue, but we focus on background seeding. Inspired by Kang et al. (2025), we perform colour thresholding and morphological operations on the image to create a binary map of background regions which is then concatenated onto the CAMs to produce $n+1$ CAMs.To refine this mask and remove

noise, we apply a morphological operation specifically, the `remove_small_objects` function from `skimage` which eliminates small, isolated regions that are unlikely to be true background. The resulting binary map highlights the main background areas, which is then concatenated with the class activation maps (CAMs) to provide an additional background channel for segmentation. We find that a `min_size` of 200 is most effective for the LUAD-HistoSeg dataset and 600 for the BCSS-WSSS dataset. We visualise the background seeding in Figure 4.5.

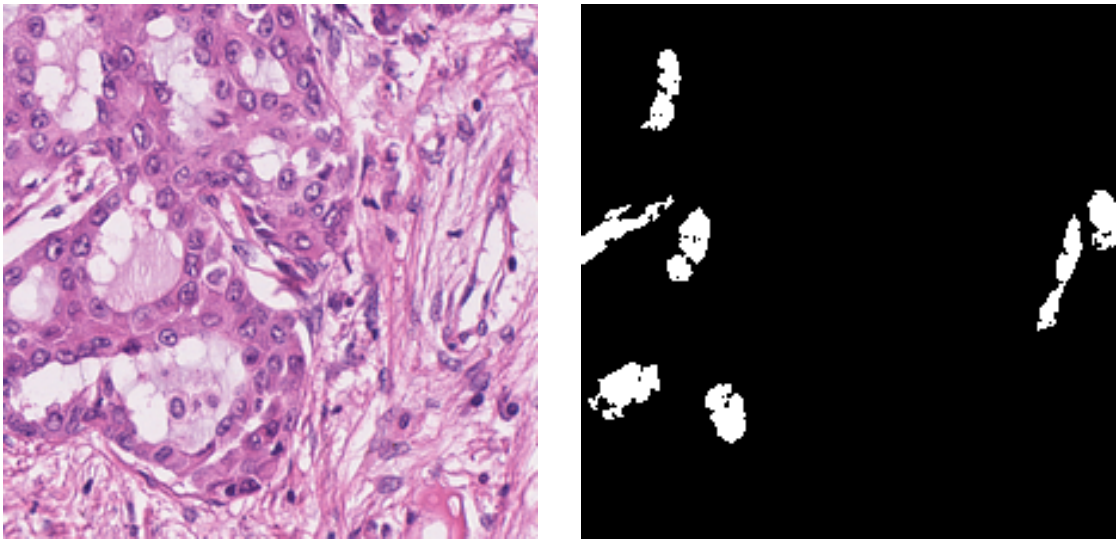

Figure 4.5: Background seeding example. Left: original image patch; Right: generated background seed highlighting background regions.

Gating Mechanism

In order to improve the quality of the generated pseudo-labels, we introduce a gating mechanism which takes the classification score and the CAMs to remove CAM responses for which the model is not confident in its classification. This is from the premise that the model's ability to classify the image is stronger than its ability to segment the image. Inspired by Han et al. (2021) we use the following gating function:

$$g_n(x) = \begin{cases} 1 & \text{if } \mathbf{c}_n(x) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

For each class n, we set τ to 0.8 for all experiments. We concatenate a constant background 1 into the classification scores to ensure that the background is always included in the segmentation task. Combining the gating function with the CAMs, we can generate the gated CAMs for each class n as:

$$CAM_n(x) = g_n(x) \cdot CAM_n(x) \tag{4.5}$$

Pseudo-Supervised Contrastive Loss (PSCL)

A significant challenge in weakly-supervised object localization is the phenomenon of **partial-activation**, where a model's Class Activation Maps (CAMs) fail to identify the entire extent of an object, instead focusing on the most discriminative regions. To address this, we introduce the Pseudo-Supervised Contrastive Loss (PSCL), a novel approach that encourages the model to learn a more semantically consistent and separable feature space. This loss operates at a pixel-level, using the model's own CAMs to generate a set of reliable pseudo-labels for feature-map supervision.

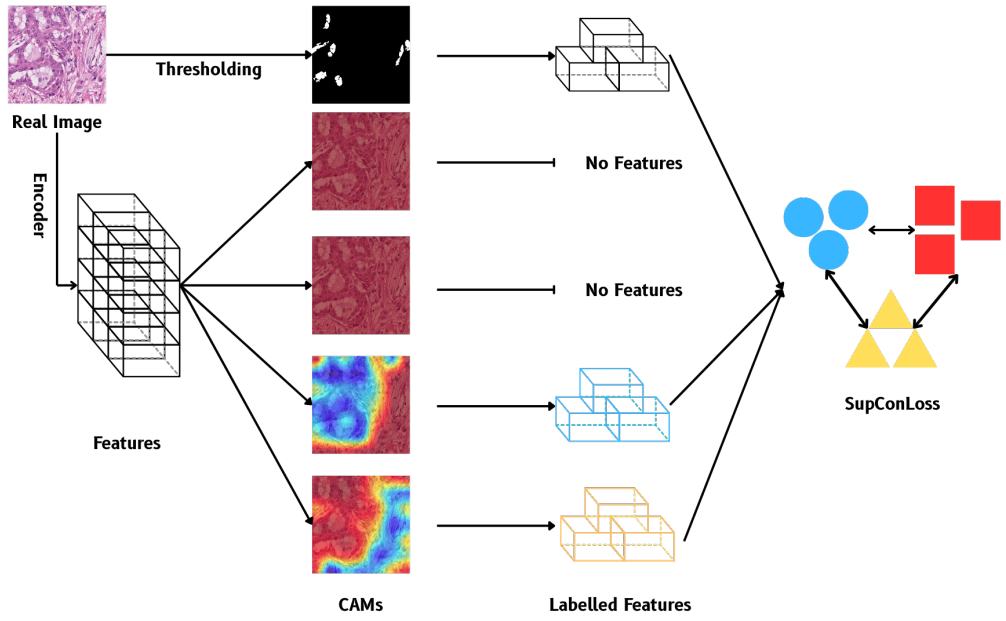

Figure 4.6: Overview of the pseudo-supervised contrastive loss approach. We make use of the CAMs as pseudo-labels to create a more separable feature space.

The core of our approach is the application of a supervised contrastive loss to the feature maps $\mathbf{f} \in \mathbb{R}^{C \times H' \times W'}$, where C, H', and W' are the number of channels, height, and width, respectively. The supervision signal is derived from the CAMs $\mathbf{c} \in \mathbb{R}^{N \times H \times W}$, where N is the number of classes. We define the pseudo-label for each pixel as the class with the maximum activation value, effectively generating a pixel-wise label map $\mathbf{l} = \text{argmax}(\mathbf{c})$.

However, relying solely on $\texttt{argmax}(\mathbf{c})$ can introduce noisy labels from low-confidence regions. To mitigate this, we filter out unreliable pixels by applying a confidence threshold τ. Pixels where the maximum CAM value is below this threshold are excluded from the

loss calculation. The pixel-wise pseudo-label is thus defined as:

$$\mathbf{l} = \begin{cases} \text{argmax}(\mathbf{c}_{ij}) & \text{if } \max(\mathbf{c}_{ij}) > \tau \\ -1 & \text{otherwise} \end{cases} \tag{4.6}$$

where $\mathbf{c}_{ij}$ is the CAM vector for pixel (i, j). In our implementation, we set τ to 0.25. The feature embeddings are extracted where the pixel-wise pseudo-label is not -1.

$$\hat{\mathcal{F}} = \{\, \mathbf{f}_{ij} \mid \mathbf{l}_{ij} \neq -1 \,\} \tag{4.7}$$

The extracted features along with their pseudo-labels are then fed into the standard supervised contrastive loss.

$$L_{supcon} = -\frac{1}{|\hat{\mathcal{F}}|} \sum_{i \in \hat{\mathcal{F}}} \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\frac{\text{sim}(z_i, z_p)}{\tau}\right)}{\sum\limits_{a \in \hat{\mathcal{F}},\, a \neq i} \exp\left(\frac{\text{sim}(z_i, z_a)}{\tau}\right)} \tag{4.8}$$

where $P(i)$ is the set of pixels with the same pseudo-label as i.

Pseudo-Label Generation

To improve the quality of the pseudo-labels produced for the second stage of training, we first perform the background seeding method outlined above, where a thresholding mechanism is used to identify background regions followed by a morphological operation (specifically, the `remove_small_objects` function from `skimage`) to eliminate small objects. Instead of concatenating the background mask to the CAMs, we generate the final pseudo-labels by taking the argmax of the CAMs (per pixel), add 1 to shift class indices, and then multiply by the complement of the background mask, as shown in the equation below. Additionally, we gate the CAMs by setting any non-present class to 0, using the patch one-hot labels from the training set of the respective datasets.

$$\hat{\mathbf{l}} = [\text{argmax}(\mathbf{CAMs}) + 1] \cdot (1 - \mathbf{bkg}) \tag{4.9}$$

4.4 Weakly Supervised Training (Stage 2)

As is common in the literature, we follow a two-stage training approach. In the second stage, we take the pseudo-labels produced by the first classification training stage and train the model, including the decoder, on the segmentation task as if it were fully supervised. This is outlined in Figure 4.7.

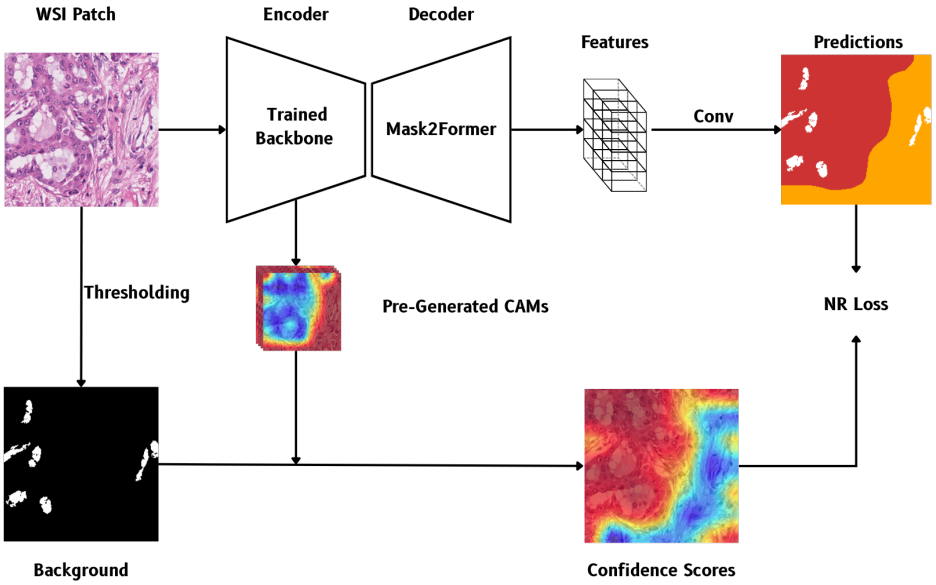

Figure 4.7: Overview of the weakly-supervised training approach for the segmentation task.

Noise Reduced Loss

Following Kang et al. (2025), we make use of the noise reduced loss (NR Loss) whilst training on the pseudo labels. This loss considers the uncertainty of the pseudo-labels when penalising the model for incorrect predictions to account for the potential for incorrect pseudo-labels.

We first generate a confidence score for a given pixel by dividing the CAM value by the sum of the CAM values for all classes.

$$c_n(x) = \frac{CAM_n(x)}{\sum_{i=1}^{n} CAM_i(x)} \tag{4.10}$$

We apply the confidence score to the standard BCE loss 2.17.

$$L_{nr} = -\sum_{i=0}^{n+1} c_i(x) \left(y_i \log(P_i(x)) + (1 - y_i) \log(1 - P_i(x)) \right) \tag{4.11}$$

where y_i is the one-hot encoded label for the class i, and $P_i(x)$ is the Gumbel Softmax probability for the class i.

4.4.1 Datasets

We follow the literature for weakly-supervised histopathology segmentation, and use the following datasets:

- **LUAD-HistoSeg** - A dataset of 17,285 224x224 image patches from WSIs at 10x reoslution from patients with lung adenocarcinoma. There are 16,678 patches with global labels in the training set and 300 and 307 patches with pixel-level labels in the validation and test sets respectively. There are four labels in the dataset: *Tumour Epithelial*, *Tumour Associated Stroma*, *Necrosis*, and *Lymphocytes* (Han et al., 2021)
- **BCSS - WSSS** - A dataset of 31,826 224x224 image patches from WSIs at 40x resolution from patients with breast cancer. It is made up of 23,422 patches with global labels in the training set, and 3,418 and 4,986 patches with pixel-level labels in the validation and test sets respectively. There are 5 labels in the dataset: *Tumour*, *Stroma*, *Lymphocytic Infiltrate*, *Necrosis*, and *Other* (Han et al., 2021)

Chapter 5

Evaluation

In this chapter, we evaluate the performance of the proposed methods. We start by outlining the implementation details before evaluating the performance of the models across the different experimental stages.

5.1 Implementation Details

5.1.1 Hardware Setup

All experiments are conducted on an Ubuntu 24.0.1 machine with an NVIDIA GeForce RTX 3090 GPU with 24GB of VRAM and 64GB of RAM.

5.1.2 Software Setup

All experiments are conducted using the PyTorch framework with a CUDA version of 12.4. Table 5.1 shows the hyperparameters used for the training of the models.

Table 5.1: Training hyperparameters used across different experimental stages.

Parameter	**Stage 1**	**Stage 2**	**Unsupervised**
Optimizer	SGD	AdamW	AdamW
Learning Rate	0.1	0.0001	0.0001
Epochs	5	20	20
Scheduler	Polynomial Decay	–	–
Batch Size	16	32	6
Loss Function	MLSM + PSCL	NR / CE	NT-Xent

We make use of the SupConLoss implementation from the pytorch-metric-learning library (Musgrave et al., 2020). Our distance metric is cosine similarity.

5.2 Weakly Supervised Training (Stage 1)

We demonstrate that our method is able to generate high quality pseudo-labels with only weak supervision on the classification task. Additionally, we demonstrate that our novel PSCL approach significantly increases performance across both datasets and helps to address the issue of partial-activation that CAMs are susceptible to. Performance of the model is competitive with state-of-the-art methods, achieving a rank of second on the BCSS-WSSS dataset.

We include our best results for Swin and ResNet50 backbone architectures in Table 5.2 and Table 5.3 for the LUAD-HistoSeg and BCSS-WSSS datasets respectively. We additionally examine various other components of the approach, including the impact of the PSCL, the background seeding mechanism, and the impact of the weighting of the PSCL in further detail.

Table 5.2: Overall and per-class IoU performance (%) of weakly supervised models on LUAD-HistoSeg dataset.

Model	**TE**	**NEC**	**LYM**	**TAS**	**mIoU**	**fwIoU**
Swin (Ours)	67.49	72.39	69.82	65.28	68.75	67.35
ResNet50 (Ours)	52.02	35.44	50.22	51.29	56.44	50.27

Table 5.3: Overall and per-class IoU performance (%) of weakly supervised models on BCSS-WSSS dataset.

Model	**TUM**	**STR**	**LYM**	**NEC**	**mIoU**	**fwIoU**
Swin (Ours)	78.66	67.05	49.11	63.96	64.70	70.13
ResNet50 (Ours)	69.94	57.64	48.26	50.11	56.49	61.68

5.2.1 Comparison to State of the Art

We find that our method performs below recent state-of-the-art methods on the LUAD-HistoSeg dataset, but is competitive on the BCSS-WSSS dataset. Table 5.4 and Table 5.5 show the top 3 state-of-the-art methods and our approach on the LUAD-HistoSeg and BCSS-WSSS datasets respectively. The full state-of-the-art results can be found in Appendix A. While our method is competitive on BCSS-WSSS, achieving second place, the lower score in the low-frequency LYM class suggests a limitation in handling class imbalance. Nevertheless, the strong performance across high-frequency and clinically relevant classes like TUM and STR validates PSCL's success in broadening CAM activation to capture the full extent of key tissue regions, directly addressing the core problem of partial activation

Table 5.4: Top 3 state-of-the-art methods and our approach on LUAD-HistoSeg dataset (mIOU %).

