Advancing Medical Image Segmentation via Self-supervised Instance-adaptive Prototype Learning

Guoyan Liang¹, Qin Zhou², Jingyuan Chen¹, Zhe Wang²* and Chang Yao¹*

¹Zhejiang University, Hangzhou, China
²East China University of Science and Technology, Shanghai, China
{guoyanl, jingyuanchen, changy}@zju.edu.cn, {sunniezq, wangzhe}@ecust.edu.cn

Abstract

Medical Image Segmentation (MIS) plays a crucial role in medical therapy planning and robot navigation. Prototype learning methods in MIS focus on generating segmentation masks through pixel-to-prototype comparison. However, current approaches often overlook sample diversity by using a fixed prototype per semantic class and neglect intra-class variation within each input. In this paper, we propose to generate instance-adaptive prototypes for MIS, which integrates a common prototype proposal (CPP) capturing common visual patterns and an instance-specific prototype proposal (IPP) tailored to each input. To further account for the intra-class variation, we propose to guide the IPP generation by re-weighting the intermediate feature map according to their confidence scores. These confidence scores are hierarchically generated using a transformer decoder. Additionally we introduce a novel self-supervised filtering strategy to prioritize the foreground pixels during the training of the transformer decoder. Extensive experiments demonstrate favorable performance of our method.

1 Introduction

Medical image segmentation (MIS) aims to divide an image into meaningful parts, representing different semantic classes. It serves as a fundamental requirement for various tasks such as cancer diagnosis, radiation therapy dosage control and surgical robot navigation. Recognizing the significant resemblance between MIS and the task of clustering pixels into semantic segments, recent researchers have focused on prototype-based methods. By learning representative prototype features, which serve as cluster centers for each semantic class, the MIS objective is reformulated as pixel-to-prototype matching. Existing prototype-based MIS methods primarily concentrate on learning representative prototypes, which remain fixed upon completion of the training process [Zhou et al., 2022]. In order to capture the intra-class variation within each semantic class, prior studies have explored the use of local prototypes [Yu et al., 2021; Wang et al., 2022]. However, these prototypes are predominantly generated based on spatial grids or pre-segmentation techniques such as superpixels. Recently, the introduction of mask transformers [Cheng et al., 2022; Yu et al., 2022; Yuan et al., 2023] has allowed for the direct learning of object queries using a transformer decoder. By considering these object queries as cluster centers, we can obtain the segmentation mask by assigning pixels to clusters and mapping clusters to their corresponding semantic classes. However, as mentioned in [Cheng et al., 2022; Yu et al., 2022; Yuan et al., 2023], without proper guidance, existing mask transformers may introduce a bias towards focusing on the background.

To tackle the aforementioned challenges in prototype-based segmentation, we introduce a novel framework called Self-supervised Instance-adaptive Prototype Learning (referred as SIPL) for MIS. Specifically, we propose an Instance-adaptive Prototype Learning (referred as IPL) scheme that adaptively learns prototypes tailored to different inputs. In IPL, the final class-specific prototypes are generated using a combination of common prototype proposals (CPPs), which capture the common visual patterns of each class, and instance-specific prototype proposals (IPPs), which dynamically emphasize the instance-specific visual characteristics of the current input.

To further account for the intra-class variation during prototype generation, we propose a novel hierarchical Self-supervised Mask Generation (referred as SMG) module to hierarchically generate pseudo masks with confidence scores to guide the generation of IPPs. These confidence scores are used to re-weight the intermediate feature map, indicating the pixel-wise importance in contributing to the IPPs for each class. In SMG, a transformer decoder is employed to generate query embeddings from the multi-scale features obtained from the pixel decoder. To mitigate background bias in the SMG module, we incorporate a self-supervised filtering strategy that prioritizes foreground pixels during the learning of query embeddings. To improve the accuracy of the confidence scores, we further introduce an auxiliary loss to supervise the training process of the transformer decoder.

The overall contributions of our method can be summarized as follows.
We propose a novel framework to learn instance-adaptive prototypes for MIS, which incorporates common and instance-specific prototype proposals to capture the common and instance-specific visual patterns for each class.

