Contrastive and Selective Hidden Embeddings for Medical Image Segmentation

Zihao Liu, Zhuowei Li, Zhiqiang Hu, Qing Xia, Ruiqin Xiong, Shaoting Zhang* and Tingting Jiang*

***Abstract*— Medical image segmentation is fundamental and essential for the analysis of medical images. Although prevalent success has been achieved by convolutional neural networks (CNN), challenges are encountered in the domain of medical image analysis by two aspects: 1) lack of discriminative features to handle similar textures of distinct structures and 2) lack of selective features for potential blurred boundaries in medical images. In this paper, we extend the concept of contrastive learning (CL) to the segmentation task to learn more discriminative representation. Specifically, we propose a novel patch-dragsaw contrastive regularization (PDCR) to perform patch-level tugging and repulsing. In addition, a new structure, namely uncertainty-aware feature re-weighting block (UAFR), is designed to address the potential high uncertainty regions in the feature maps and serves as a better feature re-weighting. Our proposed method achieves state-of-the-art results across 8 public datasets from 6 domains. Besides, the method also demonstrates robustness in the limited-data scenario. The code is publicly available at https://github.com/lzh19961031/PDCR_UAFR-MIS.**

***Index Terms*— Medical image segmentation, contrastive learning, uncertainty learning, neural network.**

I. Introduction

Medical image segmentation (MIS) has been widely recognized as a pivot procedure for clinical diagnosis, analysis, and treatment planning. Despite breakthroughs driven by convolutional neural networks (CNN) in recent years, the pace of development for MIS is hindered by two challenges: 1) hard to obtain more discriminative features to tackle the ambiguous boundary and the smooth between-class transition of medical images [1] and 2) difficult to generate more selective features for heterogeneous textures and ambiguous boundaries in medical images [2]. These two challenges prevent CNN from better representation learning and more precise segmentation results.

In the past three years, contrastive learning (CL) has brought impressive breakthroughs and established an overwhelming presence in learning more discriminative representation, which is very suitable for tackling the first challenge. The core concept of CL is the formulation of "pair" [3]–[9]. The formulation includes two aspects: the **component** of "pair" and the **relationship** of the components within one "pair". In previous work, the component is defined as an individual image. The relationship is a rigid bipartition: "positive" or "negative". For example, different augmentations of one image could be defined as a "positive pair", whose representation will be pulled together during learning. Distinct images could be defined as a "negative pair", whose representation will be pushed away. This design is firstly utilized in the unsupervised scenario for backbone pretraining, such as ResNet [10], etc. For supervised classification, [11] extends the definition of "negative pairs" from inter-image to inter-category. Although contrastive learning has been attempted to facilitate MIS [12], [13], they still follow the unsupervised training scheme, and take CL for backbone pretraining. However, the existing formulation of "pair" may not suit MIS due to the following issues:

- The component of "pair". MIS aims to predict the class label for each pixel within one image, which focuses on intra-image discrimination, instead of inter-image. In this sense, it poses demands for more distinctive *local* representation learning. Therefore, it would be more helpful to involve two entities within one image (e.g. local patches or regions) as the component of a pair, instead of two images, for modeling the intra-image relationship.
- The relationship within one "pair". If the component of a pair corresponds to one entity of an image, the relationship between components becomes the relationship between two entities within one image. Because medical images have the characteristic that the transition between different classes is smooth and the boundary is ambiguous, the representation of components of a pair could be similar. Therefore, the relationship between two entities within one image should be also smooth. So the previous rigid bipartition for relationship is inappropriate which could lead to sub-optimal solutions of MIS.

Thus how to effectively formulate "pair" to improve CL for

Zihao Liu is with the Advanced Institute of Information Technology, Peking University, Hangzhou, China, and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing, China, and also with SenseTime Research, China.

Zhuowei Li is with the Rutgers University, New Jersey, USA.

Tingting Jiang is with the Advanced Institute of Information Technology, Peking University, Hangzhou, China, and also with the National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing, China.

Ruiqin Xiong is with the National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing, China.

Zhiqiang Hu and Qing Xia are with SenseTime Research, China.

Shaoting Zhang is with SenseTime Research, China, and also with Shanghai Artificial Intelligence Laboratory, China.

Shaoting Zhang and Tingting Jiang are corresponding authors (e-mail: Zhangshaoting@pjlab.org.cn; ttjiang@pku.edu.cn).

medical image segmentation task is important.

