# Training Swin-Unet Network Using Bidirectional Copy-Paste Method

Peiyu Hu
University of Electronic Science and Technology of China
Chengdu, China

**Abstract**—Medical image segmentation is of great importance in clinical diagnosis, treatment planning, and disease monitoring. However, obtaining high-quality annotated medical image data is costly and time-consuming, making semi-supervised learning an effective solution to this problem. In recent years, the Bidirectional Copy-Paste (BCP) method [1] has shown excellent performance in semi-supervised learning for medical image segmentation tasks. Attention mechanisms are a powerful technique that can improve model segmentation accuracy by adaptively focusing on important parts of an image. However, applying attention mechanisms in the computer vision (CV) field faces challenges, mainly due to the large amount of data and computational resources required for training and the high demand for labeled data. In this study, I trained the Swin-Unet [2] network on the ACDC dataset based on the BCP method. Swin-Unet combines the attention mechanism of the Swin Transformer [3] with the segmentation architecture of Unet, aiming to improve the segmentation performance of cardiac MRI images. By using pretrained models, various dropout techniques, and weight decay, I significantly enhanced the model's performance. Experimental results show that my method achieves excellent results across multiple evaluation metrics, demonstrating the effectiveness of applying pure attention networks in semi-supervised learning. This research not only advances the development of medical image segmentation technology but also provides beginners with an opportunity to quickly grasp the basic concepts and training techniques in the relevant field.

**Index Terms**—Compueter Vision, Deep Learning, Medical Image Segmentation, Semi-Supervised Learning

## 1 Introduction

MEDICAL image segmentation is a critical task in clinical diagnosis, treatment planning, and disease monitoring. However, several challenges hinder the development and deployment of effective segmentation models:

1) **Scarcity of Annotated Data**: The annotation process for medical images requires expert knowledge, making it time-consuming and expensive.Obtaining a large amount of high-quality annotated data is challenging, which limits the application of fully supervised learning methods.
2) **Data Imbalance**: Medical image datasets often suffer from class imbalance, where certain lesions or tissue types are underrepresented. This imbalance can cause models to bias towards majority classes during training, negatively impacting their performance on minority classes.
3) **Complexity and Diversity**: The anatomical structures and pathological features in medical images are complex and diverse, with significant variations between patients. This complexity requires models to have high robustness and generalization capabilities to accurately segment different types of tissues and abnormalities.

Recent advancements in semi-supervised learning have shown promise in addressing these challenges.

To overcome the scarcity of annotated data, semi-supervised learning methods leverage a small amount of labeled data along with a larger set of unlabeled data to improve model performance. These methods include techniques such as consistency regularization, pseudo-labeling, and self-training.

The BCP method, a subset of semi-supervised learning, enhances labeled datasets by incorporating information from unlabeled data through bidirectional copy-paste techniques. This augmentation improves data diversity and model accuracy without requiring extensive labeled data.

In addition, attention mechanisms have become a trend in computer vision. They enable models to focus on relevant image regions, enhancing feature capture and performance. These mechanisms integrate global and local context, improving segmentation accuracy. They seamlessly integrate into neural network architectures like CNNs and transformers, making them versatile across tasks and domains. By highlighting important image regions, attention mechanisms aid model interpretation and validation. They also enhance efficiency by directing resources to critical areas, leading to faster inference and reduced computational costs.

In this research, I explore the application of the BCP method to train the Swin-Unet model on the ACDC dataset. Swin-Unet, which integrates the Swin Transformer's attention mechanism with Unet's segmentation architecture, aims to enhance cardiac MRI image segmentation. my findings suggest that employing pure attention networks in semi-supervised learning is a valuable endeavor, providing insights and techniques essential for advancing medical image segmentation.