We propose a self-supervised mask generation module to guide instance-specific prototype proposal generation by re-weighting intermediate features, where we incorporate a novel self-supervised filtering strategy to prioritize foreground pixels during pseudo mask generation.

Extensive experiments across three challenging MIS tasks demonstrate favorable performance of our method.

2 Related Works

2.1 Medical Image Segmentation

Early methods in medical image segmentation (MIS) primarily rely on contour-based and traditional machine learning algorithms [Tsai et al., 2003]. In contrast, contemporary MIS approaches predominantly leverage deep CNNs and transformer models. U-Net initially introduced for MIS based on CNN, stands out for its simplicity and superior performance [Ronneberger et al., 2015]. Subsequently, various U-Net variants have emerged for 2D MIS such as ResUNet [Xiao et al., 2018], U-Net++ [Zhou et al., 2018], and UNet+ [Huang et al., 2020]. Transformer-based methods have emerged due to the impressive speed-accuracy performance of the Vision Transformer (ViT) in various visual tasks [Dosovitskiy et al., 2020]. Swin Transformer has notably excelled by introducing an efficient and effective hierarchical vision transformer that leverages the shifted windows mechanism [Liu et al., 2021]. Building upon this paradigm, Swin-unet incorporates a U-shaped Encoder-Decoder architecture with skip connections, further advancing the field [Cao et al., 2022]. Segtran proposes a novel positional encoding scheme for transformers, which imposes a continuity inductive bias for images and achieves high segmentation accuracy with good cross-domain generalization capabilities [Li et al., 2021]. Some recent works explore hybrid approaches that demonstrate improved segmentation performance compared to pure CNNs and transformers-based counterparts. UNETR seamlessly integrates the long-range spatial dependencies of transformers with the inductive bias of CNNs within a “U-shaped” encoder-decoder architecture [Hatamizadeh et al., 2022]. nnFormer strategically combines CNN and transformer-based blocks, interleaving them to harness the complementary strengths of each [Zhou et al., 2021]. UNETR++ introduces an innovative Efficient Paired Attention (EPA) block, skillfully capturing enriched interdependent spatial and channel features by employing both spatial and channel attention in two distinct branches [Abdelrahman Shaker et al., 2022].

2.2 Prototype Based Methods

Prototype learning assigns query embeddings to class centers in the network’s output space, typically defined as the mean vector of the training set for each class [Mensink et al., 2013]. Prototype-based networks exhibit a straightforward output layout [Wen et al., 2016] and demonstrate fast generalization to new classes. As a result, they often concentrate on the few-shot regime, [Boney and Ilin, 2017; Snell et al., 2017] or update the prototypes online based on mini-batches [Guerrero et al., 2018]. Matching Network is proposed to leverage learned prototypes for one-shot learning, capturing class features during training and demonstrating robust classification during testing [Vinyals et al., 2016]. Prototypical Network utilizes prototype spaces to compute distances between input samples and class prototypes for few-shot learning, proving effective in scenarios with limited labeled data. [Santoro et al., 2016] incorporates memory augmentation to store task-specific prototype information, facilitating rapid adaptation to new tasks. Relation Networks [Sung et al., 2018] emphasizes modeling relationships between input samples and class prototypes for enhanced few-shot learning performance. Additionally, [Chen et al., 2019] refines prototype-based methods and explore improved strategies to leverage limited training data for more accurate classification. While existing prototype-based methods have made significant progress, they often lack the flexibility to dynamically adapt to different inputs [Zhou et al., 2022]. In contrast, our method focuses on learning instance-adaptive prototypes, resulting in improved performance in MIS.

3 Method

In this section, we will provide an overview of our proposed method. We start by discussing the background and limitations of existing prototype-based segmentation methods in Sec. 3.1. Then, we present the Self-supervised Instance-adaptive Prototype Learning (SIPL) framework in Sec. 3.2. We further explain the Instance-adaptive Prototype Learning (IPL) strategy in Sec. 3.3 and the hierarchical Self-supervised Mask Generation (SMG) module in Sec. 3.4.