In this paper, we propose a new formulation of "pair". Specifically, we argue that **local patches**, instead of the whole image, are more appropriate for the **component** of "pair". Besides, the relationship formulation should also be different. Pairs are not rigidly bipartite as either positive or negative, but should have a more **soft relationship**. So here the question is: specifically, how to associate patches with its representation for CL? And how could we measure the relationship between different patches?

To this end, we introduce the *patch-dragsaw contrastive regularization* (*PDCR*). Two ways are proposed to associate patches with their representations: the receptive-field-based patch-dragsaw (*RF-PDCR*) and the hierarchical patch-dragsaw (*H-PDCR*). RF-PDCR, which could be seen as a special case of H-PDCR, is an intuitive and direct way to achieve this. It borrows the concept of the receptive field [14] in MIS framework to link each feature vector from a certain layer with a local patch of a certain location and size. However, RF-PDCR is hard to be conducted among features from deeper levels (e.g. decoder) of the network and across multi-levels, due to the applicability and the characteristic of the receptive field concept. Thus, to extend the applicability and tackle the limitation of RF-PDCR, we further put forward H-PDCR. Specifically, for any layer in CNN, each feature vector is associated with a hierarchical patch series including different sizes of patches, to take advantage of the comprehensiveness of multi-scale information. In this way, both single-level CL and multi-level CL could be achieved.

On the other hand, for the relationship, *affinity score* is proposed to measure the similarity of two patches, which serves in a soft matter, instead of rigid bipartition. It is calculated based on the *foreground ratio* for each patch, which is a straightforward and easily-quantifiable metric designed to measure the characteristic of each patch itself. The soft similarity determines the coexisting of tugging and repulsing for the representation of each pair, which could be figuratively described as *dragsaw*. In this way, PDCR aims to strike a balance that respects the segmentation task as well as exploits the contrastive power.

For the second challenge, due to ambiguous boundaries and heterogeneous textures in medical images [2], high uncertainty would come up upon the classification of pixels in these areas, which generates high gradients during training. Along the training process, these high gradients may distract the learning attention of the network and mislead the optimization. Thus, the minority of features with high uncertainty may harm the network. To solve this problem, we introduce the *uncertainty-aware feature re-weighting* (*UAFR*), to ease the influence of pixels with high uncertainty and for a better feature re-weighting. It models the uncertainty for each pixel by generating an *uncertainty-aware map*, then highlights pixels with low uncertainty while suppressing those with high uncertainty. Notice that UAFR does not need any supervision.

It is worth mentioning that both PDCR and UAFR are designed and implemented in a "lightweight plugin" fashion, meaning that they can be easily integrated into existing segmentation frameworks or training pipelines without extra pre-training. Experiments are conducted across 8 public datasets from 6 domains. The results show that our method consistently outperforms previous work.

Our main contribution can be summarized threefold:

- A new formulation of "pair" is proposed to improve the contrastive learning for supervised MIS, including the component and the relationship of "pair". On this basis, a novel *Patch-dragsaw contrastive regularization* (*PDCR*) is proposed to regularize patch-level relations by contrastive constraints, which is implemented in two ways: Receptive-Field-based PDCR and Hierarchical PDCR.
- *Uncertainty-aware feature re-weighting* (*UAFR*) module is designed to avoid the learning-attention shift problem in MIS, which is caused by minority features with high uncertainty due to the characteristics of medical images, and select a better feature.
- State-of-the-art results have been achieved across 8 diverse datasets from 6 domains. Furthermore, we substantiate the potential applicability of proposed methods under a limited data scenario by utilizing only 25% data to outperform the baseline methods with full data.

II. Related Work

A. Architectures for Medical Image Segmentation

A vast of neural networks have been designed and tuned for MIS. Despite the unique design of each architecture, most of the networks employ an encoder-decoder structure with the bypass connection to fuse low-level features with high-level features. Among those designs, FCN [15], UNet [16] and DeepLab [17] function as three milestones. They and their variations provide a stable and consistent baseline for segmentation tasks. Recently, some other methods are proposed [2], [18]–[21]. A structure boundary preserving segmentation framework is proposed in [2] to tackle the ambiguity of boundary. A difficulty-aware deep segmentation network [19] with confidence learning is introduced to obtain more region-wise confidence information.