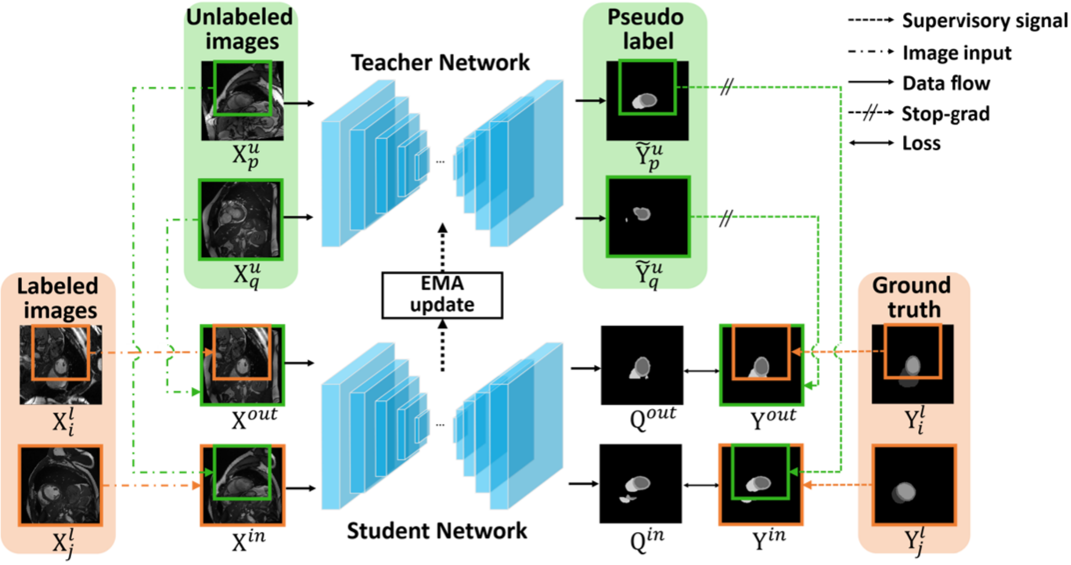

Fig. 1: Inputs to the Student network are derived from the combination of two labeled and two unlabeled images using the proposed bidirectional copy-paste method. The supervisory signal for the Student network is created by merging ground-truths and pseudo-labels generated by the Teacher network, facilitating robust supervision.

# 2 Related Work

## 2.1 Bidirectional Copy-Paste

### *2.1.1 Mean Teacher*

The BCP method is based on the Mean Teacher method, which is a popular approach in semi-supervised learning. It works by maintaining two models: a student model and a teacher model.

The student model is the primary model that is trained directly on the labeled data. It learns to make predictions based on the given input data and is updated using standard backpropagation techniques. The teacher model serves as a guide for the student model. The parameters of the teacher model are not updated through backpropagation. Instead, they are updated as an exponential moving average (EMA) of the student model's parameters. This means that at each training step, the teacher model's parameters are a weighted average of its previous parameters and the current parameters of the student model.

### *2.1.2 BCP Training Steps*

The BCP framework employs two networks: a Teacher network ($F_t$) and a Student network ($F_s$). A diagram illustrating this architecture is provided in Fig. 1. The Student network, parameterized by $\Theta_s$, is optimized using stochastic gradient descent (SGD). Conversely, the Teacher network, parameterized by $\Theta_t$, is updated through the exponential moving average (EMA) of the Student network's parameters, ensuring stability and smoothness in training.

The training strategy is divided into three steps:

1) **Pretraining**: Initially, the model is pretrained using only labeled data.
2) **Pseudo-Label Generation**: The pretrained model is then used as the Teacher network to generate pseudo-labels for the unlabeled data.
3) **Parameter Updates**: In each iteration, the Student network's parameters $\Theta_s$ are optimized using SGD, and subsequently, the Teacher network's parameters $\Theta_t$ are updated using the EMA of the Student parameters $\Theta_s$.

The BCP technique involves copy-pasting between pairs of images to maintain input diversity. A zero-centered mask $M$ is generated to indicate whether a voxel comes from the foreground or background image. This mask helps in creating two new composite images:

$$X_{\text{in}} = X_j^l \odot M + X_p^u \odot (1 - M) \tag{1}$$

$$X_{\text{out}} = X_q^u \odot M + X_i^l \odot (1 - M) \tag{2}$$

Here, $X_i^l$ and $X_j^l$ are labeled images, while $X_p^u$ and $X_q^u$ are unlabeled images. The element-wise multiplication ($\odot$) ensures that different regions of the images are combined, preserving input diversity.

To train the Student network, supervisory signals are generated using the BCP operation. The Teacher network generates probability maps $P_p^u$ and $P_q^u$ for the unlabeled images $X_p^u$ and $X_q^u$. Initial pseudo-labels are derived from these probability maps, which are further refined by selecting the largest connected component to remove outliers.

The pseudo-labels and ground truth labels are bidirectionally copy-pasted to generate supervisory signals:

$$Y_{\text{in}} = Y_j^l \odot M + Y_p^u \odot (1 - M) \tag{3}$$

$$Y_{\text{out}} = Y_q^u \odot M + Y_i^l \odot (1 - M) \tag{4}$$

These signals are used to supervise the Student network's predictions on $X_{\text{in}}$ and $X_{\text{out}}$.

Each input image comprises both labeled and unlabeled components. The loss functions for $X_{\text{in}}$ and $X_{\text{out}}$ are:

$$L_{\text{in}} = L_{\text{seg}}(Q_{\text{in}}, Y_{\text{in}} \odot M) + \alpha L_{\text{seg}}(Q_{\text{in}}, Y_{\text{in}} \odot (1 - M)) \tag{5}$$

$$L_{\text{out}} = L_{\text{seg}}(Q_{\text{out}}, Y_{\text{out}} \odot (1-M)) + \alpha L_{\text{seg}}(Q_{\text{out}}, Y_{\text{out}} \odot M) \quad (6)$$

where $L_{\text{seg}}$ is a combination of Dice loss and Cross-Entropy loss, and $\alpha$ controls the contribution of unlabeled image pixels.

The overall loss $L_{\text{all}} = L_{\text{in}} + L_{\text{out}}$ is used to update the Student network parameters via SGD. The Teacher network parameters are then updated using the EMA of the Student parameters.