Method	TE	NEC	LYM	TAS	fwIoU	mIoU
UAM (Kang et al., 2025)	**76.24**	**80.43**	**76.28**	**72.02**	**75.38**	**76.24**
TPRO (Zhang et al., 2023)	74.82	77.55	76.40	70.98	73.81	74.94
MLPS (Han et al., 2021)	71.72	76.27	73.53	67.67	70.80	72.30
Swin (Ours)	67.49	72.39	69.82	65.28	67.35	68.75

Table 5.5: Top 3 state-of-the-art methods and our approach on BCSS-WSSS dataset (mIOU %).

Method	TUM	STR	LYM	NEC	fwIoU	mIoU
UAM (Kang et al., 2025)	**78.97**	**71.72**	**58.16**	**63.59**	**72.20**	**68.11**
TPRO (Zhang et al., 2023)	77.18	63.77	54.95	61.43	68.55	64.33
MLPS (Han et al., 2021)	70.76	61.07	50.87	52.94	63.89	58.91
Swin (Ours)	78.66	67.05	49.11	63.96	70.13	64.70

5.2.2 PSCL

In this section we thoroughly explore the performance of the model with and without the PSCL. PSCL successfully addresses the core issue of partial-activation, leading to a significant increase in pseudo-label quality and overall performance across both datasets. The best results for PSCL are shown in Table 5.6 and Table 5.7.

Table 5.6: Performance comparison with and without PSCL on LUAD-HistoSeg dataset.

Model	TE	NEC	LYM	TAS	mIoU	fwIoU
Swin (Baseline)	63.98	**70.46**	70.04	62.92	66.85	64.88
Swin + PSCL	**66.70**	67.90	**71.97**	**64.69**	**67.81**	**66.75**

Table 5.7: Performance comparison with and without PSCL on BCSS-WSSS dataset.

Model	TUM	STR	LYM	NEC	mIoU	fwIoU
Swin (Baseline)	73.33	63.90	48.55	57.90	60.92	66.18
Swin + PSCL	**76.82**	**66.92**	**50.72**	**61.82**	**64.07**	**69.36**

The introduction of a contrastive loss was based on the premise that the feature space would become more separable, with less discriminative features becoming more discriminative and thus addressing the problem of activation that CAMs are susceptible to. Whilst the increase in mIOU across both datasets is evidence of this, we further confirm

this by examining the CAM activations across the two datasets.

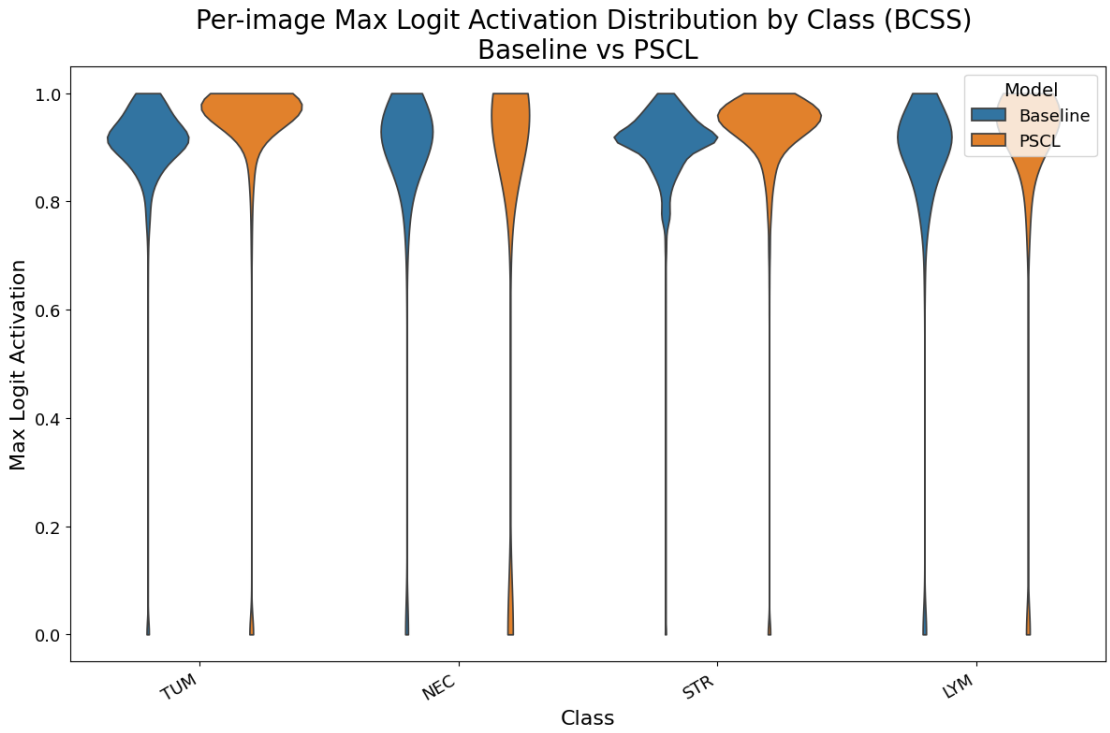

Figure 5.1: CAM activation for BCSS-WSSS dataset.

Clearly, Figures 5.1 and 5.2 show that, excluding the NEC class from the BCSS-WSSS dataset, PSCL leads to an increased number of high activation CAMs when compared to the baseline CAMs. As the CAMs are generated relative to the maximum activation for a given class, this provides direct evidence that PSCL increases the activation of previously low activated features, thereby addressing the issue of partial-activation that CAMs are susceptible to. Whilst it is perhaps evident that PSCL achieves this by clustering features, we further confirm this by analysing the inter-class feature distance. In both datasets, we see the addition of the PSCL leads to an increased density of high similarity features when compared to the baseline CAMs.

As already outlined through comparison with SOTA methods, the BCSS-WSSS dataset benefits from the PSCL approach more significantly than the LUAD-HistoSeg dataset. One possible reason for this is the difference in dataset size. The BCSS-WSSS dataset is around 40% larger than the LUAD-HistoSeg dataset.

Qualitative Analysis

The performance increases that PSCL achieves are not solely quantitative; PSCL results in significantly more refined segmentation masks and class activations when compared to

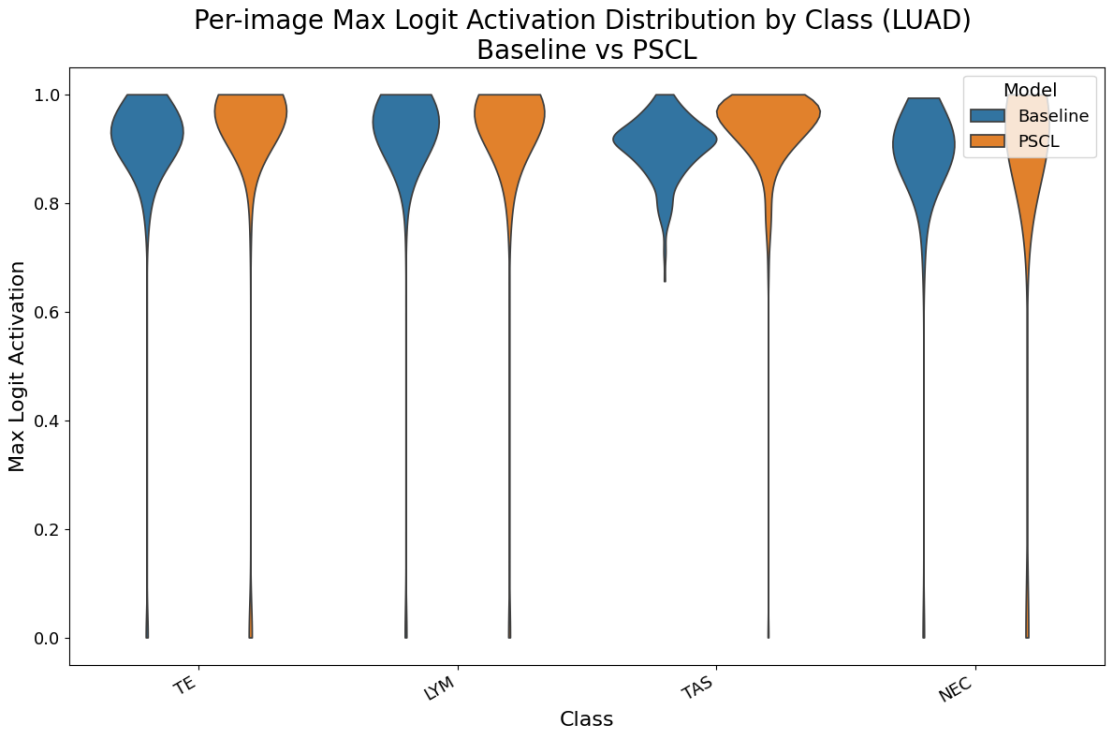

Figure 5.2: CAM activation for LUAD-HistoSeg dataset.

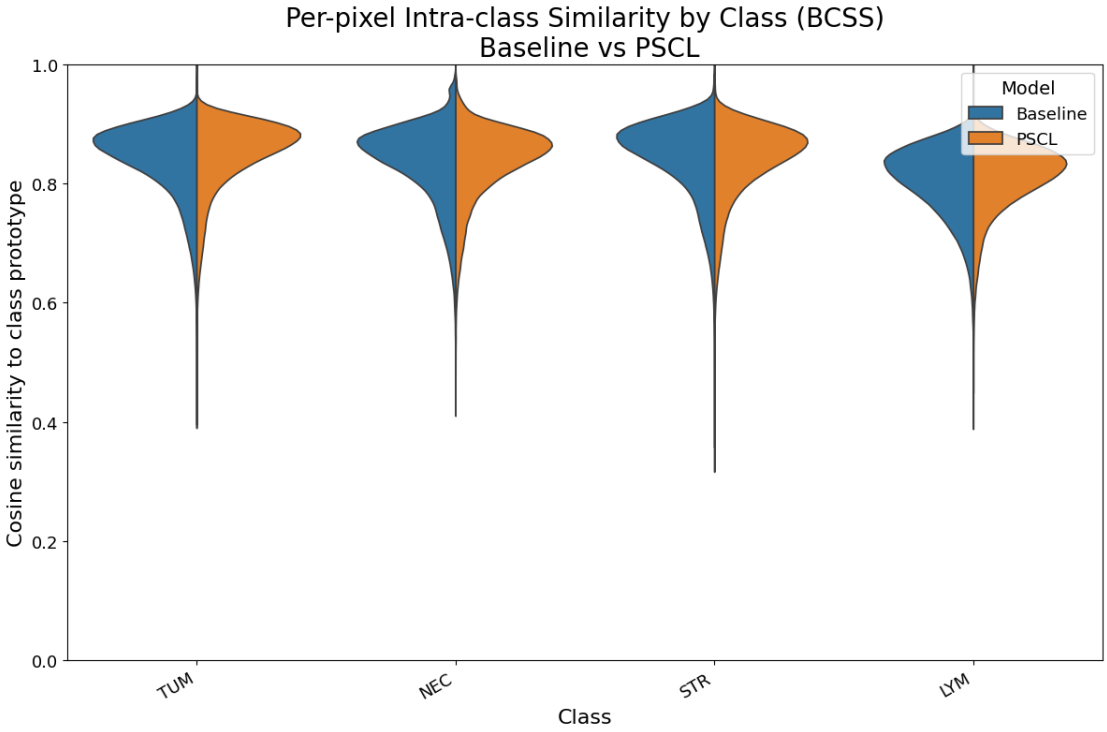

Figure 5.3: Intra-class feature distance for BCSS-WSSS dataset.

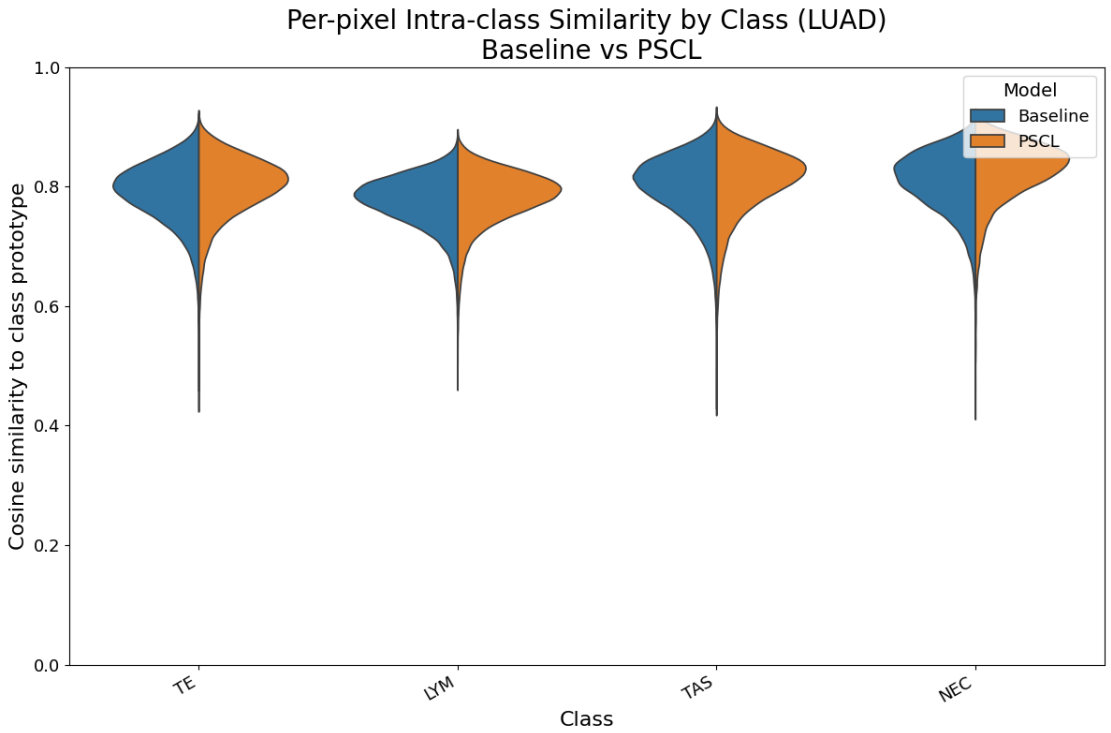

Figure 5.4: Intra-class feature distance for LUAD-HistoSeg dataset.

the baseline approach. This can be seen in Figures 5.5 and 5.6. Additional visualisations can be found in Appendix A.3.

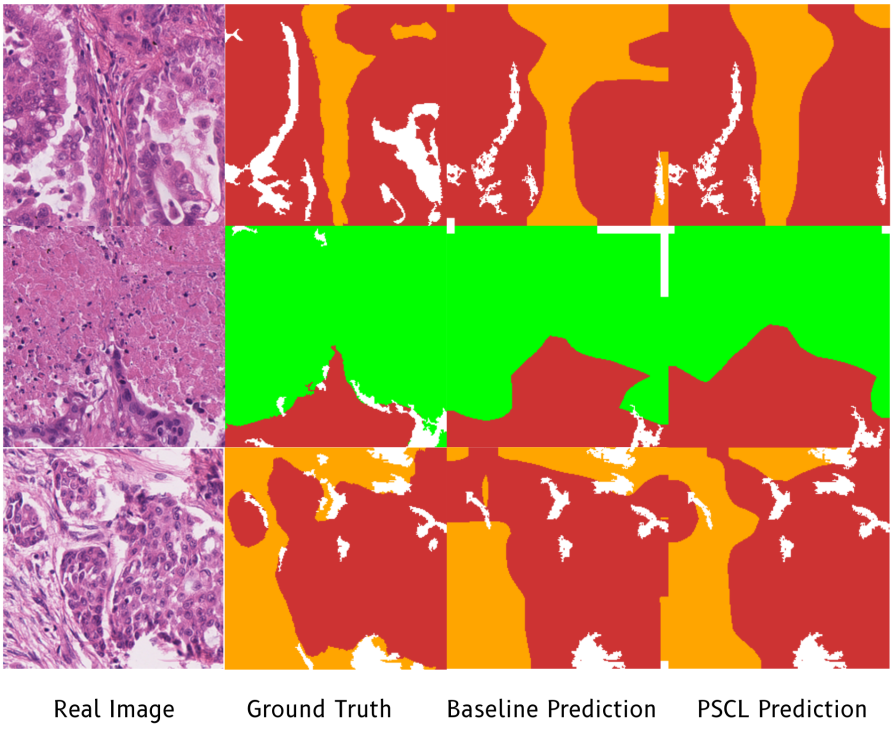

Figure 5.5: Comparison of pseudo-labels between baseline and PSCL on the LUAD-HistoSeg dataset.