For 3D MIS, nnUNet [22] shows that network modification may be inferior compared to searching suitable hyper-parameters including augmentations, pre-processing and post-processing techniques, etc. Its pipeline and code-base provide a strong baseline for a broad scope of tasks. [23] proposes global and local contrastive loss for volumetric medical images in the semi-supervised setting, but it still maintains the rigid bipartition for the relationship between patches.

B. Contrastive Learning

Contrastive learning-based model pre-training has largely bridged the gap between supervised and unsupervised model pretraining [3]–[8], [24] by learning to discriminate positive pairs against negative pairs. SimCLR [5] demonstrates the importance of augmentations and shows it is beneficial to maintain a large number of negative pairs and introduce the projection neck. Moco [4] on the other hand, utilizes the memory bank to eliminate the limitation caused by the batch size. Besides, contrastive learning is also utilized for supervised

scenarios [11], [25]. For supervised classification, [11] extends the definition of "negative pair" from inter-image to inter-category.

Besides backbone pre-training, efforts have also been made to fit CL into segmentation tasks. Most methods target at semi-supervised and unsupervised scenarios. A pixel-wise contrastive loss is proposed in [26] for segmentation pretraining and exploring advantages gained under the semi-supervised scenario. To work with meta-label annotations, [12] adapts CL even when no additional unlabeled data is available.

C. Uncertainty-guided segmentation

Several uncertainty-based methodologies are introduced for image segmentation [27]–[32]. According to [27], uncertainty can be divided into two branches in deep learning, *aleatoric uncertainty*, which is caused by data itself and often occurred in boundaries of objects, and *epistemic uncertainty*, which is attributed to limited data and insufficient model training. BEP [32] proposes location-adaptive weighting mechanism for semantics-aware feature pooling, which achieves improvements in both image semantic segmentation and classification tasks. For medical image segmentation, [28] quantifies epistemic uncertainty by running a given sample through the model several times with the dropout layers. A novel uncertainty-aware scheme is designed in [30] to enable the model to gradually learn from the meaningful and reliable targets by exploiting the uncertainty information. In [31], common voxel-wise uncertainty which measures with respect to their reliability in medical image is evaluated.

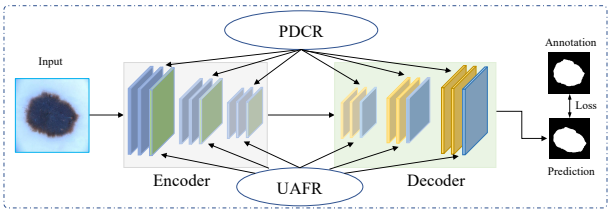

Fig. 1: An overview of proposed method. PDCR denotes patch-dragsaw contrastive regularization. UAFR denotes uncertainty-aware feature re-weighting. They can be inserted at both encoder and decoder.

III. Patch-Dragsaw Contrastive Regularization

In this section, we first give a brief introduction to self-supervised contrastive learning. Then, we introduce the proposed RF-PDCR and H-PDCR.

A. A Review of Self-Supervised Contrastive learning

The existing self-supervised contrastive learning framework generally uses image as the component of pair and defines their relationship as rigid. It contains three principle components: (i) a transformation collection $\mathcal{T}$ (ii) an encoder network $E(\cdot)$ to encode high-dimensional inputs. (iii) a projection head $P(\cdot)$ to further introduce non-linearity and reduce the dimension of output embeddings. As in [5], a positive pair $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_{j(i)})$ is

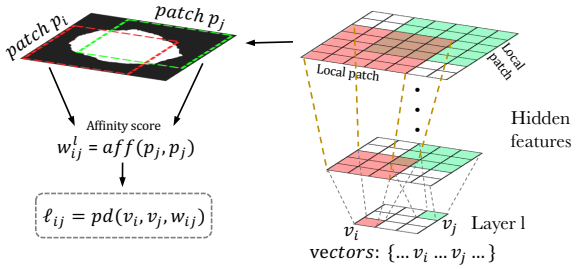

Fig. 2: Receptive-Field-based Patch-Dragsaw Contrastive Regularization (RF-PDCR). According to the concept of receptive field, each feature vector in a certain layer $\boldsymbol{v}_i$ corresponds to a local patch $\boldsymbol{p}_i$ in the image. Given $(\boldsymbol{v}_i, \boldsymbol{v}_j)$, $(\boldsymbol{p}_i, \boldsymbol{p}_j)$ form the pair. w_{ij} is the affinity score of $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. $pd(\cdot)$ calculate the contrastive loss.