#### *2.1.3 Advantages*

The BCP method significantly enhances model robustness and performance through bidirectional data augmentation, especially in scenarios with limited labeled data. This approach effectively bridges the gap between labeled and unlabeled data distributions, promoting better generalization and model adaptability.

## 2.2 Swin-Unet

### *2.2.1 Swin Transformer*

When applying Transformer to the field of image processing, there are two major challenges related to the attention mechanism in computer vision (CV):

- **Scale Variation**: Visual elements in images vary significantly in scale, unlike word tokens in text, making fixed-size token processing unsuitable for vision tasks.
- **High Resolution**: Images have much higher resolution than text, leading to high computational complexity for dense predictions when using traditional Transformer models, which scale quadratically with the number of tokens.

The Swin Transformer is a novel vision Transformer designed to address specific challenges in adapting Transformer models from natural language processing to computer vision tasks.

The important structures of Swin Transformer are as follows:

1) **Shifted Windows**: Instead of global self-attention, Swin Transformer computes self-attention within non-overlapping local windows, which reduces computational complexity to linear with respect to image size. Shifted windows between consecutive layers ensure cross-window connections, enhancing modeling power.
2) **Patch Partitioning and Embedding**: The input image is split into non-overlapping patches, which are then linearly embedded into a higher-dimensional space to form the input tokens.
3) **Patch Merging and Downsampling**: Subsequent stages in the model involve patch merging layers that reduce the number of tokens and downsample the resolution, maintaining computational efficiency while increasing the feature dimensionality.

The structure diagram of a pair of Swin Transformer Block is shown in Fig. 2. The computation process of a pair of Swin Transformer Block is as follows:

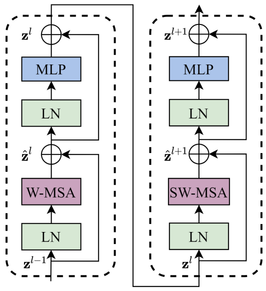

Fig. 2: Two Successive Swin Transformer Blocks

1) **Window-based Multi-head Self-Attention (W-MSA):**

$$\hat{z}_l = \text{W-MSA}(\text{LN}(z_{l-1})) + z_{l-1}$$

2) **Multi-layer Perceptron (MLP):**

$$z_l = \text{MLP}(\text{LN}(\hat{z}_l)) + \hat{z}_l$$

3) **Shifted Window-based Multi-head Self-Attention (SW-MSA):**

$$\hat{z}_{l+1} = \text{SW-MSA}(\text{LN}(z_l)) + z_l$$

4) **Multi-layer Perceptron (MLP):**

$$z_{l+1} = \text{MLP}(\text{LN}(\hat{z}_{l+1})) + \hat{z}_{l+1}$$

where $\hat{z}_l$ and $z_l$ denote the output features of the (S)WMSA module and the MLP module for block $l$, respectively; $W\text{-}MSA$ and $SW\text{-}MSA$ denote window-based multi-head self-attention using regular and shifted window partitioning configurations, respectively.

Standard global self-attention computes relationships between all token pairs, leading to quadratic complexity , shown in Eq. 7. In contrast, Swin Transformer computes self-attention within local windows, reducing complexity to linear, shown in Eq. 8.

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C \quad (7)$$

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC \quad (8)$$

Where $h$ and $w$ represent the height and width of the input image, respectively, while $C$ denotes the dimensionality of the token embeddings. $M$ corresponds to the size of the local window used in Swin Transformer's self-attention mechanism.

### *2.2.2 Swin-Unet Structure*

The overall architecture of the Swin-Unet is presented in Fig. 3.

In the encoder phase, the input medical images are divided into non-overlapping patches of size $4 \times 4$. Each patch, with a feature dimension of $4 \times 4 \times 3$ (representing the image's RGB channels), undergoes transformation through a linear embedding layer into an arbitrary dimension denoted as $C$. These transformed patch tokens then pass through multiple Swin Transformer blocks and patch merging layers to generate hierarchical feature representations. The patch merging layer facilitates downsampling and dimensionality increase, while the Swin Transformer block focuses on learning feature representations.