We highlight that the PSCL tends to expand large activated regions of the image whilst eliminating small activated regions. This is in line with the finding that the PSCL tends to increase the activation of previously poorly activated features. Although, the expansion of activated regions can result in over-activation in some cases, likely as a result of the interaction between the interpolation from the CAMs to the true image size and the PSCL implementation; Even without PSCL, activated regions of the image are inherently likely to be larger than their true image counterparts. The addition of PSCL, which we have shown is able to increase the activation of previously poorly activated features resulting in a higher average activation, potentially exacerbates this issue. When contrastive loss is applied to a reduced spatial feature space, the learned representations become more discriminative between classes, which can cause region boundaries to strongly activate for multiple classes. As we take the `argmax` of the CAMs to generate our pseudo-labels, we don't consider the nuance of two (or more) classes being strongly activated in the same region. This boundary sharpening effect may lead to the loss of nuanced spatial information that exists in the transition zones between different tissue types.

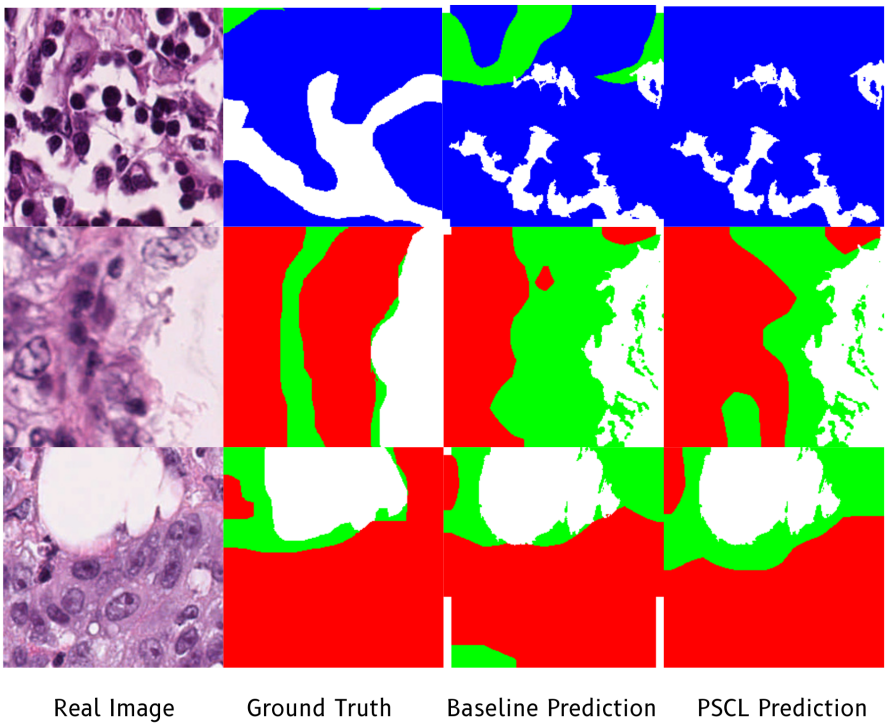

Figure 5.6: Comparison of pseudo-labels between baseline and PSCL on the BCSS-WSSS dataset.

For further insight, we examine the qualitative activations of the models across both datasets. The heatmaps can be seen in Figures 5.7 and 5.8.

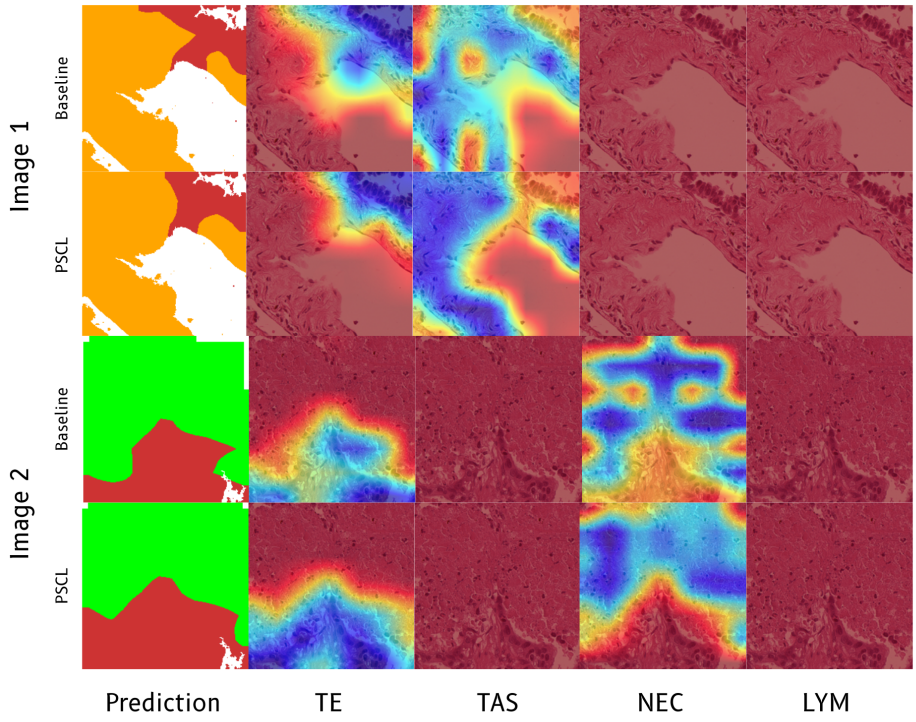

Figure 5.7: Qualitative activation of the model on the LUAD-HistoSeg dataset.

Across both datasets, we see much more consistent activations for the present classes when compared to the baseline activations. Region boundaries are more clearly defined and the contrast between activated regions and non-activated regions within a class is more pronounced. This trend occurs regardless of whether there are significant differences in the final pseudo-label, which is further evidence of PSCL addressing the issue of partial-activation. This is a particularly significant finding. CAMs are often used to increase the interpretability of a model by providing a way to visualise the decisions of the model. The need for this interpretability is particularly evident in the medical domain, where the decisions of the model could be used to guide clinical decision-making. The significantly more refined activation heatmaps produced by the PSCL approach are likely to be more useful for this purpose.

Weighting of PSCL (λ_{PSCL})

We conduct ablation studies on the weighting of the PSCL loss (λ_{PSCL}). As shown in 5.9, we find that the performance of the model on both datasets is maximised when the

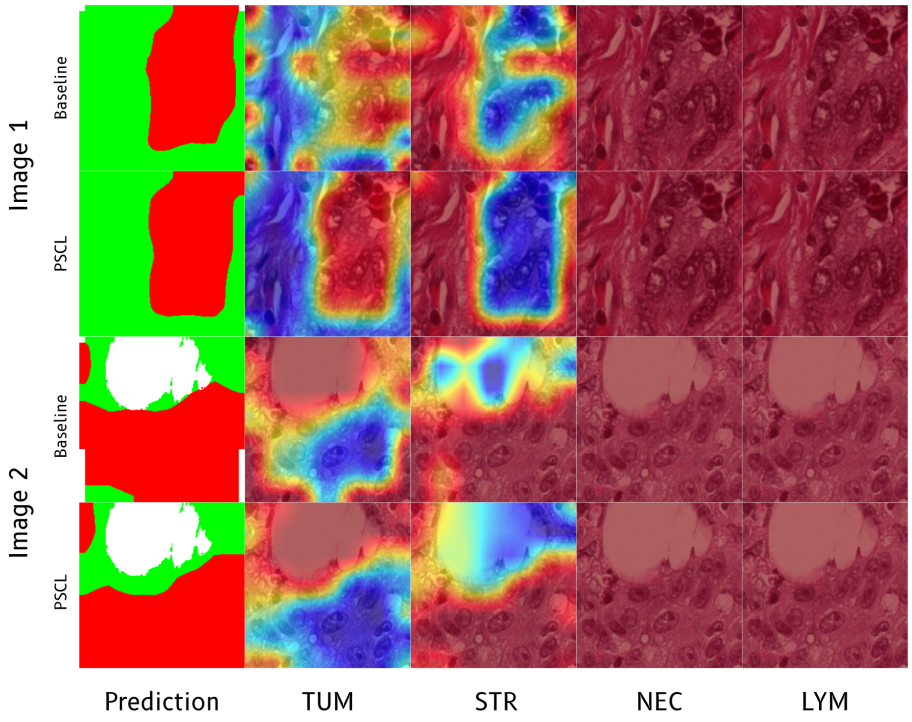

Figure 5.8: Qualitative activation of the model on the BCSS-WSSS dataset.

lambda is set to 0.1. This suggests that the PSCL loss is best applied as a regulariser for the classification task, rather than as the primary loss function.

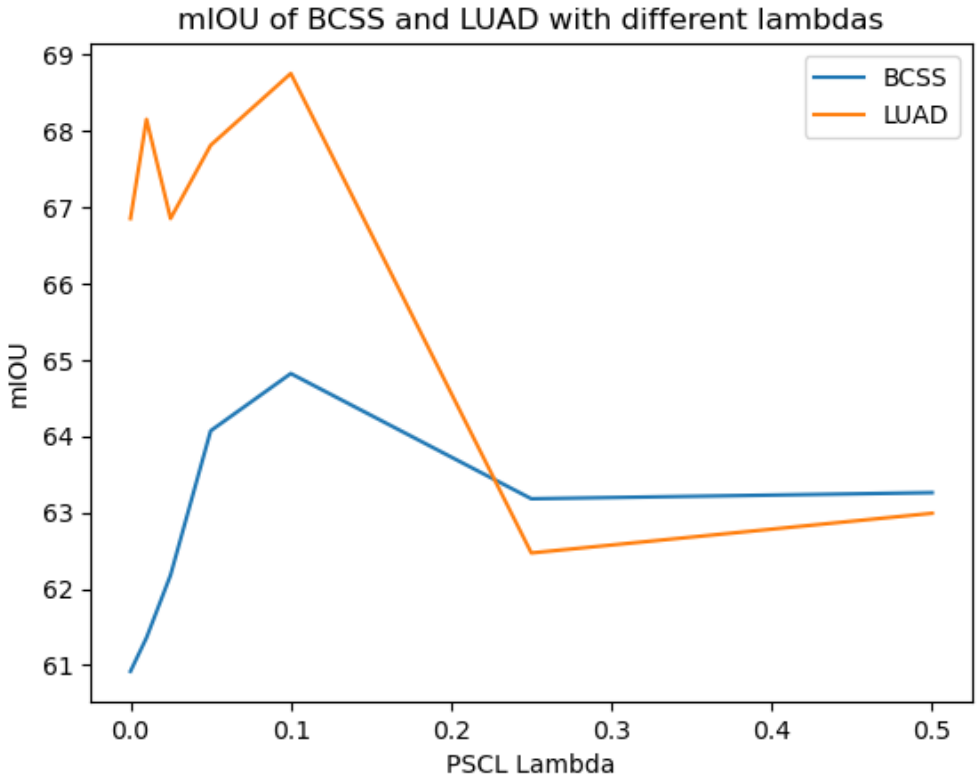

Figure 5.9: Effect of PSCL lambda on model performance.

Whilst a weight of 0.1 appears to be a relatively low weighting when compared to the MLSM loss, but it is worth nothing that the PSCL implementation makes use of an average non-zero reducer, meaning that the loss produced is not averaged strictly per-instance, producing a higher loss value overall.

Activation Threshold

As part of the PSCL loss, we apply a threshold to the activation values of the CAMs to remove particularly poorly activated regions. This is based on the premise that poorly activated regions may not necessarily be representative of the class to which they are currently assigned and thereby can work against the convergence of the feature space. To confirm this, we conduct ablation studies on the activation threshold, as shown in 5.10. We find that across both datasets, the performance of the model is maximised by smaller thresholds of around 0.1 to 0.25.

If we once again consider the problem of partial-activation, it is not necessarily surprising that the performance of the model is maximised by lower activation thresholds. The

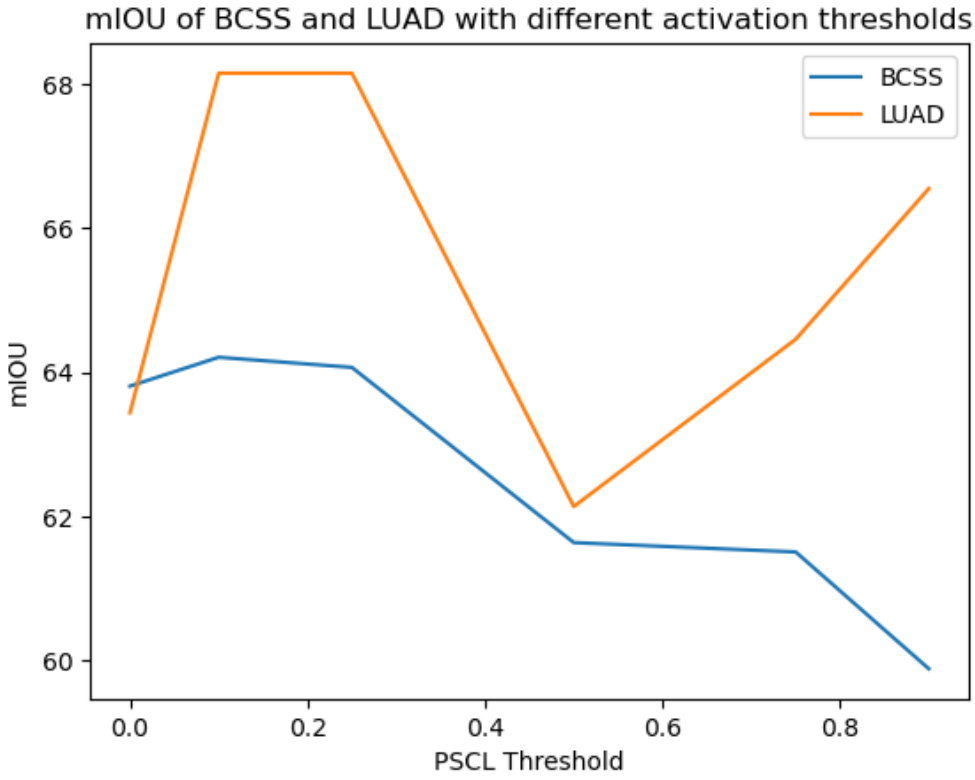

Figure 5.10: Effect of PSCL threshold on model performance.

motivation behind the PSCL loss is to help increase the activation of the poorly activated regions, which is more difficult to achieve when the threshold is higher. That being said, a higher threshold can still help progress the separation of the feature space, given that features of the same class are likely to be similar and thereby have similar gradients with respect to the loss. Although, their gradient contribution is likely to be decreased as they produce both higher activation values for the MLSM loss and have increased intra-class feature similarity. Furthermore, the impact of the threshold is likely to decrease as training progresses. As aforementioned, and demonstrated in 5.1 and 5.2, the addition of the PSCL shifts the distribution of the CAMs towards higher activation values, meaning that fewer features are excluded by the threshold as training progresses.

These results show that our approach is more sensitive to the activation thresholds for determining 'good features' than other similar approaches such as Kang et al. (2025) who demonstrated reduced sensitivity to the activation thresholds. We attribute this to the use of a contrastive loss which actively guides the clustering of the feature space, rather than the passive approach of Kang et al. (2025) who simply use it to guide the selection of features on which they perform K-means clustering.