3.1 Revisiting Prototype Based Segmentation

Prototype learning methods in Medical Image Segmentation (MIS) focus on generating segmentation masks through pixel-to-prototype comparison. These approaches typically learn one prototype feature for each class. Denote the learned prototypes as \( G = \{g_k \in \mathbb{R}^d\}, k \in \{1, \ldots, K\} \), the \( i \)-th pixel embedding as \( f_i \in \mathbb{R}^d \), where \( K \) is the number of classes and \( d \) is the feature dimension. Then the probability that the \( i \)-th pixel belongs to the \( k \)-th class can be calculated as follows:

\[
p(k|f_i) = \frac{\exp(-< f_i, g_k >)}{\sum_{k'=1}^{K} \exp(-< f_i, g_{k'} >)},
\]

where \( <, > \) denotes the cosine similarity between two items. The existing prototype-based segmentation methods encounter challenges in two aspects: 1) Fixed prototypes during inference that lack dynamic adaptation to different inputs. 2) Prototypes generated by averaging feature embeddings of each class, overlooking intra-class variation. To overcome these limitations, we propose a novel framework that generates instance-adaptive prototypes for each input. In addition, we address intra-class variation by hierarchically generating pseudo masks with confidence scores, which serve as the pixel-wise importance in the instance-specific prototype
3.2 Overall Architecture

Figure 1 illustrates the overall architecture of our method. Given the input 3D volume-mask pair \((V, Y)\), where \(V \in \mathbb{R}^{H \times W \times Z}\) is the 3D image patch, and \(Y \in \mathbb{R}^{H \times W \times Z}\) refers to the corresponding segmentation mask. We utilize a 3D backbone comprising an encoder branch \(\psi_e(\cdot)\) and a pixel decoder branch \(\psi_d(\cdot)\) for segmentation mask prediction. Concretely, the input \(V\) is first fed into the encoder to obtain the intermediate feature map \(F \in \mathbb{R}^{S \times H \times W \times Z 	imes d_i}\), where \(S\) is the stride of the feature map, \(d_i\) is the feature dimension of the intermediate feature map. Then \(F\) is fed into the pixel decoder branch \(\psi_d(\cdot)\) to generate the output per-pixel embeddings \(F_o \in \mathbb{R}^{H \times W \times Z \times d}\). The final segmentation mask is predicted as the pixel-to-prototype comparison between \(F_o\) and the learned prototypes \(G\).

To facilitate dynamic adaptation of learned prototypes to current input, we introduce the Instance-adaptive Prototype Learning (IPL) strategy, which combines Common Prototype Proposals (CPPs) and Instance-specific Prototype Proposals (IPPs) generation. The CPPs are learnable parameters obtained from the entire dataset, while IPPs are generated by reweighting the intermediate feature map \(F\) using the pixel-wise confidence scores. The final prototype proposals \(P = \{p_k\}, k \in \{1, \ldots, K\}\) are obtained by concatenating the CPPs and IPPs along the feature channel dimension. These proposals are then processed through a few fully connected layers (MLP) to obtain the final prototypes \(G\).

In order to account for the intra-class variation during the generation of IPPs, we introduce the Self-supervised Mask Generation (SMG) module to generate pixel-wise confidence scores for each class. This module utilizes a mask transformer to generate pseudo masks with confidence scores, where the pseudo masks are hierarchically aggregated from pixel-to-cluster and cluster-to-class assignments. To provide the IPL module with more accurate pseudo masks, we introduce the auxiliary loss \((L_{aux})\) between pseudo masks \(M\) and ground truth masks \(Y\).