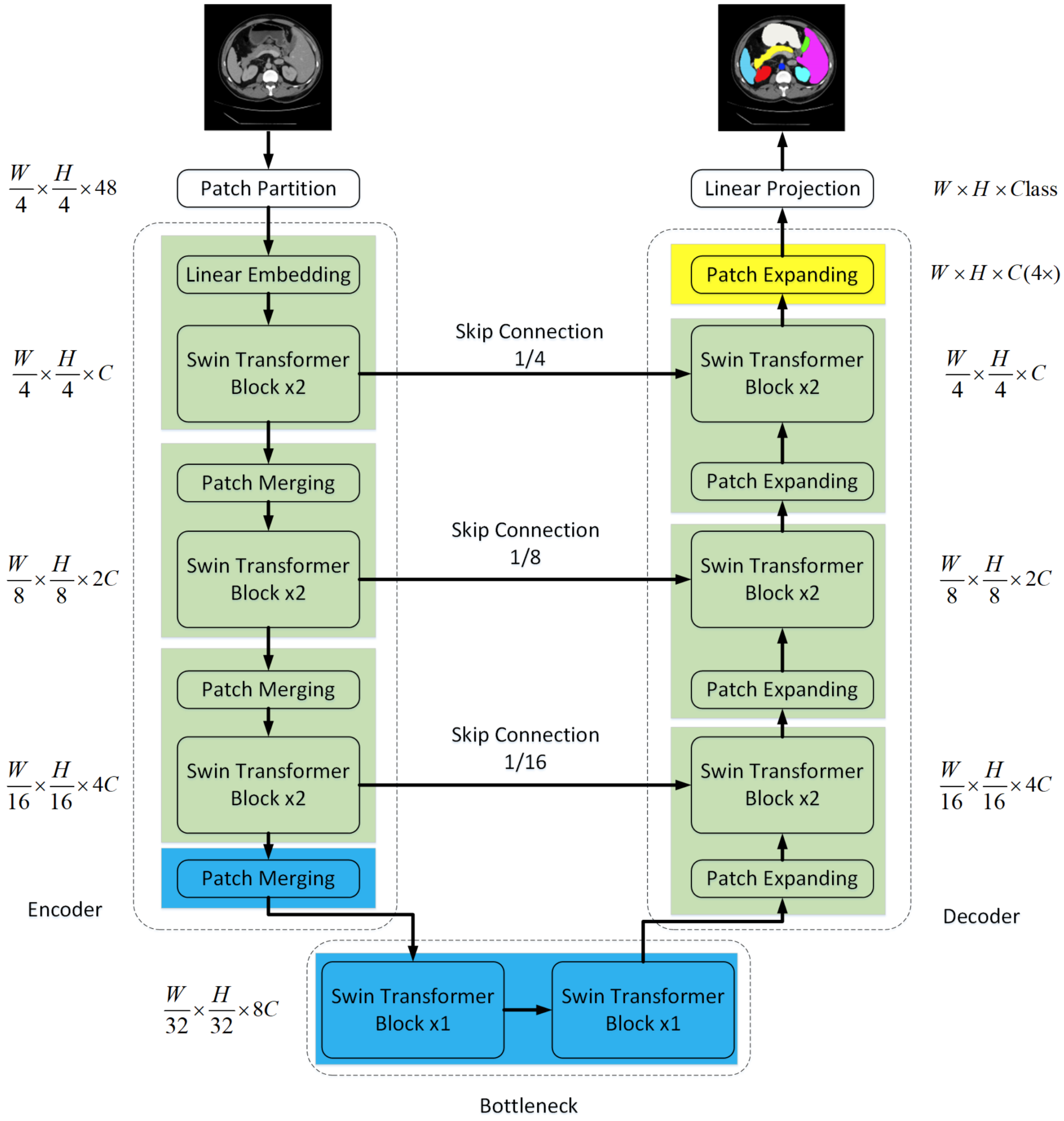
$\frac{W}{4}\times\frac{H}{4}\times 48$
Patch Partition
Linear Embedding
Swin Transformer Block x2
$\frac{W}{4}\times\frac{H}{4}\times C$
Patch Merging
Swin Transformer Block x2
$\frac{W}{8}\times\frac{H}{8}\times 2C$
Patch Merging
Swin Transformer Block x2
$\frac{W}{16}\times\frac{H}{16}\times 4C$
Patch Merging
Encoder
Skip Connection 1/4
Skip Connection 1/8
Skip Connection 1/16
Linear Projection
$W\times H\times \text{Class}$
Patch Expanding
$W\times H\times C(4\times)$
Swin Transformer Block x2
$\frac{W}{4}\times\frac{H}{4}\times C$
Patch Expanding
Swin Transformer Block x2
$\frac{W}{8}\times\frac{H}{8}\times 2C$
Patch Expanding
Swin Transformer Block x2
$\frac{W}{16}\times\frac{H}{16}\times 4C$
Patch Expanding
Decoder
$\frac{W}{32}\times\frac{H}{32}\times 8C$
Swin Transformer Block x1
Swin Transformer Block x1
Bottleneck

Fig. 3: The architecture of Swin-Unet, which is composed of encoder, bottleneck, decoder and skip connections. Encoder, bottleneck and decoder are all constructed based on swin transformer block.

The decoder, inspired by Unet, is symmetrically designed with Swin Transformer blocks and patch expanding layers. Context features extracted from the encoder are fused with multiscale features through skip connections to counteract the loss of spatial information due to downsampling. Unlike the patch merging layer, a patch expanding layer is dedicated to upsampling. It reshapes feature maps of adjacent dimensions into larger feature maps with 2× upsampling in resolution. Finally, the last patch expanding layer performs 4× upsampling to restore the feature maps' resolution to the input level ($W \times H$), followed by a linear projection layer to produce pixel-level segmentation predictions.