acquired by applying randomly sampled augmentations $t \sim \mathcal{T}$ of the same image x twice. Since $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_{j(i)})$ and $(\tilde{\boldsymbol{x}}_{j(i)}, \tilde{\boldsymbol{x}}_i)$ are treated as two pairs, a minibatch with size N_b will attribute to $2N_b$ pairs for training. The contrastive loss is:

$$\mathcal{L}_{batch} = \sum_{i=1}^{2N_b} \mathcal{L}_i \tag{1}$$

$$\mathcal{L}_i = -\log \frac{\exp\left(sim(\boldsymbol{z}_i, \boldsymbol{z}_{j(i)})/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(\boldsymbol{z}_i, \boldsymbol{z}_k))/\tau)} \tag{2}$$

where $\boldsymbol{z}_i = P(E(\tilde{\boldsymbol{x}}_i))$, $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ returns 1 iff $k \neq i$ and $sim(\cdot)$ denotes the cosine similarity.

B. Receptive-Field-based Patch-Dragsaw Contrastive Regularization

In this work, we utilize local patches as the component of pair and define a soft relationship between different patches. Two ways are adopted to generate the representation of the patch, RF-PDCR and H-PDCR. RF-PDCR is first introduced, as shown in Fig. 2. We first describe how to generate the representation for each patch, then the relationship between patches is introduced. After that, we illustrate the complete algorithm.

1) Component for pair and its representation: According to [14], each feature vector from a certain layer corresponds to a local patch. Denote the feature vector in layer l as $\boldsymbol{v}_i$, its corresponding patch is denoted as $\boldsymbol{p}_i$. For simplicity, we drop the layer "l" in the notation, since RF-PDCR conducts on feature vectors from the same layer. Here, to utilize the location information, we encode the position of each patch by concatenating its coordinate (min-x-coordinate, max-x-coordinate, min-y-coordinate, max-y-coordinate) to its feature. Meanwhile, for every layer, we utilize *Grid-sampling* strategy to select N vectors from its feature.

2) Relationship for pair: We define the **affinity score** to measure the soft similarity of every two patches in a pair. Firstly, the foreground ratio vector is obtained to measure the characteristic of each patch, which contains the ratios between the area of each class and the area of the whole patch. Based on the foreground ratio, the affinity score could be calculated.

Specifically, given $(\boldsymbol{v}_i, \boldsymbol{v}_j)$, which contains the location information of its corresponding patch, $(\boldsymbol{p}_i, \boldsymbol{p}_j)$ form the pair. The affinity score $w_{ij} \in [0, 1]$ is a scalar that measures to what extent should this pair be *pulling* and *pushing* for the contrastive loss. Denote the receptive field of $\boldsymbol{v}_i$ as R_i. w_{ij} is:

$$w_{ij} = aff(\boldsymbol{p}_i, \boldsymbol{p}_j) = 1 - \frac{1}{M}\sum_{m=1}^{M} |\phi_i^m - \phi_j^m| \tag{3}$$

where M denotes the number of classes and ϕ_i^m is the foreground ratio of class m for $\boldsymbol{p}_i$. ϕ_i^m measures the area ratio between class m and the total area of $\boldsymbol{p}_i$. The more similar the value of ϕ_i^m and ϕ_j^m, the greater the value of w_{ij}. ϕ_i^m is:

$$\phi_i^m = \frac{\text{Number of pixels in } R_i \text{ which belongs to class } m}{\text{Area of receptive field } R_i} \tag{4}$$

Specifically, ϕ_i^m is computed as following. Suppose $\boldsymbol{v}_i$ is sampled from layer l. The receptive field R_i is a square, and its side length is denoted as r_i. The position of R_i is calculated as follows. Let o_l and e_l denote the start and end location of $\boldsymbol{v}_i$. In our case $o_l = e_l + 1$, meaning sample one vector each time. In each axis (x-axis and y-axis), the start o and end e positions of the receptive field R_i in the original image take the following forms, according to [14], [33]:

$$o = o_l \prod_{t=1}^{l} s_t - \sum_{z=1}^{l} p_z \prod_{t=1}^{z-1} s_t \tag{5}$$

$$e = e_l \prod_{t=1}^{l} s_t - \sum_{z=1}^{l} (1 + p_z - k_z) \prod_{t=1}^{z-1} s_t \tag{6}$$

where k_z denotes the kernel size, s_t represents the stride size and p_z is the padding size of a convolutional layer.