3.3 Instance-adaptive Prototype Learning

The overall workflow of IPL is illustrated in Figure 2. Formally, given the intermediate feature map \(F \in \mathbb{R}^{S \times H \times W \times Z \times d_i}\) and pseudo masks from the final transformer decoder layer \(M = \{M_k \in \mathbb{R}^{H \times W \times Z \times 1}\}, k \in \{1, 2, \ldots, K + 1\}\), where \(M_k \in [0, 1]\) denotes the predicted confidence scores of the \(k\)-th class. Then the instance-specific prototype proposal (IPP) for the \(k\)-th class \(p_k^i \in \mathbb{R}^{1 \times d_i}\) is generated as,

\[
p_k^i = GAP(M_k \otimes F),
\]

where \(\otimes\) is the pixel-wise multiplication, \(GAP(\cdot)\) denotes global average pooling. In our method, pseudo masks \(M\) are generated hierarchically from the self-supervised mask generation module (please refer to Eq. (9) for details). Please note that the pseudo mask \(M_k\) are reshaped to match the size of \(F\) in Eq. (2).

In addition to the IPPs, our method includes a learnable common prototype proposal (CPP) for each class. This shared proposal captures the common visual dynamics of each class present in the entire training set. Concretely, denote the CPPs as \(P^c = \{p_k^c\}, k \in \{1, 2, \ldots, K\}\), where \(p_k^c \in \mathbb{R}^{1 \times d_i}\) denotes the prototype proposal of the \(k\)-th class. These CPPs are optimized by the overall learning objective and are randomly initialized in our approach. The IPPs and CPPs are then fused together to generate the final prototype \(G\). Formally, the final prototype belonging to the \(k\)-th class is obtained as,

\[
g_k = MLP(p_k^i \oplus p_k^c),
\]

where \(\oplus\) denotes the concatenation operation, \(MLP(\cdot)\) refers to a few fully connected layers. Given the output per-pixel embeddings \(F_o \in \mathbb{R}^{H \times W \times Z \times d}\) from the pixel decoder
branch and the learned prototype $g_k \in \mathbb{R}^{d \times 1}$, the final segmentation prediction for the $k$-th class is formulated as,

$$\hat{Y}_k = \sigma(F_o \circledast g_k),$$  \hspace{1cm} (4)$$

where $\circledast$ denotes the matrix multiplication, $\sigma(\cdot)$ denotes the Sigmoid operation. We adopt soft dice loss and binary cross-entropy to calculate the segmentation loss, which is mathematically formulated as,

$$L_{seg} = 1 - \sum_{k=1}^{K} \left( \sum_{i} Y_{k,i} \frac{\hat{Y}_{k,i}^2}{\hat{Y}_{k,i}^2 + Y_{k,i} log \hat{Y}_{k,i}} + \sum_{i} Y_{k,i} log \hat{Y}_{k,i} \right).$$  \hspace{1cm} (5)$$

Here, $Y_{k,i}, \hat{Y}_{k,i}$ correspond to the $i$-th pixel of the segmentation map for class $k$ in the ground truth and predictions, respectively.

### 3.4 Self-supervised Mask Generation

Starting from DETR [Carion et al., 2020] and MaX-DeepLab [Wang et al., 2021a], the object queries, updated by multiple transformer decoders, are employed as mask embedding vectors. In our approach, we employ learnable object queries to capture intra-class variation, as shown in Figure 3. These object queries, which can be seen as clusters, are mapped onto the semantic class to generate pseudo masks with confidence scores. This process facilitates instance-adaptive prototype learning. Mathematically, denote the learnable object queries as $Q \in \mathbb{R}^{N \times d_q}$, where $N$ ($N > K$) refers to the number of object queries, $d_q$ denotes the feature dimension of the object queries. Then $Q$ is updated by KMax decoders [Yu et al., 2022] using the multi-scale features from the pixel decoder. Denote the reshaped feature map from the $l$-th layer of the pixel decoder as $F^l_d \in \mathbb{R}^{H_l \times W_l \times Z_l \times d_q}$, the randomly initialized query embeddings as $Q^l_0$, then the updated query embeddings $Q^l$ is calculated as,