## 3 The Proposed Method

To overcome the limitations of traditional CNNs and to facilitate effective training of attention networks, I utilized the BCP method to train the Swin-Unet network.

Traditional CNNs face constraints such as limited receptive fields, spatial invariance, vanishing gradients, and semantic gap issues. Additionally, attention mechanisms can address these limitations by selectively focusing on relevant image regions. Conversely, attention networks encounter challenges related to computational complexity, interpretability, training instability, and potential overfitting. Therefore, my choice to employ the BCP method aimed at mitigating these limitations and improving the overall performance and robustness of the Swin-Unet model in medical image segmentation tasks.

## 4 Experiment

### 4.1 Dataset

For my experiments, I used the ACDC (Automated Cardiac Diagnosis Challenge) dataset, which is an open dataset released for the ACDC competition. This dataset contains 100

MRI images, and expert annotations for the right ventricle, myocardium, and left ventricle structures.

In this experiment, I divided the dataset into 70 images for training, 20 images for testing, and 10 images for validation. Each 2D image has a size of 224×224 pixels. To facilitate training, the 100 3D images have been resliced into 1700 2D images.

## 4.2 Performance Measures

### 4.2.1 DSC

The Dice Similarity Coefficient (DSC), also known as the Dice score or Dice index, is a statistical tool used to gauge the similarity between two sets of data. In the context of image segmentation, DSC is commonly employed to evaluate the performance of segmentation algorithms by comparing the overlap between the predicted segmentation and the ground truth. The Dice coefficient is defined as:

$$DSC = \frac{2|A \cap B|}{|A| + |B|} \tag{9}$$

where A is the set of pixels in the ground truth segmentation, B is the set of pixels in the predicted segmentation, $A \cap B$ is the number of pixels common to both sets, $|A|$ and $|B|$ are the number of pixels in each set, respectively.

### 4.2.2 Jaccard

The Jaccard Index, also known as the Intersection over Union (IoU), is a commonly used metric in image segmentation to evaluate the accuracy of a predicted segmentation against the ground truth. It measures the similarity and diversity of sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets. The Jaccard Index $J$ is given by:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

where A is the set of pixels in the ground truth segmentation, B is the set of pixels in the predicted segmentation, $A \cap B$ is the number of pixels common to both sets, $|A \cup B|$ is the number of pixels in either set (i.e., the union of both sets).

### 4.2.3 95HD

The 95th Percentile Hausdorff Distance (95HD) is a metric used to measure the similarity between two sets of points, which is particularly useful in evaluating image segmentation algorithms. It is an adaptation of the Hausdorff Distance (HD) that provides robustness to outliers by focusing on the 95th percentile of the distances rather than the maximum distance. The Hausdorff Distance $H(A, B)$ is defined as:

$$H(A, B) = \max\left(\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(b, a)\right) \tag{11}$$

where $d(a, b)$ is the Euclidean distance between points $a$ and $b$, sup denotes the supremum (or least upper bound), inf denotes the infimum (or greatest lower bound).

The 95th Percentile Hausdorff Distance modifies this definition to focus on the 95th percentile of the distances rather than the maximum distance:

$$H_{95}(A, B) = \max\left(\mathrm{HD}_{95}(A, B), \mathrm{HD}_{95}(B, A)\right) \tag{12}$$

where $\mathrm{HD}_{95}(A, B)$ is the 95th percentile of the distances from each point in $A$ to the closest point in $B$, and vice versa for $\mathrm{HD}_{95}(B, A)$.

### 4.2.4 ASD

The Average Surface Distance (ASD) is a commonly used metric in image segmentation to evaluate the accuracy of predicted segmentations by comparing them to the ground truth. ASD provides a measure of how close the predicted boundary is to the ground truth boundary by calculating the average distance between the surfaces of the two segmentations.

## 4.3 Experiment Process

### 4.3.1 Introducing Pre-trained Model

At the outset, with a learning rate of 0.01 and labeled samples comprising 10% of the total dataset, the results for fmy metrics were in Table 1:

| Metrics | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| Initial Results | 68.19 | 54.72 | 15.35 | 5.23 |

Table 1

The performance is very poor, mainly due to the complexity of the model and the limited size of the dataset, making it difficult for the model to receive effective training.

Afterwards, I decided to initialize the model parameters with a pre-trained model, Swin-T, which was pre-trained on ImageNet as Swin Transformer. The training results were as expected, and the performance showed a significant improvement in Table 2

| Metrics | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| After Loading Swin-T | 84.83 | 74.62 | 4.42 | 1.36 |

Table 2

Fig. 4 and Fig. 5 display the training loss curve and the validation Dice performance curve, respectively.

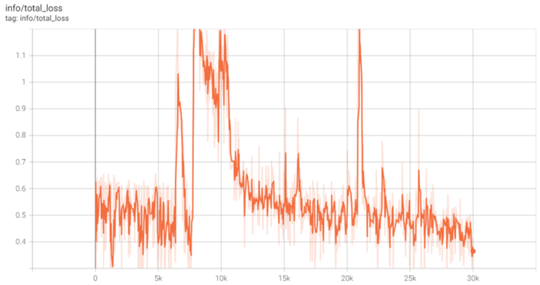

Fig. 4: Training Loss

These curves show that during the training process, the model's parameter updates are not stable, resulting in significant fluctuations.