Background Seeding

Our method proposes the use of a background seeding mechanism to help identify which regions of the image are likely to be the background. This is imperative to ensure that the SupConLoss is able to correctly cluster the features of the background class. We find that simply thresholding the CAMs, with a fixed value, to identify the background region is not sufficient, and the use of a colour thresholding combined with morphological operations is more effective.

Table 5.8: Comparison of performance when background class is not included in the PSCL.

Background Seeding	**LUAD-HistoSeg**		**BCSS-WSSS**	
	mIoU	**fwIoU**	**mIoU**	**fwIoU**
Activation Thresholding	65.46	63.98	56.52	63.08
Colour Thresholding (Ours)	**70.60**	**69.40**	**66.22**	**71.60**

Whilst it is clear that the use of a colour thresholding combined with morphological operations is more effective than simply thresholding the CAMS it is an oversimplification of the problem. Firstly, it relies on the assumption that the background class is solely the background, which isn't the case for the BCSS-WSSS dataset. Secondly, the more accurate background mask is interpolated to the PSCL feature space, which is a reduction in spatial information and likely leads to an over-activation of the background class. That is all to say that features that are classified as the background class through this method are not necessarily the background class. We investigate this by examining the performance of the model when the background class, and thereby features of this class,

are not included in the PSCL. We find that the performance of the LUAD-HistoSeg dataset is significantly reduced, where the performance of the BCSS-WSSS dataset is slightly improved.

Table 5.9: Comparison of performance when background class is not included in the PSCL.

Background Included	LUAD-HistoSeg		BCSS-WSSS	
	mIoU	fwIoU	mIoU	fwIoU
Yes	**70.60**	**69.40**	**66.22**	**71.60**
No	64.07	63.84	64.60	70.18

Impact on Training

Our integration of PSCL as part of the training process, rather than using a contrastive loss as a secondary step such as in Chaitanya et al. (2021) is a particularly important distinction. We underscore this by examing the impact of the PSCL loss when applied post an initial classification training: We train the model on the classification task for 5 epochs and then introduce the PSCL loss to the model both independently and jointly with the classification loss. We find that neither approach replicates the performance of initially training with PSCL combined with the classification loss. Additionally, solely performing the PSCL loss alone unsurprisingly significantly degrades performance.

Table 5.10: Comparison of performance when background class is not included in the PSCL.

Post-Training	LUAD-HistoSeg		BCSS-WSSS	
	mIoU	fwIoU	mIoU	fwIoU
None (MLSM + PSCL Pre-Training)	68.75	67.35	64.21	70.02
PSCL	23.07	25.61	19.27	24.02
MLSM + PSCL	63.48	61.91	63.82	69.52

Clearly, PSCL is an intrinsic component for corrective feature learning, not a post-hoc filter. We present two justifications for this effect. Firstly, classification training inherently optimises for partial activation. The motivating problem of this work is partial activation, which is well underscored in literature as a significant problem with traditional CAM based approaches which solely train on the classification task (Han et al., 2021; Kang et al., 2025). Partially activated features are those which are not important for the classification task and therefore are unlikely to be labelled correctly. As such, using the CAMs as pseudo-labels for the contrastive loss is not effective. This is furthered reinforced by the second justification, that PSCL requires global feature reorganisation. Such global feature reorganisation is particularly evident in the poor

performance of the model when trained solely on the PSCL loss; the optimal clustering of the feature space is not that which best suits the classification task.

Backbone Choice

Under the premise that an ideal methodology is backbone-agnostic, we assess a variety of backbone models to determine the impact of the backbone choice on the performance of the model and to confirm whether the PSCL approach is backbone-agnostic. We find that the Swin Transformer performs significantly better than the ResNet50, with the SwinV2, and ConvNeXt performing similarly to the Swin Transformer. We suggest therefore that the transformer-based or inspried backbones are the ideal backbones for the PSCL approach. Given its ubiquity and ease of use, we validate our hypothesis of making use of the Swin Transformer as the backbone for the PSCL approach with these results. The results for each dataset can be found in Tables 5.11 and 5.12.

Table 5.11: Performance comparison of different backbone architectures on LUAD-HistoSeg dataset.

Backbone	**TE**	**NEC**	**LYM**	**TAS**	**mIoU**	**fwIoU**
Swin	66.70	67.90	71.97	64.69	67.81	66.75
ResNet50	53.60	43.98	57.05	53.08	51.93	53.15
SwinV2	66.31	69.99	**74.56**	64.80	68.92	67.13
ConvNeXt	**67.19**	**74.34**	69.13	**66.56**	**69.31**	**67.75**

Table 5.12: Performance comparison of different backbone architectures on BCSS-WSSS dataset.

Backbone	**TUM**	**STR**	**LYM**	**NEC**	**mIoU**	**fwIoU**
Swin	76.82	66.92	50.72	**61.82**	**64.07**	69.36
ResNet50	70.55	61.08	46.19	54.40	58.06	63.40
SwinV2	**78.60**	**67.61**	48.36	60.44	63.75	**70.12**
ConvNeXt	75.23	65.59	**48.07**	60.27	62.29	67.77

Whilst the similar performance across the Swin and ConvNeXt models is not notable, the significantly degraded performance of the ResNet50 model is. This can likely be attributed to the lack of global context information of the ResNet50 model. Whilst the Swin Transformer incorporates a similar hierarchical architecture to the ResNet50 model, it importantly includes a self-attention mechanism which allows for the capture of global context information (Liu et al., 2021). Similarly, the ConvNeXt explicitly attempts to implement transformer-like features, such as patch-embeddings and depth-wise convolutions, which allows for the capture of global context information (Liu et al., 2022). Further confirmation of the importance of these transformer-like architectures

for SSL is implicitly provided by their success in major SSL tasks such as DINO (Caron et al., 2021).

5.2.3 Pre-Training

As is common in the literature, we make use of ImageNet (Deng et al., 2009b) pre-trained Weights for the backbone models in the classification task. Unsurprisingly, we find that the use of pre-trained weights significantly improves the performance of the model.

Table 5.13: Performance (%) of models pre-trained on ImageNet for classification on LUAD-HistoSeg and BCSS-WSSS datasets.

Pre-Training	**LUAD-HistoSeg**		**BCSS-WSSS**	
	mIoU	**fwIoU**	**mIoU**	**fwIoU**
Randomly Initialised	35.62	43.35	44.64	55.11
ImageNet Pre-Trained	**68.7**	**67.35**	**64.7**	**70.13**

Whilst the use of natural image pre-trained weights is popular in the literature and does significantly improve model performance, we also examine the impact of producing histopathology or dataset specific pre-trained weights. Such weights are produced by means of the unsupervised method described in Section 4.2. In Table 5.14 we investigate the impact on training on each dataset independently and jointly and find that the performance of the resulting model is mixed, generally performing worse than the use of ImageNet pre-trained weights.

Table 5.14: Comparison of performance of models pre-trained on LUAD-HistoSeg and BCSS-WSSS datasets and joint weights.

Pre-Trained Dataset	**LUAD-HistoSeg**		**BCSS-WSSS**	
	mIoU	**fwIoU**	**mIoU**	**fwIoU**
LUAD-HistoSeg	62.45	63.18	57.54	64.43
BCSS-WSSS	55.68	59.23	61.41	67.26
Joint	**65.89**	**63.84**	**63.10**	**69.27**

It may seem that our findings are in contradiction with the literature, which has demonstrated that unsupervised pre-training on a large dataset of unannotated images can significantly improve model performance, we suggest this is more indicative of the constraints of this approach as implemented in this work. It is well established that pre-training, particularly unsupervised pre-training benefits significantly from a large varied dataset Caron et al. (2021); Zeng et al. (2025a); Ciga et al. (2021). Additionally, most training processes make use of larger batch sizes to increase the input to the contrastive loss (SimCLR use batch sizes of 256 to 8192 Chen et al. (2020)), which is not the case in this work; we make use of a batch size of 6 due to hardware constraints.

At the same we also observe several trends which are in line with the literature. We find that between the single dataset weights, the performance of the model is maximised when the model is pre-trained on the downstream dataset. Additionally, we find that the joint weights perform significantly better than the single-dataset weights. This is indicative of the increased generalisation of the model through exposure to both datasets.

We also conduct an ablation study on the weighting of the intra-image similarity loss $\mathcal{L}_{intra}$ on the joint dataset to extend the work of Zeng et al. (2025a). We find only a slight correlation between the weighting of the inter-image similarity loss and the performance of the model; the inter-image similarity loss is more important for the BCSS-WSSS dataset than the LUAD-HistoSeg dataset.

Table 5.15: Comparison of Cross Entropy and Noise Reduced Losses in Stage 2 training on LUAD-HistoSeg and BCSS-WSSS datasets.

Intra-Image Weighting	**LUAD-HistoSeg**		**BCSS-WSSS**	
	mIoU	**fwIoU**	**mIoU**	**fwIoU**
0.0	**67.18**	**67.21**	<u>63.90</u>	<u>69.34</u>
0.25	62.66	65.14	**64.70**	**70.13**
0.5	<u>64.96</u>	<u>65.89</u>	63.10	69.27
1.0	64.17	65.74	64.52	69.97

5.2.4 Gating Mechanism

We find mixed results when examining the impact of gating the CAMs with the classification scores which suggest that the gating mechanism is not an essential part of the weakly supervised training approach. Whilst performance on the BCSS-WSSS dataset is improved, performance on the LUAD-HistoSeg dataset is slightly reduced. We additionally find that the gating mechanism is not particularly sensitive to the threshold (τ in Equation 4.3.2) used.

Table 5.16: Ablation study of the gating mechanism on LUAD-HistoSeg: per-class mIoU and fwIoU for different gating thresholds.

Gating	**TE**	**NEC**	**LYM**	**TAS**	**mIoU**	**fwIoU**
None	67.86	73.18	71.36	65.5	69.48	67.85
Gating ($\epsilon = 0$)	67.49	72.39	69.82	65.28	68.75	67.35
Gating ($\epsilon = 0.25$)	67.22	72.39	69.51	65.10	68.56	67.13
Gating ($\epsilon = 0.5$)	67.13	72.39	69.02	64.93	68.37	66.96
Gating ($\epsilon = 0.75$)	67.11	72.39	68.40	65.13	68.26	66.94

Our findings of minimal to mixed impact from the gating mechanism somewhat contradict that of Han et al. (2021) who find a 0.4% improvement in mIoU when gating

Table 5.17: Ablation study of the gating mechanism on BCSS-WSSS: per-class mIoU, overall mIoU, and fwIoU for different gating thresholds.

Gating	**TUM**	**STR**	**LYM**	**NEC**	**mIoU**	**fwIoU**
None	78.05	66.44	48.29	60.62	63.35	69.39
Gating ($\epsilon = 0$)	78.66	67.05	49.11	63.96	64.70	70.13
Gating ($\epsilon = 0.25$)	78.77	67.16	49.27	64.39	64.90	70.26
Gating ($\epsilon = 0.5$)	78.85	67.23	49.41	64.18	64.92	70.33
Gating ($\epsilon = 0.75$)	77.60	66.90	49.80	61.10	63.85	69.70

the CAMs with the classification scores. A possible explanation regarding the minimal and mixed impact of the gating mechanism is that the PSCL's ability to enforce high-quality, semantically consistent CAMs may negate the need for traditional post-processing heuristics like classification-score-based gating, further simplifying the overall weakly-supervised workflow

In terms of the threshold, it is not particularly surprising that the performance differs only slightly across the different thresholds given that not performing the gating mechanism maintains an mIoU of within 2% of the performance of the gating mechanism. When we consider the sigmoid function in the gating mechanism 4.3.2 which has the general purpose of shifting the logits towards 0 or 1 based on the threshold, it becomes obvious that only classes that are uncertain about their average logits (i.e an average logit around 0) would be excluded.

To attempt to explain why the gating mechanism performs differently on the two datasets, we examine the classification performance of the model under the belief that the model is a better classifier on the BCSS-WSSS dataset than the LUAD-HistoSeg dataset.

Table 5.18: Classification performance (accuracy, F1-score, precision, recall) on LUAD-HistoSeg and BCSS-WSSS datasets.

Dataset	**Accuracy**	**F1 Score**	**Precision**	**Recall**
LUAD-HistoSeg	94.22	94.46	98.53	90.70
BCSS-WSSS	90.51	88.60	89.90	87.33

Surprisingly, we find that the LUAD-HistoSeg dataset performs better on the classification task than the BCSS-WSSS dataset across all metrics. If we turn to the recall and precision metrics of both datasets, we find that the BCSS-WSSS dataset has a more balanced recall and precision (89.90 and 87.33 respectively), whereas the LUAD-HistoSeg dataset's precision outweighs its recall (98.53 and 90.70 respectively). This precision-recall imbalance suggests that the LUAD-HistoSeg model is overconfident in its negative predictions. If the classifier wrongly predicts a class is absent (a "miss" that lowers recall), the gating mechanism then incorrectly suppresses a valid CAM, creating a false

negative in the segmentation output and lowering the IoU.

5.3 Weakly Supervised Training (Stage 2)

Introducing the second stage of training in which we trained the supervised task with the pseudo-labels produced by the first stage of training, we demonstrate an increase in performance across both datasets of around 1%. We find that the noise reduced loss performs similarly to the CrossEntropy loss, with a slight improvement in performance on the LUAD-HistoSeg dataset.

Table 5.19: Performance comparison between Stage 1 and Stage 2 training on LUAD-HistoSeg dataset.

Model	TE	NEC	LYM	TAS	mIoU	fwIoU
Swin (Stage 1)	66.70	67.90	**71.97**	64.69	67.81	66.75
Swin (Stage 2)	**70.42**	**73.67**	71.81	**66.51**	**70.60**	**69.40**

Table 5.20: Performance comparison between Stage 1 and Stage 2 training on BCSS-WSSS dataset.

Model	TUM	STR	LYM	NEC	mIoU	fwIoU
Swin (Stage 1)	76.82	66.92	50.72	**61.82**	64.07	69.36
Swin (Stage 2)	**79.41**	**68.91**	**52.63**	63.91	**66.22**	**71.60**

5.3.1 Qualitative Analysis

Akin to stage 1, we compare the pseudo-labels produced by the model in stage 2 to those produced by the model in stage 1. We find that the pseudo-labels produced by the model in stage 2 are slightly more refined, but the difference in quality is not as significant as the difference between the baseline and the PSCL approach in stage 1.