$$A = \arg\max_{N} (Q^{l-1}_d \otimes (F^l_d)^T),$$

$$Q^l = MHSA(Q^{l-1}) + A \otimes F^l_d,$$  \hspace{1cm} (6)$$

where $Q^{l-1}$ denotes the query embeddings obtained from the $l - 1$ layer, $MHSA(\cdot)$ denotes the multi-head self-attention. Based on the updated query embeddings $Q^l$ from layer $l$, we can obtain the corresponding pixel-to-cluster assignment map $M^{l}_{pc}$ as,

$$M^{l}_{pc} = \text{sim}(F^l_d, Q^l),$$  \hspace{1cm} (7)$$

where $\text{sim}(\cdot)$ calculates the feature similarity between each pixel of $F^l_d$ and each query embedding of $Q^l$. The generated pixel-to-cluster assignment map $M^{l}_{pc}$ is of size $H_l \times W_l \times Z_l \times N$. Since the IPL module requires class-specific pseudo masks, we further introduce a cluster-to-class projection head to generate cluster-to-class projection matrix $M^{l}_{cc} \in \mathbb{R}^{N \times (K + 1)}$,

$$M^{l}_{cc} = MLP(Q^l).$$  \hspace{1cm} (8)$$

We then aggregate the pixel-to-cluster assignments $M^{l}_{pc}$ and the cluster-to-class mapping matrix $M^{l}_{cc}$ to obtain the pseudo masks for the $l$-th layer as,

$$M^l = M^{l}_{pc} \odot M^{l}_{cc}.$$  \hspace{1cm} (9)$$

$M^l$ undergoes softmax processing to get the normalized confidence scores. It should be noted that during the generation of $M^l$, the background class (i.e., $k = K + 1$) is considered to obtain reasonable confidence scores using softmax.

**Self-supervised Filtering.** As mentioned in [Yuan et al., 2023], generating object queries without proper guidance can introduce a bias towards focusing on the background. To mitigate this issue and prioritize foreground features during training, we introduce a novel self-supervised filtering strategy. This strategy involves comparing the overlapping ratios between the activated regions of each object cluster and the foreground regions of each class. The comparison is performed between the pseudo masks generated by the previous layer $M^{l-1}$ and the pixel-to-cluster assignments at current layer $M^{l}_{pc}$, which is formulated as,

$$\text{Overlap}(M^{l-1}, M^{l}_{pc}) = \arg\max_{k} \frac{\sum_{n}(M^{l-1}_{k,n} \cap M^{l}_{pc,n})}{\sum_{n} M^{l}_{pc,n}},$$  \hspace{1cm} (10)$$
Figure 4: Qualitative comparison between SIPL and SOTA models on BTCV dataset. Current methods face challenges in accurately segmenting various organs (highlighted within yellow dashed boxes). In contrast, our proposed framework exhibits commendable segmentation performance.

where $M_{pc}^{l,n}$ denotes the binarized mask prediction of the $n$-th cluster in $M_{pc}^{l}$, $M_{k-1}^l$ refers to the binarized mask prediction of the $k$-th semantic class in $M_{k-1}^l$. The numerator of $Overlap(\cdot, \cdot)$ calculates the intersection between the mask of the current cluster and the masks of each semantic class, and $Overlap(\cdot, \cdot)$ returns the maximum ratio between the cluster-to-class intersection and the area of the related cluster. The underlying assumption is that each cluster represents a specific part of a particular semantic class. Therefore, only clusters with an overlapping ratio higher than the threshold $\tau$ are chosen for query embedding update. In our method, the value of $\tau$ is progressively increased from 0.1 to 0.5 in a linear manner and remains constant after epoch 50, as shown below,

$$\tau = \min(\frac{\text{epoch}}{50}, 1.0) \times 0.4 + 0.1, \quad (11)$$

where $\text{epoch}$ denotes the index of current training epoch, $\min(\cdot, \cdot)$ returns the smaller value. In order to enhance the quality of the pseudo masks, we introduce an auxiliary loss,

$$L_{aux}^l = 1 - \sum_{k=1}^{K+1} \sum_{i} \frac{Y_{k,i}M_{k,i}^l}{\sum_{i} Y_{k,i}^2 + M_{k,i}^l} + \sum_{i} Y_{k,i} \log M_{k,i}^l, \quad (12)$$

where $Y_{k,i}$, $M_{k,i}^l$ refer to the results of the $i$-th pixel. The overall objective for training is formally defined as,

$$L = L_{seg} + \alpha \sum_{l} L_{aux}^l, \quad (13)$$

where $\alpha$ is empirically set as 0.05 in our experiments.