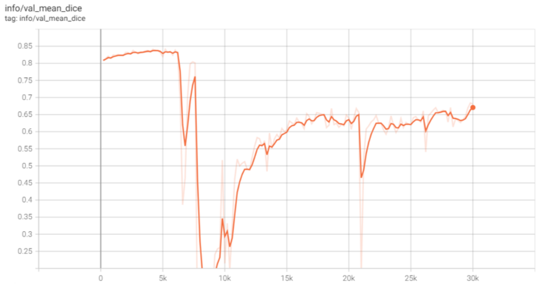

Fig. 5: Validation Dice Performance

| Metrics | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| After Grad Clipping | 86.1 | 76.39 | 4.17 | 1.12 |

Table 3

### 4.3.2 Grad Clipping

I choose to implement gradient clipping, which helps mitigate the problem of exploding gradients during training by imposing a threshold on the gradients.The performance after training is as follows:

The performance of the model has been further improved. However, there are still some issues during training. Fig. 6 and Fig. 7 show training loss curve and the Dice curve on the validation set for this training, respectively.

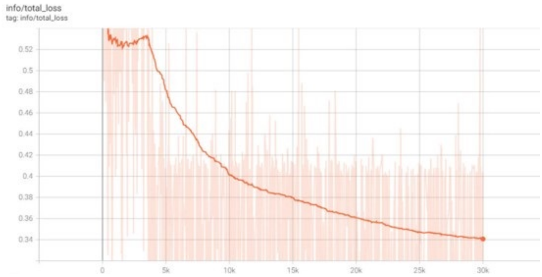

Fig. 6: Training Loss

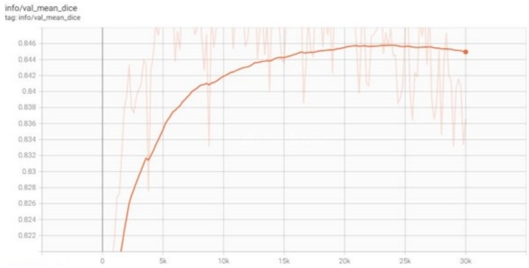

Fig. 7: Validation Dice Performance

Fig. 6 and Fig. 7 indicate that the model's performance continues to improve on the training set, but on the validation set, the performance of the model starts to decline after 20k epochs, suggesting the occurrence of overfitting.

### 4.3.3 Attention Dropout

Subsequently, I conducted numerous experiments and employed various methods in an attempt to address the issues present during training. However, none of these approaches yielded satisfactory results. Upon reviewing the code, I identified a critical issue related to parameter updates. Specifically, the student model was being updated using stochastic gradient descent (SGD), but during gradient computation, the parameters of the teacher model were also involved. This contradicted the original intention of the BCP method.

I corrected the code and further utilized dropout to mitigate the issue of overfitting.

The dropout strategies employed in the Swin-Unet model include:

1) Applying dropout to the MLP within the FFN section.
2) Applying dropout after computing the similarity between the query and key during the attention calculation.
3) Applying dropout on the main path prior to adding it with the residual connection.

In the Swin-Unet model, the default setting for the Path Drop Rate is 0.1. Given that dropout in the MLP section is expected to significantly impact the training effectiveness of the model, I conducted multiple experiments focusing specifically on dropout related to attention mechanisms. The experimental results are presented in Table 4, where 'ad' refers to 'attn_drop_rate', and 'pd' refers to 'path_drop_rate'.

| Hyperparameter | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| **0.3ad+0.1pd** | 87.2 | 78.12 | 3.12 | 1.02 |
| **0.4ad+0.1pd** | 86.6 | 77.19 | 2.65 | 0.85 |
| **0.2ad+0.1pd** | 86.14 | 76.52 | 4.12 | 1.42 |
| **0.1pd** | 85.83 | 76.04 | 3.58 | 1.06 |

Table 4. Attention Dropout Results

In my experiments, the optimal performance was achieved when the Attention Drop Rate was set to 0.3. The corresponding curves from this experiment are presented in Fig. 8 and Fig. 9.

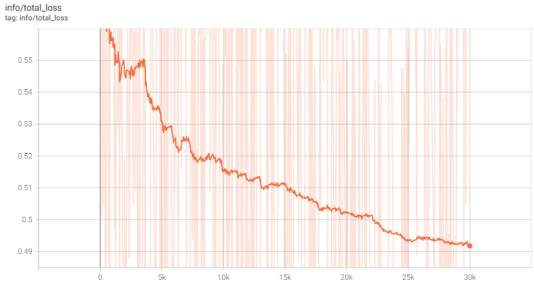

Fig. 8: Training Loss

According to Fig. 8 and Fig. 9, it is evident that during the final 5k training epochs, the model exhibited minimal improvement on the training set and even demonstrated a decline in performance on the validation set. This phenomenon can be attributed to unstable parameter updates and oscillating loss values.