On the other hand, the side length of R_i denoted as r_i , computed as the following form:

$$r_i = \sum_{z=1}^{l} \left((k_z - 1) \prod_{t=1}^{z-1} s_t \right) + 1 \tag{7}$$

where k_z and s_t denote the kernel size and the stride size of a convolutional layer, respectively. Meanwhile, $r_i = o - e + 1$. Thus, a feature vector corresponds to a local patch with a specific location and size. Note that the size of the receptive field r_i only depends on layer l, not related to specific vectors. The area of R_i equals ${r_i}^2$, which is the denominator in Eq. (4). Given the locations of vector $\boldsymbol{v}_i$ in the hidden feature, we can calculate the location of R_i in the original image according to Eq. (5) and Eq. (6). With the calculated location of R_i, the number of pixels that belong to each class m in R_i can be computed according to the image annotation G, which is the numerator in Eq. (4).

3) The Algorithm of RF-PDCR: Algorithm 1 summarizes the ***Receptive-Field-based Patch-dragsaw***. Specifically, it takes the following formation:

$$s_{ij} = sim(\boldsymbol{v}_i, \boldsymbol{v}_j) = \frac{(\boldsymbol{v}_i)^T \boldsymbol{v}_j}{\|\boldsymbol{v}_i\| \, \|\boldsymbol{v}_j\|} \tag{8}$$

$$\mathcal{L} = \sum_{l=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N} -\log \frac{exp([s_{ij} \cdot w_{ij}]/\tau)}{\sum_{n=1}^{N} exp([s_{in} \cdot (1 - w_{in})]/\tau)} \tag{9}$$

where s_{ij} denotes the cosine similarity of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, l denotes the layer and L denotes the number of layers. $(1 - w_{ij})$ measures the dissimilarity.

Algorithm 1 Receptive-field-based Patch-Dragsaw.

Input: encoder $E(\cdot)$, constant N and τ

for image I, annotation G in dataset **do**
  for layer $l \in \{1, ..., L\}$ in encoder **do**
    sampling $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_i, ..., \boldsymbol{v}_j, ..., \boldsymbol{v}_N\}$
    # just conduct CL within single layer
    calculate receptive field according to Eq. (7)
    encode the location information of each patch
    for i $\in \{1, ..., N\}$ **do**
      for j $\in \{1, ..., N\}$ **do**
        compute the foreground ratio of each class for each patch with Eq. (4)
        compute w_{ij} with Eq. (3)
        compute s_{ij} with Eq. (8)
      end for
    end for
  end for
  compute the loss $\mathcal{L}$ with Eq. (9)
  update network E to minimize $\mathcal{L}$
end for

C. Hierarchical Patch-Dragsaw Contrastive Regularization

Despite the simplicity of RF-PDCR, it has two limitations: 1) hard to be conducted in architectures of the network other than the encoder, due to the applicability of the receptive field concept; 2) difficult to be conducted among features from multi-level, due to the different size of patches corresponding to different levels of features. Thus, to improve the implementation generalization and for better representation learning, we introduce hierarchical patch-dragsaw contrastive regularization (H-PDCR), as illustrated in Fig. 3.

1) Component for pair and its representation: According to [34], each vector from the hidden feature of a certain layer corresponds to a specific location in the original image. Then, a hierarchical patch series with this location as the center and different sizes would be obtained, where the sizes are manually assigned. Denote the i_{th} feature vector from layer l as $\boldsymbol{v}_i^l$, it is associated with a hierarchical patch series $\boldsymbol{p}_i^l$, in which the k_{th} patch is denoted as $\boldsymbol{p}_i^l(k)$. Same with RF-PDCR, the location information of each patch would be encoded, and the grid-sampling strategy will be also conducted.