Generally, we see that the decoder produces higher fidelity, more refined outputs when compared to its input pseudo-labels. As seen in Figures 5.11 and 5.12, Tissue regions have jagged edges, more closely follow obvious tissue boundaries and are generally more accurate in their representation of the true image. This can be attributed to the significant increase in resolution of the feature maps that the decoder operates on when compared to the input pseudo-labels. The decoder receives feature maps from all stages of the encoder, compared to the input pseudo-labels which are interpolated from the last stage of the encoder (7×7 resolution). It is also evident that the decoder favours representing larger regions of the image when compared to the input pseudo labels. Small partitions such as bands or isolate islands of tissue are likely to be ignored or conglomerated into larger regions. This perhaps

One large benefit of the PSCL approach is the strength of the activations produced by the model, and thus the explainability of the model. We compare the activation

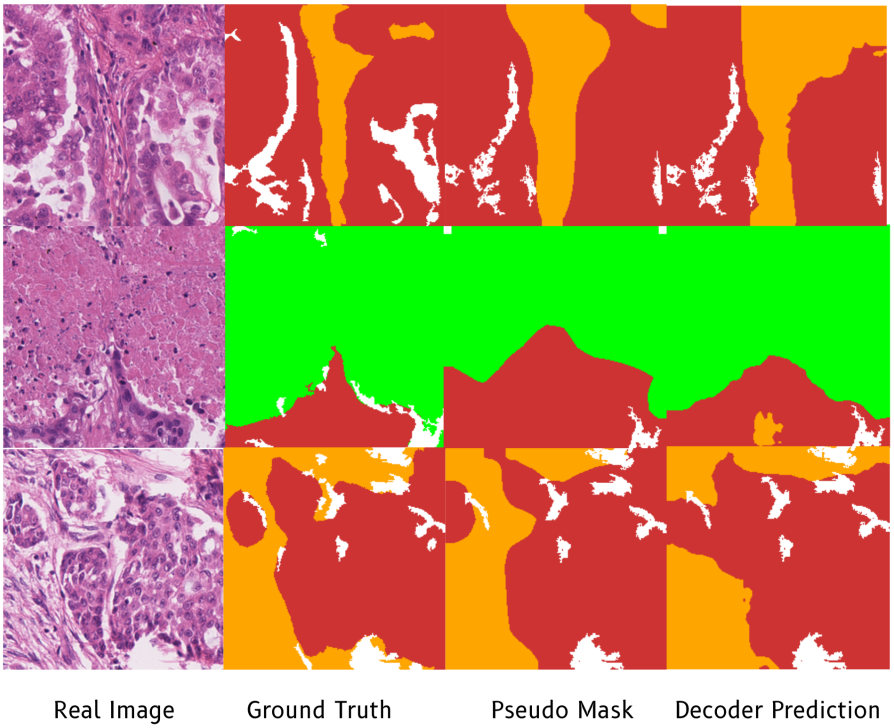

Figure 5.11: Comparison of pseudo-labels between Stage 1 and Stage 2 on the LUAD-HistoSeg dataset.

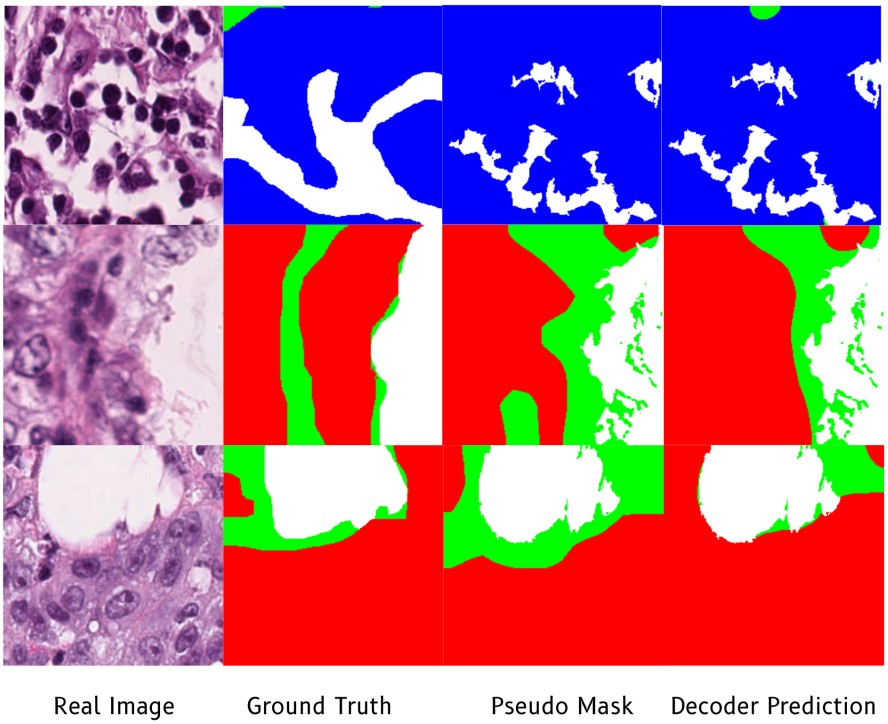

Figure 5.12: Comparison of pseudo-labels between Stage 1 and Stage 2 on the BCSS-WSSS dataset.

heatmaps between the PSCL and decoder on both datasets in Figures 5.13 and 5.14 and find them to be similar. Although, we visualise that the decoder's activations are significantly more noisy, with less contrast between activated and non-activated regions. One explanation for this is the lack of `ReLU` activation on the decoder's output, which is present in the CAM generation process. This is particularly evident in the classes which aren't present for a given image, such as NEC and LYM in Image 1 of 5.13; PSCL produces entirely zero (and therefore uniform activations). Additionally, we return to the PSCL shifting the distribution of the activations towards a higher average activation, resulting in less intermediary activations between the present and non-present classes.

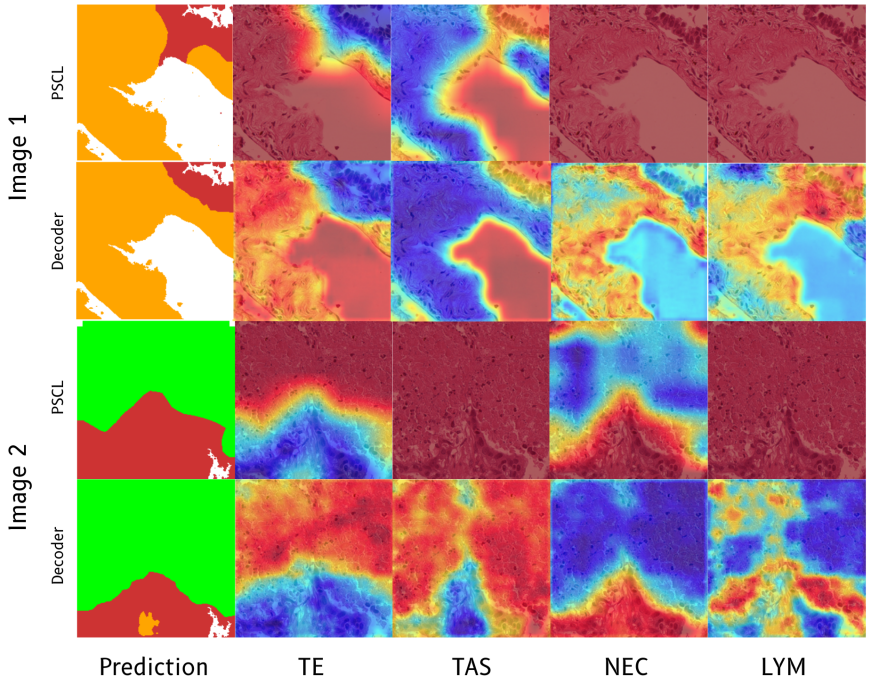

Figure 5.13: Comparison of activations between Stage 1 and Stage 2 on the LUAD-HistoSeg dataset.

5.3.2 Comparison with State of the Art

In comparing our approach to state-of-the-art methods, we observe similar standings to the PSCL approach: competitive performance on the LUAD-HistoSeg dataset, and a second place finish on the BCSS-WSSS dataset. Table 5.21 and Table 5.22 show the top 3 pseudo-label-based supervised state-of-the-art methods and our approach on the LUAD-HistoSeg and BCSS-WSSS datasets respectively.

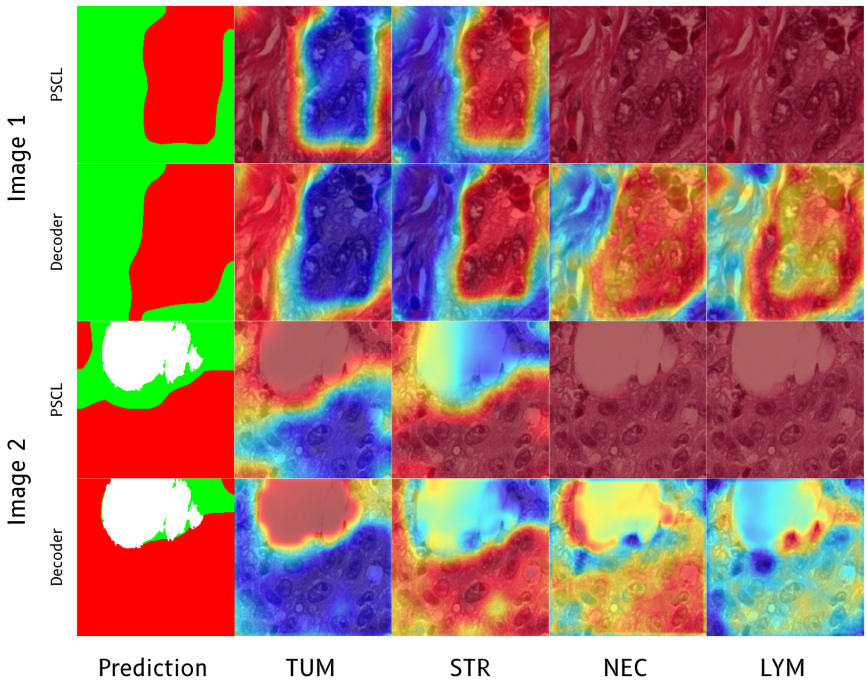

Figure 5.14: Comparison of activations between Stage 1 and Stage 2 on the BCSS-WSSS dataset.

Table 5.21: Top 3 supervised state-of-the-art methods and our approach on LUAD-HistoSeg dataset (%).

Method	TE	NEC	LYM	TAS	fwIoU	mIoU
UAM (Kang et al., 2025)	**78.62**	**82.31**	**79.03**	**73.31**	**76.98**	**78.31**
TPRO (Zhang et al., 2023)	75.80	80.56	78.14	72.69	75.31	76.80
MLPS (Han et al., 2021)	73.90	77.48	73.61	69.53	72.51	73.63
Ours	70.42	73.66	71.81	66.51	69.40	70.60

Table 5.22: Top 3 supervised state-of-the-art methods and our approach on BCSS-WSSS dataset (%).

Method	TUM	STR	LYM	NEC	fwIoU	mIoU
UAM (Kang et al., 2025)	**79.89**	**74.66**	**64.71**	**70.88**	**75.76**	**70.88**
TPRO (Zhang et al., 2023)	77.95	65.10	54.55	64.96	67.36	65.64
MLPS (Han et al., 2021)	74.54	64.45	52.54	58.67	66.48	62.55
Ours	79.41	68.91	52.63	63.91	71.60	66.22

5.3.3 Noise Reduced Loss

Despite claims of up to 3% performance gains when using the NR loss (Kang et al., 2025), we find that the NR loss degrades performance when compared to the CrossEntropy loss.

Table 5.23: Comparison of Cross Entropy and Noise Reduced Losses in Stage 2 training on LUAD-HistoSeg and BCSS-WSSS datasets.

Loss Function	**LUAD-HistoSeg**		**BCSS-WSSS**	
	mIoU	**fwIoU**	**mIoU**	**fwIoU**
Cross Entropy	**70.60**	**69.40**	**66.22**	**71.60**
Noise Reduced	69.15	68.42	63.42	68.96

We argue that the performance degradation likely suggests that the NR loss and other attempts to account for the uncertainty of the pseudo-labels generated in the first stage of training may not be aligned with the PSCL approach. The NR Loss aims to account for the uncertainty of the pseudo-labels generated in the first stage of training by weighting the loss by the confidence score of the pseudo-labels (Kang et al., 2025). We have already demonstrated that the PSCL approach shifts the distribution of the activations towards a higher average activation which addresses the issue of partial activation (Figures 5.1 and 5.2). However, this shift potentially undermines the uncertainty mechanism of the NR Loss, as the model is now more confident in its predictions, even when its predictions are incorrect. Furthermore, if we consider the NR Loss equation 4.4, we find that as the confidence score goes to 1, the loss becomes the standard BCE loss 2.17. The performance degradation is thus not unexpected, as CE better models the task of picking one true class per pixel rather than the multi-class classification task for which BCE is more appropriate (Goodfellow et al., 2016).

Chapter 6

Concluding Remarks

6.1 Conclusion

This thesis thoroughly investigates the development of weakly-supervised semantic segmentation approaches for histopathology. Driven by the desire to apply machine learning approaches to aid in diagnosis but confronted by the high cost of obtaining high-quality per-pixel labels that are required for fully supervised training, we, as do others, turn to the weak supervision paradigm. By alleviating the need for high-quality per-pixel labels that are required for fully supervised training, we increase accessibility to machine learning approaches to aid in diagnosis. This outlines the primary motivation for this work: to investigate and advance weakly-supervised segmentation approaches as a means to reduce this annotation burden and increase the accessibility of automated tissue analysis.

However, the well-documented limitation of partial-activation commonly seen in existing weakly-supervised segmentation approaches poses a significant challenge to the performance of such approaches. As such, we attempted to address this issue by introducing the Pseudo-Supervised Contrastive Loss (PSCL), a novel loss function that leverages a model's own Class Activation Maps as pseudo-labels to learn a more semantically separable feature space, directly addressing the issue of partial activation. We demonstrate that PSCL is highly effective in achieving this goal, both qualitatively and quantitatively, and as a result achieves significant performance gains and more rapid convergence. Of note is the strong performance of PSCL on the BCSS-WSSS dataset, achieving a second-in-class performance. Additionally, we provide an overview of the wholistic weakly-supervised training approach, examining the motivation for and impact of various components of a Weakly Supervised Semantic Segmentation (WSSS) approach.

The findings of this work validate the effectiveness of weakly-supervised segmentation approaches in the histopathology domain, confirming that they can achieve strong perfor-

mance with respect to fully supervised approaches. Furthermore, we provide a tangible method (PSCL) that can help address a well-known limitation in the field, potentially making computer-aided diagnosis tools more accessible and faster to develop.

6.2 Future Work

Whilst we have performed a thorough evaluation of the end-to-end approach to weakly-supervised segmentation, there are a number of potential future directions for this work.

6.2.1 CAMs

A fundamental limitation of CAM approaches in general is their input is lower spatial-resolution features produced by an encoder and thereby lose spatial resolution. Whilst this has the possible advantage of not diluting the classification across a significant number of features it means interpolation of the CAMs to the original image size is required. The greater the difference in resolution, the greater the loss of spatial information. Some attempts have been made to increase the resolution of the feature maps by extracting them from earlier layers of the encoder or using methods such as Grad-CAM to back-propagate the weak supervision signal to the earlier layers of the encoder.

Further investigation into increasing the resolution of the image features produced by the encoder could provide a significant improvement in performance. The use of constant feature map resolution ViTs and reducing patch sizes or window sizes in Swin Transformers could provide a starting point for such an investigation.

6.2.2 PSCL

We identify that the performance of the contrastive loss is dependent on the accuracy of the pseudo-labels, particularly for the background class. As such, methods which better identify background regions and can more effectively cluster its features could provide a significant improvement in performance. Such an approach would ideally bring the performance gain of the LUAD-HistoSeg dataset in line with that of the BCSS-WSSS dataset as we identified that identification of the background class is particularly important for this dataset.

The applicability of the PSCL approach to other datasets with more complex tissues and larger number of classes should be investigated to assess its generalisation ability. This is particularly important for datasets with many classes as contrastive losses are susceptible to the problem of class imbalance. Datasets with more or less background tissue could help reveal the importance of the background class in the PSCL approach.

Weighting the individual loss for each feature embedding by the confidence score of its pseudo-label could provide a way to counter for low quality pseudo-labels in addition to or as a replacement for the confidence threshold. In this way, features which are less

representative of a class or perhaps incorrectly attributed to a class do not contribute as much to the movement of the feature space.

6.2.3 Unsupervised Pre-training

Whilst not explored deeply in this work, the use of unsupervised pre-training techniques to assist the weakly-supervised training approach could be further investigated to improve performance. Given the results of incorporating unsupervised weights into the weakly-supervised training approach, where we find that in some settings, domain-specific unsupervised pre-training can match and in some cases better performance from industry standard pre-trained datasets, we believe this is a promising avenue for further investigation. Whilst the medical image domain has seen significant investment in pre-training such as Zeng et al. (2025a), their application to Histopathology has been limited. As such, the application of unsupervised pre-training techniques to the histopathology domain could be further investigated to improve performance.

Appendix A

State of the Art Results

A.1 Stage 1 (Pseudo Labels)

Table A.1: Performance comparison on the LUAD-HistoSeg dataset (%).