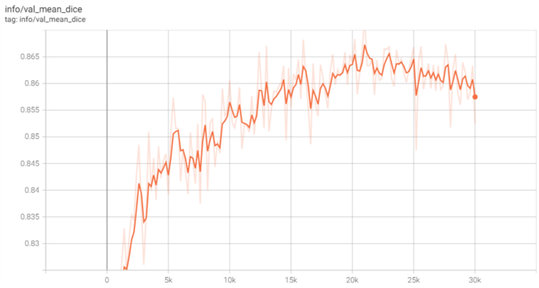

Fig. 9: Validation Dice Performance

### 4.3.4 More Hyperparameter Optimization

To mitigate the issue of oscillating loss values observed during training, I first considered reducing the learning rate. By lowering the learning rate, I aim to achieve more stable parameter updates and smoother convergence, ultimately leading to improved model performance. The experimental results of adjusting the learning rate are shown in the Table 5, where 'lr' refers to learning rate.

| Hyperparameter | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| **0.3ad+0.1pd+0.02lr** | 86.49 | 77.03 | 2.63 | 78.14 |
| **0.3ad+0.1pd+0.005lr** | 86.21 | 76.6 | 4.06 | 1.26 |
| **0.3ad+0.1pd+0.007lr** | 86.51 | 77.09 | 3.85 | 1.21 |
| **0.3ad+0.1pd+0.008lr** | 86.77 | 77.44 | 2.6 | 0.84 |
| **0.3ad+0.1pd+0.009lr** | 86.86 | 77.6 | 4.78 | 1.42 |
| **0.3ad+0.1pd+0.0085lr** | 86.47 | 76.99 | 3.88 | 1.15 |
| **0.3ad+0.1pd+0.0087lr** | 86.63 | 77.27 | 2.4 | 0.85 |
| **0.3ad+0.1pd+0.0086lr** | 86.68 | 77.35 | 4.01 | 1.19 |
| **0.3ad+0.1pd+0.015lr** | 85.7 | 75.98 | 4.82 | 1.53 |

Table 5. Results

However, issues of overfitting and underfitting still persist. As shown in the Fig. 10, when the learning rate is set to 0.0085, the model's loss on the training set decreases rapidly between 24k and 30k epochs. However, in Fig. 11, the performance on the validation set does not improve, indicating an overfitting issue.

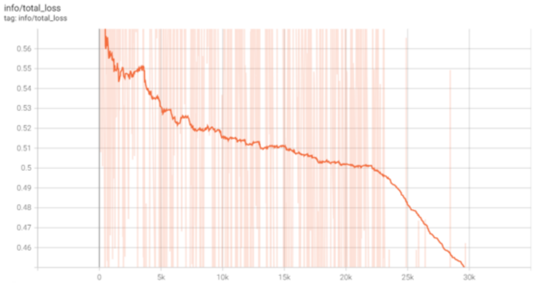

Fig. 10: Training Loss

But when the learning rate is set to 0.0086, as shown in the Fig. 12 and Fig. 13, the loss values once again exhibit oscillations, and there is no improvement in the performance on the validation set.

I experimented with adjusting the dropout probability of attention dropout, reducing its probability to aid in its fitting. However, the model remains quite sensitive and cannot find a balance between overfitting and oscillations.

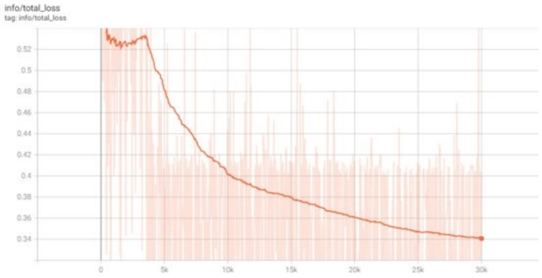

Fig. 11: Validation Dice Performance

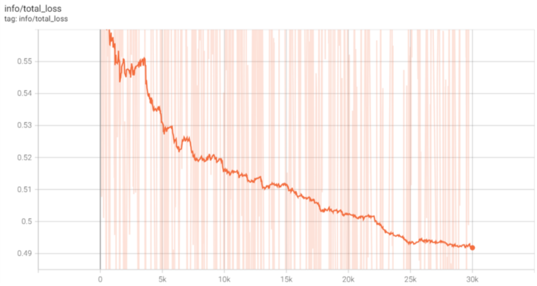

Fig. 12: Training Loss

The experimental results are shown in the following Table 6.

| Hyperparameter | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| **0.15ad+0.1pd** | 85.79 | 76.09 | 3.55 | 1.2 |
| **0.18ad+0.1pd** | 86.31 | 76.85 | 3.15 | 1.07 |
| **0.19ad+0.1pd** | 86.43 | 76.97 | 3.64 | 1.24 |
| **0.195ad+0.1pd** | 86.6 | 77.21 | 3.58 | 0.96 |
| **0.197ad+0.1pd** | 86.4 | 76.92 | 2.63 | 0.87 |
| **0.196ad+0.1pd** | 86.41 | 76.92 | 3.14 | 0.93 |

Table 6. Training Loss curve

I also attempted dropout on the main path and MLP, as well as experimenting with various combinations of hyperparameters. The results are depicted in the following Table 7, where 'd' refers to drop_rate.