2) Relationship for pair: In H-PDCR, CL is conducted among features from single-level and multi-level. The computation of *affinity score* w still follows Eq. (3), but with two kinds of notations:

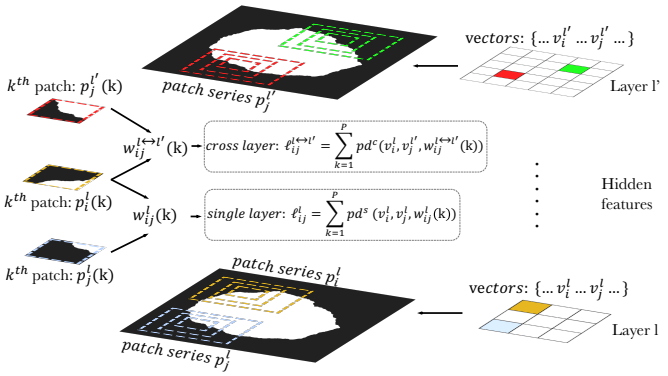

Fig. 3: Hierarchical Patch-Dragsaw Contrastive Regularization (H-PDCR). Each feature vector in a certain layer $\boldsymbol{v}_i^l$ corresponds to a patch series $\boldsymbol{p}_i^l$, in which the k_{th} patch is denoted as $\boldsymbol{p}_i^l(k)$. Given $(\boldsymbol{v}_i^l, \boldsymbol{v}_j^l)$, $(\boldsymbol{p}_i^l(k), \boldsymbol{p}_j^l(k))$ form the pair, since their size are similar, which could benefits the contrastive learning. w_{ij}^l is the affinity score of $\boldsymbol{p}_i^l(k)$ and $\boldsymbol{p}_j^l(k)$. Similarly for $\boldsymbol{v}_i^l$ and $\boldsymbol{v}_j^{l'}$ from different layers, $(\boldsymbol{p}_i^l(k), \boldsymbol{p}_j^{l'}(k))$ form the pair. $w_{ij}^{l\leftrightarrow l'}$ is the affinity score of the pair. $pd_s(\cdot)$ calculate single layer contrastive loss for a pair. $pd_c(\cdot)$ calculate cross layer contrastive loss for a pair.

i. patches from same layer (layer l):

Given $(\boldsymbol{v}_i^l, \boldsymbol{v}_j^l)$, $\boldsymbol{p}_i^l(k)$ and $\boldsymbol{p}_j^l(k)$ form the pair, since their size is similar, which could benefit the contrastive learning.

$w_{ij}^l(k) = aff(\boldsymbol{p}_i^l(k), \boldsymbol{p}_j^l(k))$ is the affinity score of this pair.

ii. patches from different layers (layer l and layer l'):

Given $(\boldsymbol{v}_i^l, \boldsymbol{v}_j^{l'})$, $\boldsymbol{p}_i^l(k)$ and $\boldsymbol{p}_j^{l'}(k)$ form the pair.

Note that before conducting the contrastive learning across different layers, the channel of the two features will be normalized to 32 by a 1×1 conv layer, to ensure that v_i^l (1×32) and v_j^l (1×32) has the same size.

$w_{ij}^{l\leftrightarrow l'}(k) = aff(\boldsymbol{p}_i^l(k), \boldsymbol{p}_j^{l'}(k))$ denotes the affinity score.

3) The Algorithm of H-PDCR: Algorithm 2 summarizes the ***Hierarchical Patch-dragsaw*** contrastive loss. It is:

$$s_{ij}^l = sim(\boldsymbol{v}_i^l, \boldsymbol{v}_j^l) = \frac{(\boldsymbol{v}_i^l)^T \boldsymbol{v}_j^l}{\|\boldsymbol{v}_i^l\| \, \|\boldsymbol{v}_j^l\|} \tag{10}$$

$$s_{ij}^{l\leftrightarrow l'} = sim(\boldsymbol{v}_i^l, \boldsymbol{v}_j^{l'}) = \frac{(\boldsymbol{v}_i^l)^T \boldsymbol{v}_j^{l'}}{\|\boldsymbol{v}_i^l\| \, \|\boldsymbol{v}_j^{l'}\|} \tag{11}$$

$$\begin{aligned}\mathcal{L} = &\sum_{l=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{P} -\log \frac{exp([s_{ij}^l \cdot w_{ij}^l(k)]/\tau)}{\sum_{n=1}^{N} exp([s_{in}^l \cdot (1-w_{in}^l(k))]/\tau)} \\ &+\sum_{l=1}^{L-1}\sum_{l'=l+1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{P} -\log \frac{exp([s_{ij}^{l\leftrightarrow l'} \cdot w_{ij}^{l\leftrightarrow l'}(k)]/\tau)}{\sum_{n=1}^{N} exp([s_{in}^{l\leftrightarrow l'} \cdot (1-w_{in}^{l\leftrightarrow l'}(k))]/\tau)}\end{aligned} \tag{12}$$

where l and l' denotes the layer, P is the number of patches in the patch series. $\boldsymbol{v}_i^l$ is the i_{th} vector from layer l. $(1 - w_{ij}^l(k))$ and $(1 - w_{ij}^{l\leftrightarrow l'}(k))$ measure the dissimilarity.