Method	TE	NEC	LYM	TAS	fwIoU	mIoU
CAM [9]	69.66	72.62	72.58	66.88	69.59	70.44
Grad-CAM (Selvaraju et al., 2019)	70.07	66.01	70.18	64.76	67.81	67.76
SC-CAM (Chang et al., 2020)	68.29	64.28	62.07	61.79	64.73	64.10
TransWS (Zhang et al., 2022)	65.92	60.16	73.34	69.11	67.67	67.13
MLPS (Han et al., 2021)	71.72	76.27	73.53	67.67	70.80	72.30
SIPE (Chen et al., 2022a)	72.68	62.44	63.86	64.11	64.74	65.77
HAMIL (Yang et al., 2023)	72.82	69.79	69.82	70.96	71.50	70.85
TPRO (Zhang et al., 2023)	74.82	77.55	76.40	70.98	73.81	74.94
UAM (Kang et al., 2025)	**76.24**	**80.43**	**76.28**	**72.02**	**75.38**	**76.24**

Table A.2: Performance comparison on the BCSS-WSSS dataset (%).

Method	TUM	STR	LYM	NEC	fwIoU	mIoU
CAM [9]	66.83	58.71	49.41	51.12	60.96	56.52
Grad-CAM (Selvaraju et al., 2019)	65.96	56.71	43.36	30.04	58.27	49.02
SC-CAM (Chang et al., 2020)	64.28	56.16	42.87	30.14	56.96	48.36
TransWS (Zhang et al., 2022)	64.85	58.17	44.96	50.60	59.42	54.64
MLPS (Han et al., 2021)	70.76	61.07	50.87	52.94	63.89	58.91
SIPE (Chen et al., 2022a)	72.36	62.88	47.85	48.72	63.46	57.95
HAMIL (Yang et al., 2023)	69.84	59.45	49.98	51.29	62.64	57.64
TPRO (Zhang et al., 2023)	77.18	63.77	54.95	61.43	68.55	64.33
UAM (Kang et al., 2025)	**78.97**	**71.72**	**58.16**	**63.59**	**72.20**	**68.11**

A.2 Stage 2 (Supervised Training)

Table A.3: Supervised performance comparison on the LUAD-HistoSeg dataset (%).

Method	TE	NEC	LYM	TAS	fwIoU	mIoU
Baseline	52.64	58.71	64.59	61.26	57.89	59.30
HistoSegNet (**?**)	45.59	36.30	58.28	50.82	48.54	47.75
TransWS (Zhang et al., 2022)	57.04	49.98	59.46	58.59	57.41	56.27
OEEM (**?**)	73.81	70.49	71.89	69.48	71.70	71.42
MLPS (Han et al., 2021)	73.90	77.48	73.61	69.53	72.51	73.63
SIPE (Chen et al., 2022a)	73.14	65.26	66.18	67.23	66.82	67.95
HAMIL (Yang et al., 2023)	73.46	75.83	72.94	70.86	72.60	73.27
TPRO (Zhang et al., 2023)	75.80	80.56	78.14	72.69	75.31	76.80
Ours	**78.62**	**82.31**	**79.03**	**73.31**	**76.98**	**78.31**

Table A.4: Supervised performance comparison on the BCSS-WSSS dataset (%).

Method	**TUM**	**STR**	**LYM**	**NEC**	**fwIoU**	**mIoU**
Baseline	45.89	51.89	43.54	43.65	48.15	46.24
HistoSegNet (**?**)	33.14	46.46	29.05	1.91	37.19	27.64
TransWS (Zhang et al., 2022)	44.71	36.49	41.72	38.08	40.61	40.25
OEEM (**?**)	74.86	64.68	48.91	61.03	66.34	62.37
MLPS (Han et al., 2021)	74.54	64.45	52.54	58.67	66.48	62.55
SIPE (Chen et al., 2022a)	73.29	63.87	49.28	52.49	64.77	59.73
HAMIL (Yang et al., 2023)	71.65	62.37	51.52	54.29	64.95	59.96
TPRO (Zhang et al., 2023)	77.95	65.10	54.55	64.96	67.36	65.64
Ours	**79.89**	**74.66**	**64.71**	**70.88**	**75.76**	**70.88**

A.3 Unsupervised Pre-training Transformations

Contrastive Loss Type	**Transformations Applied**
Intra-Image	• `RandomAdjustSharpness(factor=2, p=0.5)` • `RandomAutocontrast(p=0.5)` • `RandomEqualize(p=0.5)` • `GaussianBlur(kernel_size=3, sigma=(0.1, 2.0))` • `ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1)`
Inter-Image	• `RandomHorizontalFlip(p=0.5)` • `RandomVerticalFlip(p=0.5)` • `RandomRotation(degrees=(-90, 90))`