4 Experiments & Results

Datasets and Evaluation. We conduct experiments on three challenging datasets, including the BTCV dataset [Landman et al., 2015], the Lungs dataset [Simpson et al., 2019] and the BraTS dataset [Menze et al., 2014]. The BTCV dataset consists of abdominal CT scans from 30 subjects for training and 20 subjects for testing. It includes 13 organs: spleen, right kidney, left kidney, gallbladder, liver, stomach, aorta, pancreas, esophagus, inferior vena cava, portal and splenic veins, right and left adrenal gland, respectively. The Lungs dataset consists of 63 CT volumes for a binary classification task, specifically focused on delineating lung cancer from the background. The dataset is divided into the training and testing set, with a split ratio of 80:20. The BraTS dataset comprises 484 MRI images, each with FLAIR, T1w, T1gd, and T2w channels. The dataset is divided into training, validation, and testing set, with a split ratio of 80:5:15. The target categories for segmentation are whole tumor (WT), enhancing tumor (ET), and tumor core (TC). We utilize the Dice Similarity Coefficient (DSC) as the performance metric for evaluation in all our experiments.

Implementation Details. We run all experiments based on Python 3.9, PyTorch 2.1.1 and Ubuntu 22.04. All training procedures have been performed on a single A800 GPU with 80GB memory. We report results with $96 \times 96 \times 96$ input size and patch resolution of $(4, 4, 4)$ for BTCV dataset.
where the models are trained for 2k epochs with the default initial learning rate of $4e^{-4}$, momentum of 0.9 and decay of $1e^{-5}$. For Lungs and BraTS datasets, consistent with [Abdellahman Shaker et al., 2021], the models are trained with input 3D patches of size $128 \times 128 \times 64$ for 1k epochs and the learning rate is 0.01 and weight decay is $3e^{-5}$. The transformer decoder consists of six kMax decoders evenly distributed across feature map with spatial resolutions of $\frac{1}{2^{3}}, \frac{1}{2^{2}}, \frac{1}{2}$, and $\frac{1}{2}$. Two kMax decoders are deployed for each resolution. Additionally, the number of object queries is empirically set to 32, and the threshold $\tau$ for query update linearly varies from 0.1 to 0.5 until epoch 50, and remains constant thereafter.

4.1 Comparison with State-of-the-Art Methods
We compare the segmentation performance between our proposed SIPL framework and the state-of-the-art (SOTA) methods, including nnUNet [Isensee et al., 2021], UNETR [Hatamizadeh et al., 2022], Swin-UNETR [Liu et al., 2021], TransBTS [Wang et al., 2021b], nnFormer [Zhou et al., 2021] and UNETR++ [Abdellahman Shaker et al., 2022]. For fair comparison, all SOTA models were trained/tested on the same dataset splits. The base parameter settings are kept the same as ours.

Multi-organ Segmentation on the BTCV Dataset. We provide a detailed comparison with other methods in the testing set of BTCV in Table 1. In comparison, SIPL outperforms the second-best method by 3.72% in terms of mean DSC scores, demonstrating a significant improvement. Specifically, SIPL achieves the highest DSC scores in 11 organs. The visualization also demonstrates that SIPL obtains favorable segmentation performance with a DSC score of 92.99%, while SIPL achieves the best segmentation performance on the Lungs dataset. As shown, SIPL outperforms the second-best method (UNETR++) by a margin of 0.0001 in terms of mean DSC. The p-values from paired t-tests between the second best method (UNETR++) by a margin of 0.0001.