IV. Uncertainty-Aware Feature re-weighting

This module serves as a better feature re-weighting. As shown in Fig. 4, it generates an uncertainty-aware map, models the uncertainty for each element, and then highlights pixels with low uncertainty while suppressing those with high uncertainty.

Algorithm 2 Hierarchical Patch-Dragsaw.

Input: encoder $E(\cdot)$, decoder $D(\cdot)$, constant N and τ

for image I, annotation G in dataset **do**
  for layer $l \in \{1, ..., L\}$ **do**
    sampling $\{\boldsymbol{v}_1^l, ..., \boldsymbol{v}_i^l, ..., \boldsymbol{v}_j^l, ..., \boldsymbol{v}_N^l\}$
    # CL within single layer
    calculate the center location in the image for each $\boldsymbol{v}$ and generate patch series
    encode the location information of each patch
    for i $\in \{1, ..., N\}$ **do**
      for j $\in \{1, ..., N\}$ **do**
        for k $\in \{1, ..., P\}$ **do**
          # for each patch in patch series
          compute the foreground ratio of each class for each patch by Eq. (4)
          compute $w_{ij}^l(k)$ by referencing Eq. (3)
          compute s_{ij}^l with Eq. (10)
        end for
      end for
      for vector $\{\boldsymbol{v}_1^{l'}, ..., \boldsymbol{v}_N^{l'}\}$ in layer l' $(l \neq l')$ **do**
        # CL across different layers
        for j $\in \{1, ..., N\}$ **do**
          for k $\in \{1, ..., P\}$ **do**
            normalize v_i^l and $v_j^{l'}$ to the same size
            compute the foreground ratio of each class for each patch by Eq. (4)
            compute $w_{ij}^{l\leftrightarrow l'}(k)$ for each patch by Eq. (3)
            compute $s_{ij}^{l\leftrightarrow l'}$ with Eq. (11)
          end for
        end for
      end for
    end for
    # summation of single layer loss and cross layer loss
  end for
  compute the loss $\mathcal{L}$ with Eq. (12)
  update network E to minimize $\mathcal{L}$
end for

Given a hidden feature $Q \in \mathbb{R}^{H\times W\times C}$. Three steps are taken to build the proposed module. Firstly, Q will go through a learning function $\mathcal{F}(\cdot)$ for further representation learning, and generates $A \in \mathbb{R}^{H\times W\times M}$:

$$A = softmax(\mathcal{F}(Q)) \tag{13}$$

Here, $\mathcal{F}(\cdot)$ is a 3×3 conv layer followed by an identity conv layer. Relu activation function and batch-normalization are used in between.

The uncertainty u_{ij} for each location $\boldsymbol{a}_{ij} \in \mathbb{R}^{1\times M}$ in A is modeled according to normalized *Shannon-entropy* [35]:

$$u_{ij} = -\sum_{m}^{M} \frac{\boldsymbol{a}_{ij}^m log_2(\boldsymbol{a}_{ij}^m)}{log_2(M)}. \tag{14}$$

Finally the uncertainty-aware feature map $\hat{U} \in \mathbb{R}^{H\times W}$ is acquired and imposed back to the hidden feature:

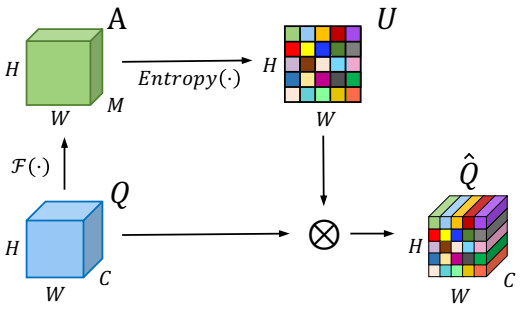

Fig. 4: Uncertainty-Aware Feature Re-weighting (UAFR). $\mathcal{F}(\cdot)$ performs segmentation at specific scale and $Entropy(\cdot)$ represents softmax function followed by Shannon-entropy across channels.

$$\hat{U} = 1 - U, \qquad \hat{Q} = Q * (1 + \hat{U}) \tag{15}$$

where "*" denotes element-wise multiplication. The new hidden feature $\hat{Q}$ is then passed to the following layers instead of the original one Q.