Finally, I adjusted the weight decay coefficient and achieved a satisfactory result. The results are as shown in the following Table 8.

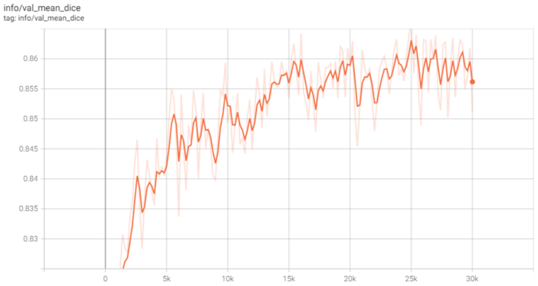

Fig. 13: Validation Dice Performance

| Hyperparameter | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| **0.3ad+0.05pd** | 86.23 | 76.7 | 3.54 | 1.04 |
| **0.3ad+0.08pd** | 85.74 | 76.01 | 3.53 | 1.22 |
| **0.3ad+0.085pd** | 86.29 | 76.78 | 3.46 | 1.16 |
| **0.3ad+0.083pd** | 86.35 | 76.89 | 3.65 | 1.13 |
| **0.3ad+0.084pd** | 86.06 | 76.5 | 4.04 | 1.31 |
| **0.3ad+0.084pd+0.0087lr** | 86.44 | 76.98 | 4.13 | 1.18 |
| **0.3ad+0.085pd+0.0087l** | 86.27 | 76.76 | 3.74 | 1.18 |
| **0.3d+0.1pd** | 85.32 | 75.41 | 3.2 | 1.11 |
| **0.2d+0.1pd** | 83.99 | 73.37 | 6 | 1.48 |
| **0.1d+0.1pd** | 85.6 | 75.87 | 4.43 | 1.4 |

Table 7. Results

| Hyperparameter | Dice (%) | Jaccard (%) | 95HD | ASD |
|---|---|---|---|---|
| **0.3ad+0.1pd+0wd** | 86.29 | 76.88 | 4.83 | 1.42 |
| **0.3ad+0.1pd+0.00001wd** | 86.66 | 77.23 | 3.02 | 0.96 |
| **0.3ad+0.1pd+0.00005wd** | 87.14 | 78.02 | 2.32 | 0.76 |
| **0.3ad+0.1pd+0.00009wd** | 86.36 | 76.77 | 3.44 | 1.08 |
| **0.3ad+0.1pd+0.00007wd** | 87.16 | 77.98 | 2.1 | 0.67 |
| **0.3ad+0.1pd+0.00008wd** | 85.84 | 76.19 | 4.19 | 1.23 |

Table 8. Results

## 5 Results and Analysis

The comparison results of my model and other models are illustrated in the Table 9.

| Method | Dice | Jaccard | 95HD | ASD |
|---|---|---|---|---|
| **UA-MT** | 81.65 | 70.64 | 6.88 | 2.02 |
| **SASSNet** | 84.50 | 74.34 | 5.42 | 1.86 |
| **DTC** | 84.29 | 73.92 | 12.81 | 4.01 |
| **URPC** | 83.10 | 72.41 | 4.84 | 1.53 |
| **MC-Net** | 86.44 | 77.04 | 5.50 | 1.84 |
| **SS-Net** | 86.78 | 77.67 | 6.07 | 1.40 |
| **BCP-UNet** | 88.76 | 80.39 | 3.88 | 1.28 |
| **Ours** | 87.16 | 77.98 | 2.10 | 0.67 |

Table 9. Results

While my model may not achieve the highest scores in all metrics, its superior performance in 95HD and ASD metrics indicates its effectiveness in accurately delineating boundaries and capturing the structural details of the objects of interest. This suggests that my model may be particularly well-suited for applications where precise delineation of boundaries is crucial.

## 6 Conclusion

In conclusion, I utilized the Bidirectional Copy-Paste method, based on Mean Teacher semi-supervised learning, to train the Swin-Unet model on the ACDC dataset. Employing pre-training, various dropout techniques, weight decay, and other training strategies, I improved the model's performance and achieved satisfactory results. I believe that exploring pure attention-based networks in the semi-supervised learning domain is meaningful. Attention-based networks are often deep, and achieving good results with limited labeled data can be challenging. Furthermore, this process requires extensive analysis and experimentation, which is advantageous for beginners like me to quickly grasp the fundamental concepts and training techniques in related fields.

## References

[1] Yunhao Bai, Duowen Chen, Qingli Li, Wei Shen, and Yan Wang. Bidirectional copy-paste for semi-supervised medical image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11514–11524, 2023.

[2] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022.

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