Table 1: The comparison results with the state-of-the-art models on BTCV test set. The abbreviations are as follows: Spl: spleen, RKid: right kidney, LKid: left kidney, Gal: gallbladder, Eso: esophagus, Liv: liver, Sto: stomach, Aor: aorta, IVC: the inferior vena cava, PSV: portal and splenic veins, Pan: pancreas, RAG: right adrenal gland, LAG: left adrenal gland.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Spl</th>
<th>RKid</th>
<th>LKid</th>
<th>Gal</th>
<th>Eso</th>
<th>Liv</th>
<th>Sto</th>
<th>Aor</th>
<th>IVC</th>
<th>PSV</th>
<th>Pan</th>
<th>RAG</th>
<th>LAG</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNETR</td>
<td>90.48</td>
<td>82.51</td>
<td>86.05</td>
<td>58.23</td>
<td>71.21</td>
<td>94.64</td>
<td>72.06</td>
<td>86.57</td>
<td>76.51</td>
<td>70.37</td>
<td>66.06</td>
<td>66.25</td>
<td>63.04</td>
<td>76.00</td>
</tr>
<tr>
<td>Swin-UNETR</td>
<td>94.59</td>
<td>88.97</td>
<td>92.39</td>
<td>65.37</td>
<td>75.43</td>
<td>95.61</td>
<td>75.57</td>
<td>88.28</td>
<td>81.61</td>
<td>76.30</td>
<td>74.52</td>
<td>68.23</td>
<td>66.02</td>
<td>80.44</td>
</tr>
<tr>
<td>TransBTS</td>
<td>94.55</td>
<td>89.20</td>
<td>90.97</td>
<td>68.38</td>
<td>75.61</td>
<td>96.44</td>
<td>85.82</td>
<td>88.55</td>
<td>82.48</td>
<td>74.21</td>
<td>76.02</td>
<td>67.23</td>
<td>67.03</td>
<td>81.31</td>
</tr>
<tr>
<td>nnFormer</td>
<td>94.58</td>
<td>88.62</td>
<td>93.68</td>
<td>65.29</td>
<td>76.22</td>
<td>96.17</td>
<td>83.59</td>
<td>89.09</td>
<td>80.80</td>
<td>75.97</td>
<td>77.87</td>
<td>70.20</td>
<td>66.05</td>
<td>81.62</td>
</tr>
<tr>
<td>UNETR++</td>
<td>94.94</td>
<td>91.90</td>
<td>93.62</td>
<td>70.75</td>
<td>77.18</td>
<td>95.95</td>
<td>85.15</td>
<td>89.28</td>
<td>83.14</td>
<td>76.91</td>
<td>77.42</td>
<td>72.56</td>
<td>68.17</td>
<td>83.28</td>
</tr>
<tr>
<td>SIPL (ours)</td>
<td>94.90</td>
<td>94.85</td>
<td>95.23</td>
<td>85.37</td>
<td>81.54</td>
<td>97.14</td>
<td>92.99</td>
<td>89.98</td>
<td>85.77</td>
<td>80.79</td>
<td>83.92</td>
<td>71.38</td>
<td>77.15</td>
<td>87.00</td>
</tr>
</tbody>
</table>

Table 2: Comparison results with SOTA methods on the Lungs dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>DSC (%)</th>
<th>p-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNETR</td>
<td>73.29</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>nnUNet</td>
<td>74.31</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Swin-UNETR</td>
<td>75.55</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>nnFormer</td>
<td>77.95</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>UNETR++</td>
<td>80.68</td>
<td>0.0014</td>
</tr>
<tr>
<td>SIPL (ours)</td>
<td>81.50</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 3: Comparison results with SOTA methods on the BraTS dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>DSC (%)</th>
<th>p-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNETR</td>
<td>81.2</td>
<td>0.0003</td>
</tr>
<tr>
<td>Swin-UNETR</td>
<td>81.5</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>nnFormer</td>
<td>82.3</td>
<td>0.0025</td>
</tr>
<tr>
<td>UNETR++</td>
<td>82.8</td>
<td>0.0297</td>
</tr>
<tr>
<td>SIPL (ours)</td>
<td>83.0</td>
<td>--</td>
</tr>
</tbody>
</table>

Figure 5: SIPL ablation study. (a) The influence of different object query numbers. (b) The influence of different threshold $\tau$. Evaluation Results on the Lungs and BraTS Datasets. Table 2 demonstrates the quantitative segmentation performance on the Lungs dataset. As shown, SIPL outperforms the second best method (UNETR++) by a margin of 0.82% in terms of mean DSC. The p-values from paired t-tests between...
our method and the listed state-of-the-art (SOTA) methods are all less than 0.05, indicating a statistically significant performance difference between our method and the listed SOTA methods.