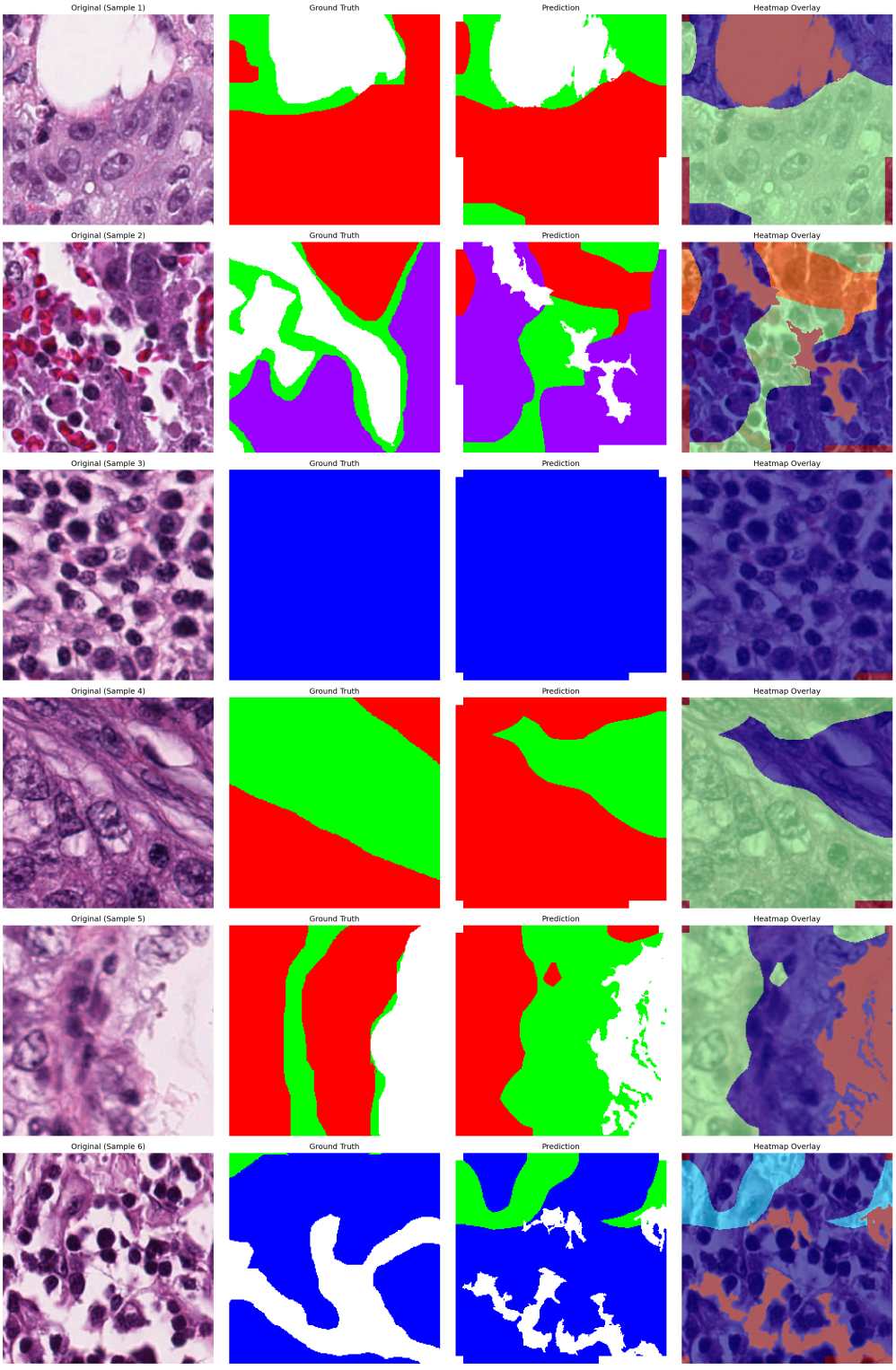

Figure A.1: BCSS Baseline Results

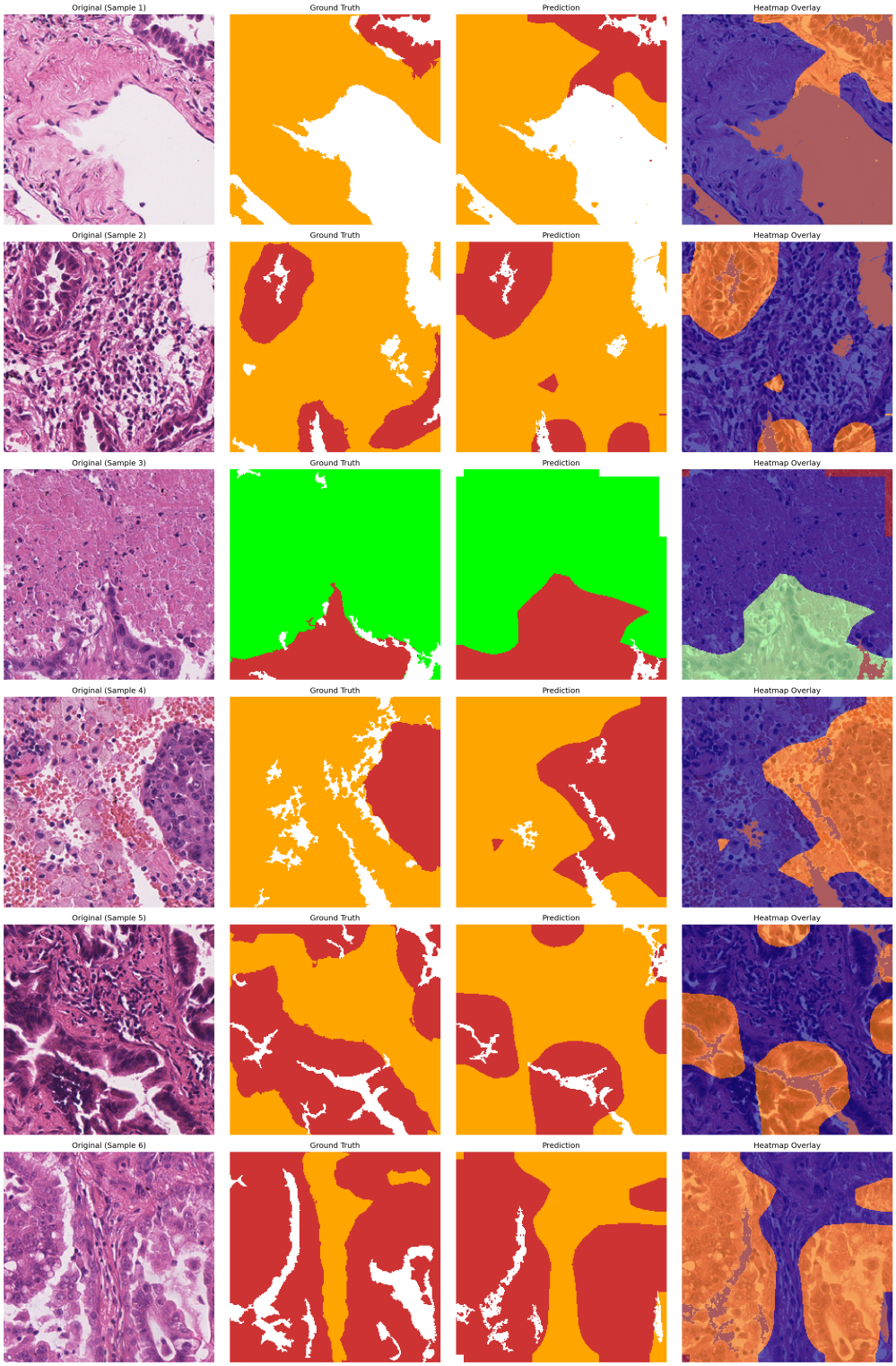

Figure A.2: LUAD Baseline Results

A.3.1 PSCL

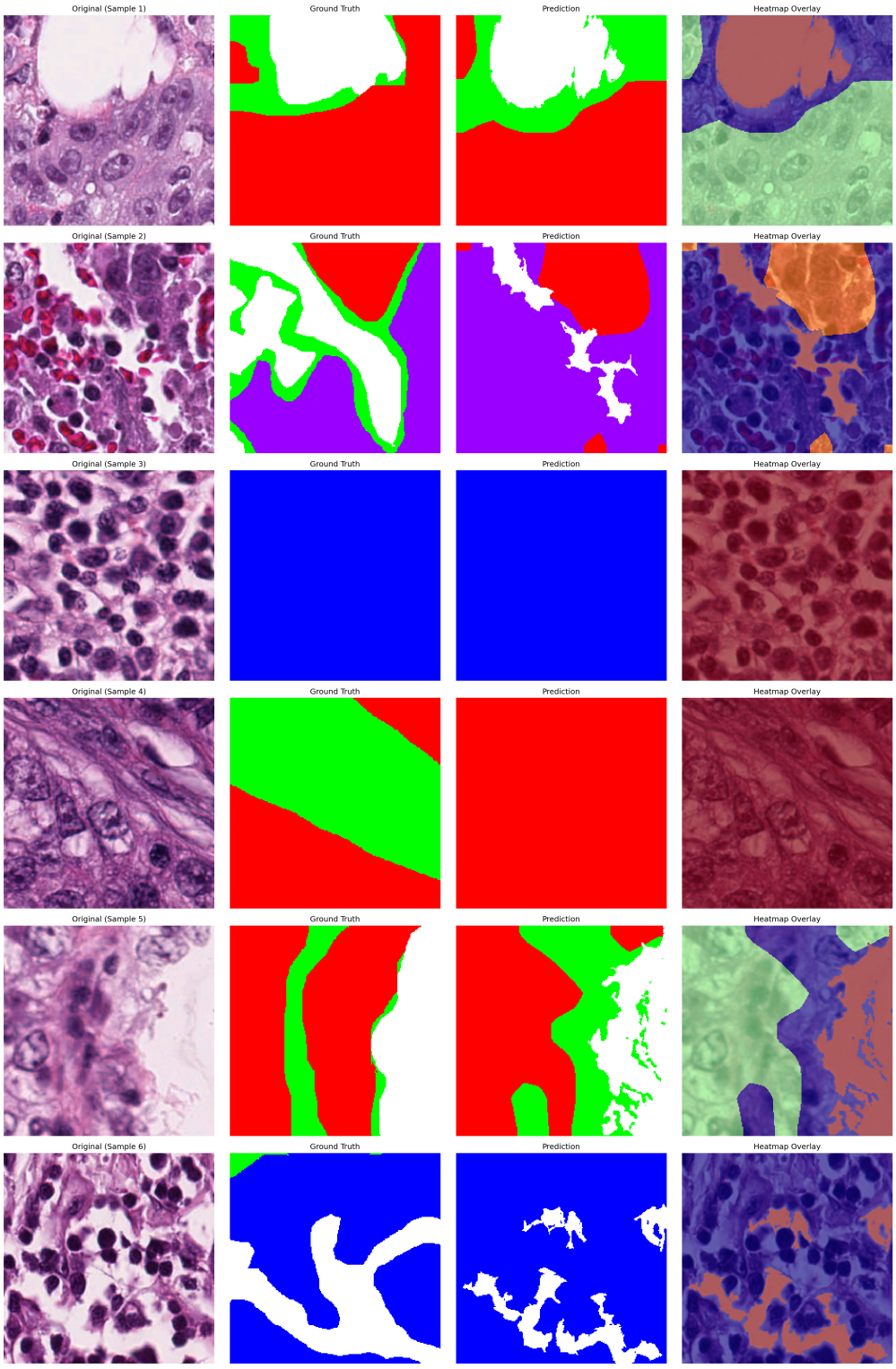

Figure A.3: BCSS PSCL Results

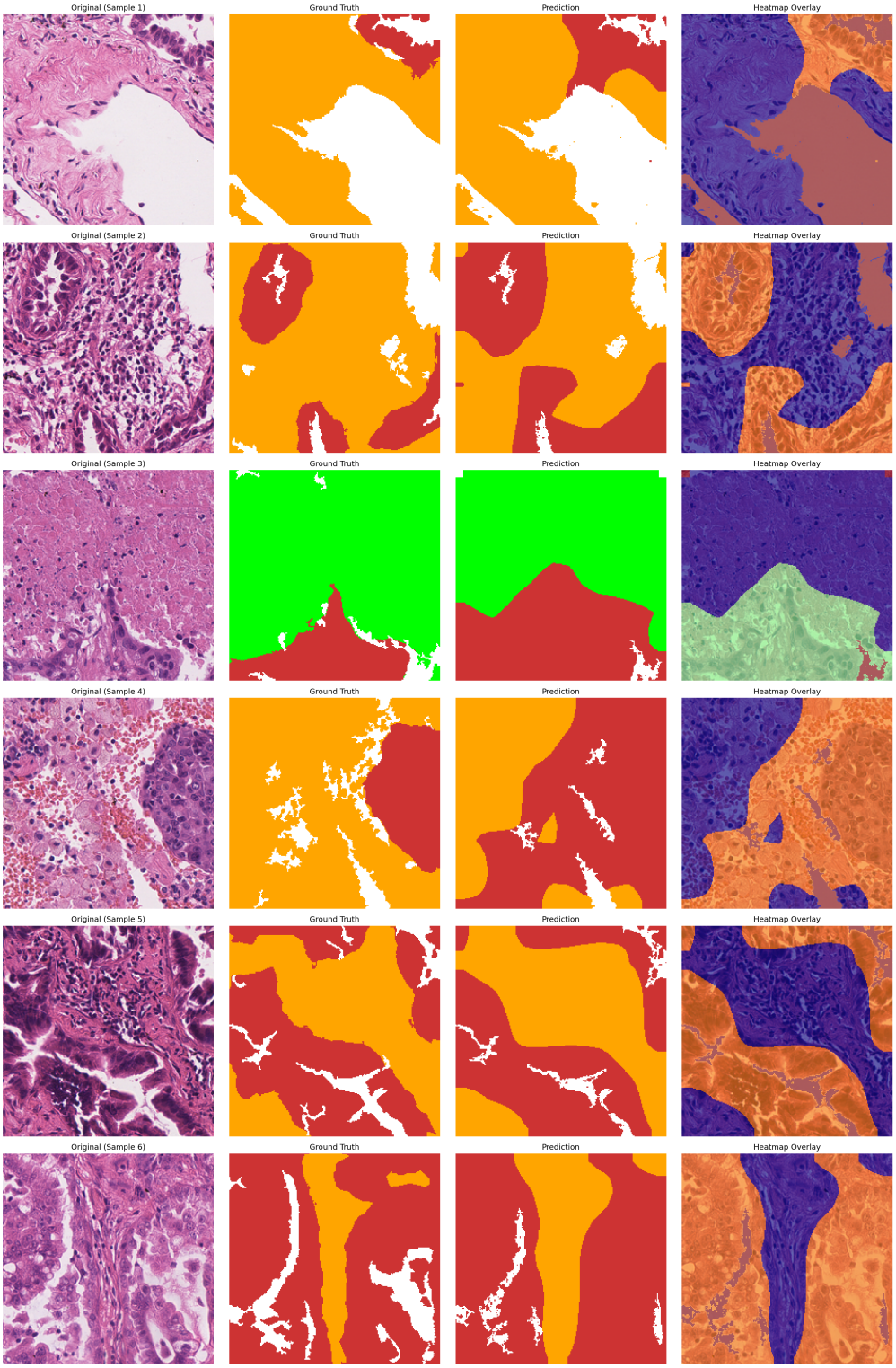

Figure A.4: LUAD PSCL Results

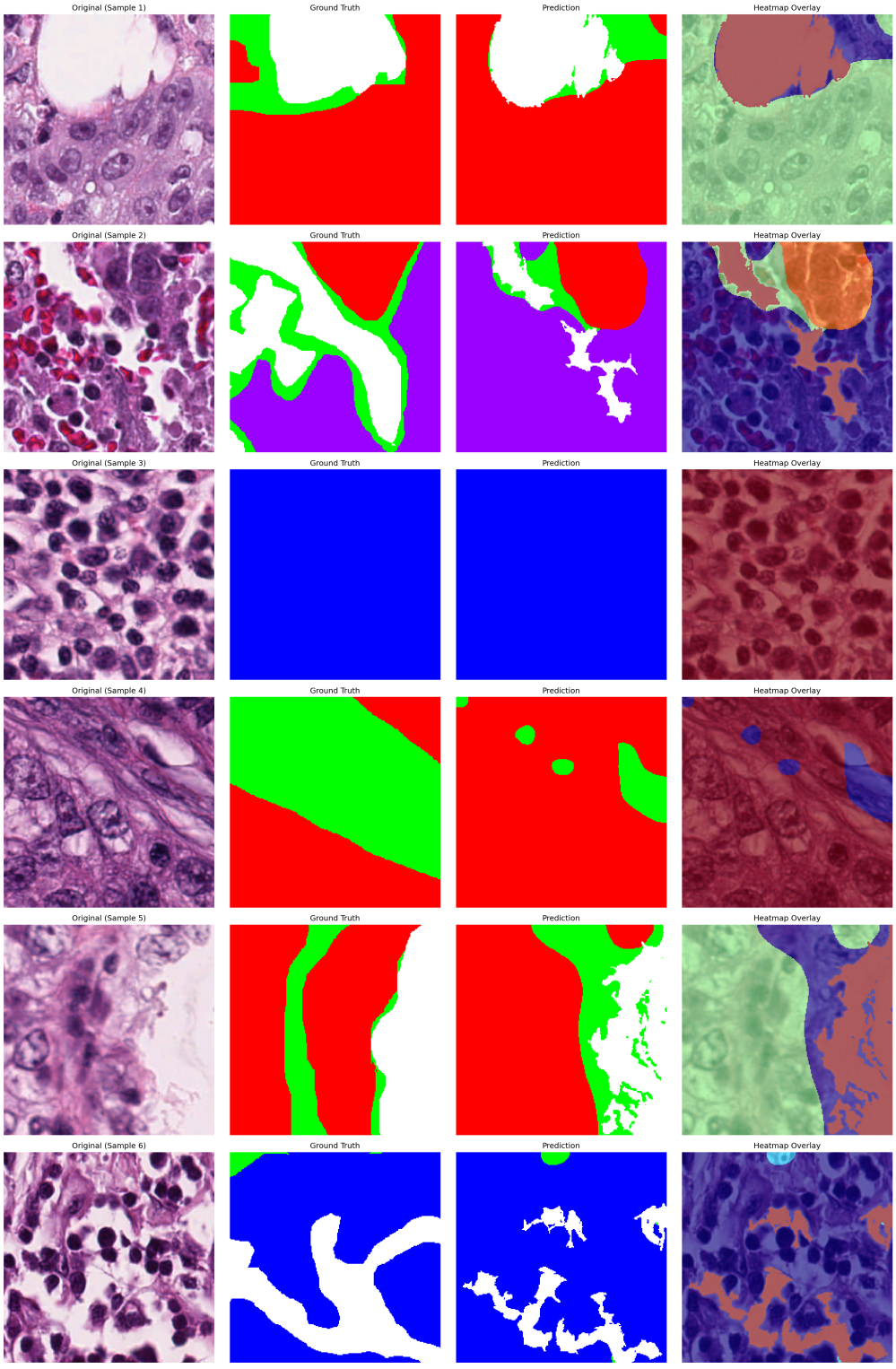

Figure A.5: BCSS Pseudo Label Results

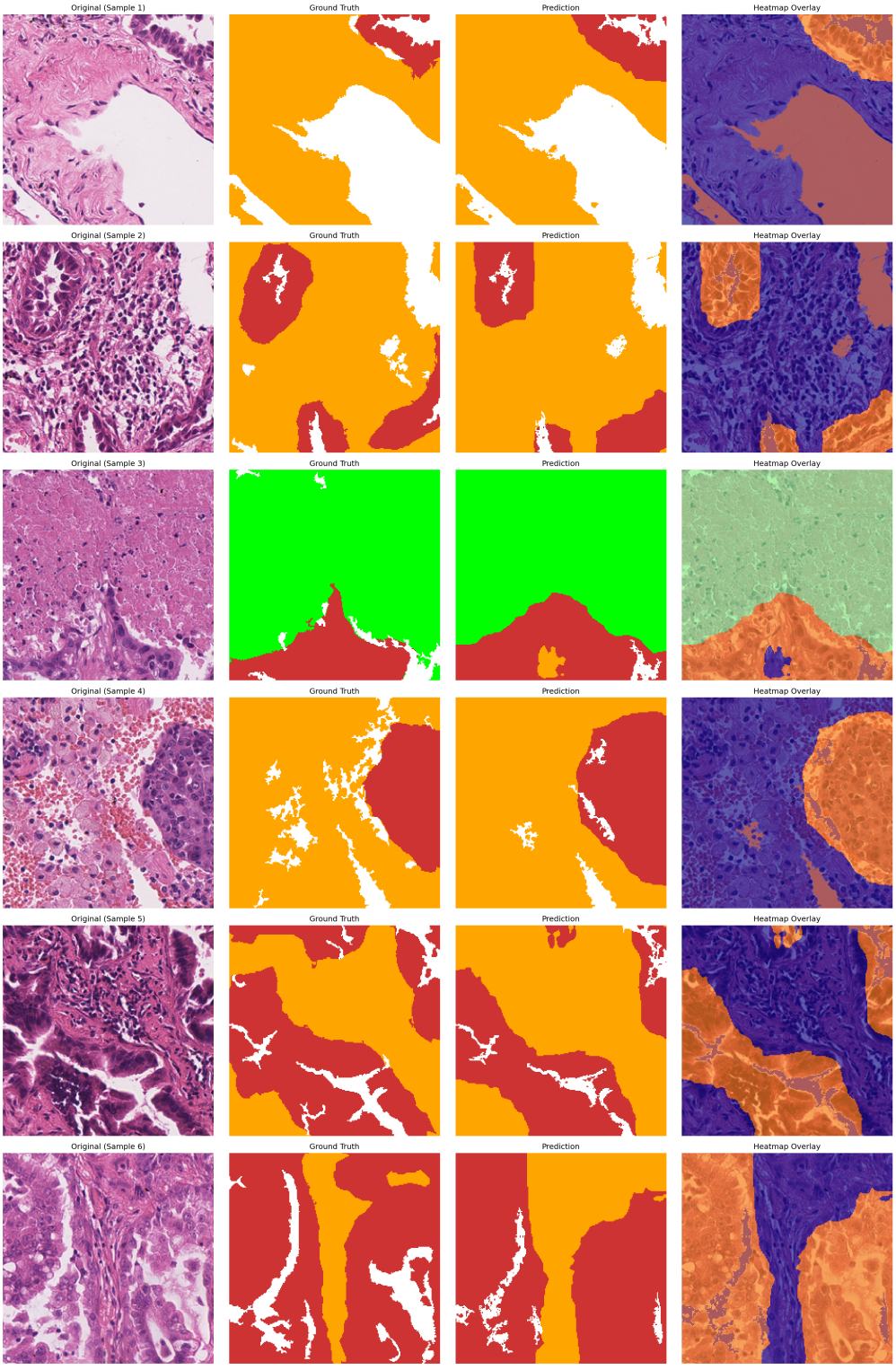

Figure A.6: LUAD Pseudo Label Results

Bibliography

Ahmadi, R. and Kasaei, S., 2024. Leveraging swin transformer for local-to-global weakly supervised semantic segmentation. https://arxiv.org/abs/2401.17828. [Cited on page 38.]

Ahn, J. and Kwak, S., 2018. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. https://arxiv.org/abs/1803.10464. [Cited on page 38.]

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L., 2016. What's the point: Semantic segmentation with point supervision. https://arxiv.org/abs/1506.02106. [Cited on page 36.]

Berman, A. G.; Orchard, W. R.; Gehrung, M.; and Markowetz, F., 2021. Pathml: A unified framework for whole-slide image analysis with deep learning. *medRxiv*, (2021). doi:10.1101/2021.07.07.21260138. https://www.medrxiv.org/content/early/2021/07/13/2021.07.07.21260138. [Cited on page 31.]

Buckner, C., 2019. Deep learning: A philosophical introduction. *Philosophy Compass*, 14 (08 2019). doi:10.1111/phc3.12625. [Cited on page 21.]

Buhl, N., 2024. Semantic segmentation in computer vision: Full guide. Encord Blog. https://encord.com/blog/guide-to-semantic-segmentation/. [Cited on pages 29, 30, and 31.]

Bulten, W.; Pinckaers, H.; van Boven, H.; Vink, R.; de Bel, T.; van Ginneken, B.; van der Laak, J.; Hulsbergen-van de Kaa, C.; and Litjens, G., 2020. Automated deep-learning system for gleason grading of prostate cancer using biopsies: a diagnostic study. *The Lancet Oncology*, 21, 2 (Feb. 2020), 233–241. doi:10.1016/s1470-2045(19)30739-9. http://dx.doi.org/10.1016/S1470-2045(19)30739-9. [Cited on pages 2 and 8.]

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A., 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*. [Cited on pages 25, 28, 34, 36, 40, and 69.]

Chaitanya, K.; Erdil, E.; Karani, N.; and Konukoglu, E., 2021. Local contrastive loss with pseudo-label based self-training for semi-supervised medical image segmentation. https://arxiv.org/abs/2112.09645. [Cited on pages 34 and 67.]

Chan, L.; Hosseini, M. S.; Rowsell, C.; Plataniotis, K. N.; and Damaskinos, S., 2019. Histosegnet: Semantic segmentation of histological tissue type in whole slide images. In *The IEEE International Conference on Computer Vision (ICCV)*. [Cited on page 39.]

Chang, Y.-T.; Wang, Q.; Hung, W.-C.; Piramuthu, R.; Tsai, Y.-H.; and Yang, M.-H., 2020. Weakly-supervised semantic segmentation via sub-category exploration. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [Cited on pages 2, 21, 83, and 84.]

Chapelle, O.; Schölkopf, B.; and Zien, A. (Eds.), 2005. *Semi-Supervised Learning*. The MIT Press, Cambridge, Massachusetts and London, England. [Cited on page 18.]

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. https://arxiv.org/abs/1802.02611. [Cited on pages 35 and 39.]

Chen, Q.; Yang, L.; Lai, J.-H.; and Xie, X., 2022a. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4288–4298. [Cited on pages 2, 37, 38, 83, 84, and 85.]

Chen, R. J.; Chen, C.; Li, Y.; Chen, T. Y.; Trister, A. D.; Krishnan, R. G.; and Mahmood, F., 2022b. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. https://arxiv.org/abs/2206.02647. [Cited on pages 25, 28, and 40.]

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G., 2020. A simple framework for contrastive learning of visual representations. https://arxiv.org/abs/2002.05709. [Cited on pages 34 and 69.]

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R., 2022. Masked-attention mask transformer for universal image segmentation. [Cited on pages 21, 26, and 44.]

Cheng, B.; Schwing, A. G.; and Kirillov, A., 2021. Per-pixel classification is not all you need for semantic segmentation. https://arxiv.org/abs/2107.06278. [Cited on page 35.]

Ciga, O.; Xu, T.; and Martel, A. L., 2021. Self supervised contrastive learning for digital histopathology. https://arxiv.org/abs/2011.13971. [Cited on page 69.]