Table 3 displays the comparison results on the BraTS dataset, with the DSC averaged across the three target regions (whole tumor, enhancing tumor, and tumor core). Our method achieves the best segmentation performance, as demonstrated. The p-values (all less than 0.05) further validate that the performance improvements between our method and the listed SOTA methods are statistically significant.

### 4.2 Ablation Study

We now analyze SIPL through a series of ablation studies and all experiments are performed on the BTCV dataset. The DSC scores are leveraged as the default evaluation metric.

**Effectiveness of Each Component.** We evaluate the effectiveness of each component by systematically removing them individually. Table 4 illustrates the results of these ablation experiments. When the IPL module is absent, our SIPL framework experiences a reduction of 13.87% in overall performance. Similarly, the absence of the SMG module leads to a 2.84% reduction in the average DSC score. These findings highlight the significant impact of the IPL module, which enables the learned prototypes to dynamically adapt to different inputs.

**Evaluation on the Influence of Different Prototype Initializations.** We further show the influence of different initialization strategies of the common prototype proposals (CPPs), including one-hot encoding [Zhang et al., 2021], BioBERT embedding [Yasunaga et al., 2022], CLIP embedding [Radosford et al., 2021] and random initialization embedding (referred as RanInit), as shown in Table 5. The performance comparison reveals that CLIP embedding, which utilizes image-text pre-training, significantly improves performance compared to traditional one-hot encoding and text-only pre-trained embedding (BioBERT). Surprisingly, random initialization embedding (RanInit) achieves an even larger improvement of 1.28% compared to CLIP embedding. This unexpected result may be attributed to the fact that predefined fixed embeddings could potentially restrict the learning capabilities of the CPPs in our method.

**Assessment of the Impact of Object Query Numbers.** The impact of varying object query numbers is depicted in Figure 5 (a). It is observed that when the number of object queries is set to 32, the best segmentation performance with a mean DSC of 87.00% is achieved. Results deteriorate when the number of object queries is either too high or too low. A low object query count fails to capture intra-class variation, while a high object query count increases the complexity of pseudo mask generation and may interfere with the main task learning.

**Evaluation of the Different Threshold τ.** We conduct an in-depth analysis of the constant threshold values of τ in Figure 5 (b). The results demonstrate that our framework achieves the best performance when the highest threshold is set to 0.5. Too large or too small threshold values yield inferior outcomes. This is because a large threshold filters out a significant portion of informative pixels during training, hampering the update of query embeddings. Conversely, if the threshold is too small, the learned object embeddings tend to focus on background regions, negatively affecting the segmentation performance.

### 5 Conclusion

In this paper, we propose a novel Self-supervised Instance-adaptive Prototype Learning (SIPL) method for MIS. Our SIPL framework introduces an Instance-adaptive Prototype Learning (IPL) scheme that adaptively learns prototypes tailored to different inputs. Moreover, we also design a Self-supervised Mask Generation (SMG) module to hierarchically generate pseudo masks with confidence scores. These pseudo masks have been validated to improve the segmentation performance by generating more discriminative instance-specific prototype proposals. Experiments and ablation studies validate the effectiveness of each component and demonstrate favorable performance compared to other SOTA methods.
Acknowledgements
This study was supported under the Key Research and Development Program of Zhejiang Province (Grant No. 2023C03192). It was also funded by the National Science Foundation of China (Grant No. 62201341).

Contribution Statement
Guoyan Liang and Qin Zhou made equal contributions. All the authors participated in designing research, performing research, analyzing data, and writing the paper.

References