Csurka, G.; Volpi, R.; and Chidlovskii, B., 2023. Semantic image segmentation: Two decades of research. https://arxiv.org/abs/2302.06378. [Cited on page 30.]

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L., 2009a. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. doi:10.1109/CVPR.2009.5206848. [Cited on page 18.]

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L., 2009b. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. doi:10.1109/CVPR.2009.5206848. [Cited on page 69.]

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. https://arxiv.org/abs/2010.11929. [Cited on pages 24 and 25.]

Draelos, R. L. and Carin, L., 2021. Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks. https://arxiv.org/abs/2011.08891. [Cited on page 36.]

Gansbeke, W. V.; Vandenhende, S.; Georgoulis, S.; and Gool, L. V., 2021. Unsupervised semantic segmentation by contrasting object mask proposals. https://arxiv.org/abs/2102.06191. [Cited on pages 35 and 36.]

Goodfellow, I.; Bengio, Y.; and Courville, A., 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org. [Cited on pages 15, 16, 17, 18, 19, 20, and 78.]

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y., 2014. Generative adversarial networks. https://arxiv.org/abs/1406.2661. [Cited on page 9.]

Griebel, T.; Archit, A.; and Pape, C., 2025. Segment anything for histopathology. https://arxiv.org/abs/2502.00408. [Cited on page 40.]

Gurcan, M. N.; Boucheron, L. E.; Can, A.; Madabhushi, A.; Rajpoot, N. M.; and Yener, B., 2009. Histopathological image analysis: a review. *IEEE Rev Biomed Eng*, 2 (2009), 147–171. doi:10.1109/RBME.2009.2034865. [Cited on pages 6, 7, and 40.]

Hadsell, R.; Chopra, S.; and LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 1735–1742. doi:10.1109/CVPR.2006.100. [Cited on page 27.]

Han, C.; Lin, J.; Mai, J.; Wang, Y.; Zhang, Q.; Zhao, B.; Chen, X.; Pan, X.; Shi, Z.; Xu, X.; Yao, S.; Yan, L.; Lin, H.; Xu, Z.; Huang, X.; Han, G.; Liang, C.; and Liu, Z., 2021. Multi-layer pseudo-supervision for histopathology

tissue semantic segmentation using patch-level classification labels. https://arxiv.org/abs/2110.08048. [Cited on pages 2, 6, 18, 39, 47, 48, 52, 55, 67, 70, 77, 83, 84, and 85.]

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2015. Deep residual learning for image recognition. https://arxiv.org/abs/1512.03385. [Cited on pages 21, 22, 26, and 34.]

He, L.; Long, L. R.; Antani, S.; and Thoma, G. R., 2012. Histology image analysis for carcinoma detection and grading. *Computer Methods and Programs in Biomedicine*, 107, 3 (2012), 538–556. doi:10.1016/j.cmpb.2011.12.007. [Cited on page 6.]

Hornik, K.; Stinchcombe, M.; and White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2, 5 (1989), 359–366. doi:https://doi.org/10.1016/0893-6080(89)90020-8. https://www.sciencedirect.com/science/article/pii/0893608089900208. [Cited on page 14.]

Huang, K.; Altosaar, J.; and Ranganath, R., 2020. Clinicalbert: Modeling clinical notes and predicting hospital readmission. https://arxiv.org/abs/1904.05342. [Cited on page 39.]

Kang, Y.; Li, H.; Shi, X.; Zhang, X.; Xing, Y.; Wen, Y.; Wang, Y.; Cui, L.; Feng, J.; and Yang, L., 2025. Exploring Unbiased Activation Maps for Weakly Supervised Tissue Segmentation of Histopathological Images. *IEEE Transactions on Medical Imaging*, 44, 6 (jun 2025), 2631–2642. doi:10.1109/TMI.2025.3541115. [Cited on pages 1, 2, 18, 21, 37, 38, 39, 41, 47, 51, 55, 66, 67, 77, 78, 83, and 84.]

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D., 2021. Supervised contrastive learning. https://arxiv.org/abs/2004.11362. [Cited on page 33.]

Kingma, D. P. and Ba, J., 2017. Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980. [Cited on page 17.]

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R., 2023. Segment anything. https://arxiv.org/abs/2304.02643. [Cited on page 36.]

Komura, D.; Ochi, M.; and Ishikawa, S., 2025. Machine learning methods for histopathological image analysis: Updates in 2024. *Computational and Structural Biotechnology Journal*, 27 (2025), 383–400. doi:https://doi.org/10.1016/j.csbj.2024.12.033. https://www.sciencedirect.com/science/article/pii/S2001037024004549. [Cited on pages 1, 8, 9, and 34.]

Kruse, R.; Borgelt, C.; Klawonn, F.; Moewes, C.; Steinbrecher, M.; and Held, P., 2013. *Computational Intelligence - A Methodological Introduction.* Texts in Computer Science. Springer. ISBN 978-1-4471-5013-8. [Cited on pages 13, 14, and 15.]

Krähenbühl, P. and Koltun, V., 2012. Efficient inference in fully connected crfs with gaussian edge potentials. https://arxiv.org/abs/1210.5644. [Cited on page 38.]

Li, R.; Mai, Z.; Zhang, Z.; Jang, J.; and Sanner, S., 2023. Transcam: Transformer attention-based cam refinement for weakly supervised semantic segmentation. *Journal of Visual Communication and Image Representation*, 92 (Apr. 2023), 103800. doi: 10.1016/j.jvcir.2023.103800. http://dx.doi.org/10.1016/j.jvcir.2023.103800. [Cited on pages 25, 35, and 38.]

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J., 2016. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. https://arxiv.org/abs/1604.05144. [Cited on page 36.]

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S., 2017. Feature pyramid networks for object detection. https://arxiv.org/abs/1612.03144. [Cited on page 44.]

Liu, H.; Yang, H.; van Diest, P. J.; Pluim, J. P. W.; and Veta, M., 2024a. Wsi-sam: Multi-resolution segment anything model (sam) for histopathology whole-slide images. https://arxiv.org/abs/2403.09257. [Cited on page 40.]

Liu, Z.; Kainth, K.; Zhou, A.; Deyer, T. W.; Fayad, Z. A.; Greenspan, H.; and Mei, X., 2024b. A review of self-supervised, generative, and few-shot deep learning methods for data-limited magnetic resonance imaging segmentation. *NMR in Biomedicine*, 37, 8 (2024), e5143. doi:https://doi.org/10.1002/nbm.5143. https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/nbm.5143. [Cited on page 37.]

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. https://arxiv.org/abs/2103.14030. [Cited on pages 25, 26, 35, 38, 43, and 68.]

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S., 2022. A convnet for the 2020s. https://arxiv.org/abs/2201.03545. [Cited on pages 26 and 68.]

Long, J.; Shelhamer, E.; and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. https://arxiv.org/abs/1411.4038. [Cited on page 30.]

Loshchilov, I. and Hutter, F., 2019. Decoupled weight decay regularization. https://arxiv.org/abs/1711.05101. [Cited on page 17.]

Ma, J.; He, Y.; Li, F.; Han, L.; You, C.; and Wang, B., 2024. Segment anything in medical images. *Nature Communications*, 15, 1 (2024), 654. doi:10.1038/s41467-024-44824-z. https://doi.org/10.1038/s41467-024-44824-z. [Cited on page 41.]

Marrón-Esquivel, J.; Duran-Lopez, L.; Linares-Barranco, A.; and Dominguez-Morales, J. P., 2023. A comparative study of the inter-observer variability on gleason grading against deep learning-based approaches for prostate cancer. *Computers in Biology and Medicine*, 159 (04 2023), 106856. doi:10.1016/j.compbiomed.2023.106856. [Cited on page 8.]

Minsky, M. and Papert, S., 1969. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA. [Cited on page 14.]

Mohan, D. D.; Jawade, B.; Setlur, S.; and Govindaraj, V., 2023. Deep metric learning for computer vision: A brief overview. https://arxiv.org/abs/2312.10046. [Cited on pages 27 and 28.]

Moyes, A., 2019. *Deep learning for processing histopathology images*. Ph.D. thesis, Queen's University Belfast. https://pureadmin.qub.ac.uk/ws/portalfiles/portal/238764526/thesis.pdf. [Cited on page 6.]

Muhammad, M. B. and Yeasin, M., 2020. Eigen-cam: Class activation map using principal components. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE. doi:10.1109/ijcnn48605.2020.9206626. http://dx.doi.org/10.1109/IJCNN48605.2020.9206626. [Cited on page 36.]

Musgrave, K.; Belongie, S.; and Lim, S.-N., 2020. Pytorch metric learning. https://arxiv.org/abs/2008.09164. [Cited on page 53.]

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H. V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Howes, R.; Huang, P.-Y.; Xu, H.; Sharma, V.; Li, S.-W.; Galuba, W.; Rabbat, M.; Assran, M.; Ballas, N.; Synnaeve, G.; Misra, I.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P., 2023. Dinov2: Learning robust visual features without supervision. [Cited on page 40.]

Orchard Software, 2025. The growth of digital pathology adoption. Technical report, Orchard Software. https://www.orchardsoft.com/white-paper/the-growth-of-digital-pathology-adoption/. White Paper. [Cited on page 7.]

Radford, A.; Kim, J. W.; Chen, C.; Xu, M.; Goehring, G.; Meyer, G.; Park, T.; Shtedrits, A.; Fidelman, S.; Amodei, D.; et al., 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, (2021). [Cited on pages 36, 39, and 40.]

Ridnik, T.; Ben-Baruch, E.; Noy, A.; and Zelnik-Manor, L., 2021. Imagenet-21k pretraining for the masses. https://arxiv.org/abs/2104.10972. [Cited on pages 1 and 18.]

Rolls, G., 2025. An introduction to specimen processing. https://www.leicabiosystems.com/en-au/knowledge-pathway/an-introduction-to-specimen-processing/. [Cited on page 6.]

Ronneberger, O.; Fischer, P.; and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. https://arxiv.org/abs/1505.04597. [Cited on pages 31, 32, and 35.]

Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65, 6 (1958), 386. [Cited on page 13.]

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J., 1986. Learning representations by back-propagating errors. *Nature*, 323 (1986), 533–536. https://api.semanticscholar.org/CorpusID:205001834. [Cited on page 16.]

Ryu, J.; Song, H.; Lee, S.; Cho, S. I.; Shin, J.; Paeng, K.; and Pereira, S., 2025. Scorpion: Addressing scanner-induced variability in histopathology. https://arxiv.org/abs/2507.20907. [Cited on pages 7 and 8.]

Schneider, J., 2022. Foundation models in brief: A historical, socio-technical focus. https://arxiv.org/abs/2212.08967. [Cited on page 28.]

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D., 2019. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128, 2 (Oct. 2019), 336–359. doi:10.1007/s11263-019-01228-7. http://dx.doi.org/10.1007/s11263-019-01228-7. [Cited on pages 2, 36, 83, and 84.]

Tang, Q.; Fan, L.; Pagnucco, M.; and Song, Y., 2025. Prototype-based image prompting for weakly supervised histopathological image segmentation. https://arxiv.org/abs/2503.12068. [Cited on pages 33, 34, 35, and 39.]

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I., 2023. Attention is all you need. https://arxiv.org/abs/1706.03762. [Cited on pages 22, 23, and 24.]

Vorontsov, E.; Bozkurt, A.; Casson, A.; Shaikovski, G.; Zelechowski, M.; Liu, S.; Severson, K.; Zimmermann, E.; Hall, J.; Tenenholtz, N.; Fusi, N.; Mathieu, P.; van Eck, A.; Lee, D.; Viret, J.; Robert, E.; Wang, Y. K.; Kunz, J. D.; Lee, M. C. H.; Bernhard, J.; Godrich, R. A.; Oakley, G.; Millar, E.; Hanna, M.; Retamero, J.; Moye, W. A.; Yousfi, R.; Kanan, C.; Klimstra, D.; Rothrock, B.; and Fuchs, T. J., 2024. Virchow: A million-slide digital pathology foundation model. https://arxiv.org/abs/2309.07778. [Cited on pages 28 and 40.]

Wang, X.; Yang, S.; Zhang, J.; Wang, M.; Zhang, J.; Yang, W.; Huang, J.; and Han, X., 2022a. Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical Image Analysis*, 81 (2022), 102559. doi:https://doi.org/10.1016/j.media.2022.102559. https://www.sciencedirect.com/science/article/pii/S1361841522002043. [Cited on page 35.]

Wang, Z.; Wu, Z.; Agarwal, D.; and Sun, J., 2022b. Medclip: Contrastive learning from unpaired medical images and text. https://arxiv.org/abs/2210.10163. [Cited on pages 39 and 40.]

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M., 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Association for Computational Linguistics, Online. https://www.aclweb.org/anthology/2020.emnlp-demos.6. [Cited on page 44.]

Xu, Z.; Myronenko, A.; Yang, D.; Roth, H. R.; Zhao, C.; Wang, X.; and Xu, D., 2022. Clinical-realistic annotation for histopathology images with probabilistic semi-supervision: A worst-case study. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, 77–87. Springer Nature Switzerland, Cham. [Cited on pages 1, 2, and 8.]

Yang, Y.; Tu, Y.; Lei, H.; and Long, W., 2023. Hamil: Hierarchical aggregation-based multi-instance learning for microscopy image classification. *Pattern Recognition*, 136 (2023), 109245. doi:https://doi.org/10.1016/j.patcog.2022.109245. https://www.sciencedirect.com/science/article/pii/S0031320322007245. [Cited on pages 83, 84, and 85.]

Zarella, M. D.; Bowman;, D.; Aeffner, F.; Farahani, N.; Xthona;, A.; Absar, S. F.; Parwani, A.; Bui, M.; and Hartman, D. J., 2018. A practical guide to whole slide imaging: A white paper from the digital pathology association. *Archives of Pathology & Laboratory Medicine*, 143, 2 (10 2018), 222–234. doi:10.5858/arpa.2018-0343-RA. https://doi.org/10.5858/arpa.2018-0343-RA. [Cited on pages 1 and 7.]

Zeng, S.; Zhu, L.; Zhang, X.; He, H.; and Lu, Y., 2025a. Supercl: Superpixel guided contrastive learning for medical image segmentation pre-training. *arXiv preprint arXiv:2504.14737*, (2025). https://arxiv.org/abs/2504.14737. [Cited on pages 31, 33, 45, 69, 70, and 81.]

Zeng, S.; Zhu, L.; Zhang, X.; He, H.; and Lu, Y., 2025b. Supercl: Superpixel guided contrastive learning for medical image segmentation pre-training. https://arxiv.org/abs/2504.14737. [Cited on page 34.]

Zeng, S.; Zhu, L.; Zhang, X.; Nnamdi, M. C.; Shi, W.; Tamo, J. B.; Chen, Q.; He, H.; Jin, L.; Tian, Z.; Ren, Q.; Xie, Z.; and Lu, Y., 2025c. Multi-level asymmetric contrastive learning for volumetric medical image segmentation pre-training. https://arxiv.org/abs/2309.11876. [Cited on page 34.]

Zhang, S.; Zhang, J.; and Xia, Y., 2022. Transws: Transformer-based weakly supervised histology image segmentation. In *Machine Learning in Medical Imaging*, 367–376. Springer Nature Switzerland, Cham. [Cited on pages 83, 84, and 85.]

Zhang, S.; Zhang, J.; Xie, Y.; and Xia, Y., 2023. Tpro: Text-prompting-based weakly supervised histopathology tissue segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, 109–118. Springer Nature Switzerland, Cham. [Cited on pages 39, 55, 77, 83, 84, and 85.]

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A., 2015. Learning deep features for discriminative localization. https://arxiv.org/abs/1512.04150. [Cited on page 36.]

Zhou, T.; Xia, W.; Zhang, F.; Chang, B.; Wang, W.; Yuan, Y.; Konukoglu, E.; and Cremers, D., 2024. Image segmentation in foundation model era: A survey. https://arxiv.org/abs/2408.12957. [Cited on page 36.]