V. Experiments

A. Datasets and Implementation Details

To better evaluate the generalization on a broad scope of medical image segmentation tasks, we conduct experiments on both 2D and 3D datasets, including eight datasets across six domains in total. Table I elaborates task details. The performance is measured by the Jaccard index (JA), Dice coefficient (DI), and pixel-wise accuracy (AC).

We utilize DeeplabV3+ [17] (2D) and nnUNet [22] (3D) as the baseline architecture. To supervise the segmentation task, we utilize cross-entropy loss. If PDCR is used, we add it as a regularization to cross-entropy loss with a weight of 0.01. H-PDCR is equipped at encoder, decoder and cross-layers according to the ablation study. UAFR, is added to all encoder and decoder layers (except the neck module in DeeplabV3+). N is set to 128, 64, 32, 8, 8, 8, 8, 8 for each layer. P is set to 3, the size of patches in patch series of H-PDCR is set to 5 × 5, 7 × 7 and 9 × 9. Other hyper-parameters like learning rate scheduler, augmentation strategy, etc, are kept the same across all 2D or 3D tasks. For 2D tasks, we utilize the Adam optimizer with the base learning rate of 0.001 and the cosine scheduler. The weight decay is set to 0.00001. Applied augmentation includes scale, flip, rotate, shift, and shear transformation. Specifically, we conduct random scaling between 0.8 and 1.2, random rotation from -90 to 90 degrees, shear from -20 to 20 degrees, shift with 0.1, and random flips for both axes. As for training, the input image is resized to 512 × 512 and batch size is set to 64 on 16 NVIDIA GTX 1080Ti GPUs. We train each model for 300 epochs. For 3D tasks, we follow the settings of [57], [59], and the hyper-parameters follow the default settings of nnUNet. Specifically, the input images are first resampled to have an isotropic voxel spacing of 0.7mm × 0.7mm × 0.7mm, followed by z-normalization separately applied to each input channel. In the training phase, each image is randomly cropped to the region of nonzero values with the pre-defined resolution. For testing, the final segmentation results are obtained by combining patch-based inference results with a 50 percent overlap. The input size of each patch is modified for each task like nnUNet. We utilize the same data augmentation strategy as [57] in both the training phase and testing phase.

B. Benchmark Studies

Table II shows the results for the proposed model against other state-of-the-art methods for the aforementioned medical image segmentation tasks. It is encouraging to see that our model, with a task-agnostic design, achieves a consistent advantage over other task-tailored counterparts. We believe it is evidence that our proposed method exhibits universal applicability for the medical image segmentation research area.

Besides, we conduct t-tests and calculate the standard deviation on each dataset against methods ranked second and third. Shown in Table III, the results confirm that the improvement against other methods is statistically significant.

C. Ablation Studies

We conduct ablation studies on the ISIC 2016 dataset using DeeplabV3+ with xception encoder as the backbone network. The dataset is chosen for a balance of dataset size and training budget.

1) Effect of the proposed two modules and computation complexity analysis: In this subsection, we investigate whether the proposed two modules are beneficial on top of different architectures on ISIC 2016 dataset. To this end, we add either or both of the two modules to FCN, UNet, in addition to DeeplabV3+, which are arguably the most common segmentation frameworks among others. Table IV shows the results. Two observations are worth noting: Firstly, both PDCR (including RF-PDCR and H-PDCR) and UAFR demonstrate consistent improvements on all of the three backbones, indicating the general applicability; Secondly, the two modules are complementary in the sense that they are shown to benefit on top of each other. Besides, we compare the number of parameters and FLOPs in 512×384 input size for inference on several existing methods, as shown in Table IV. Note that PDCR is only involved in training, and UAFR is a lightweight module.

2) Ablation Study for PDCR: We adopt PDCR and its variations on the backbone to investigate its best configuration.

1. The best configuration. According to Section III, RF-PDCR is only used to encoder layers for the applicability of the receptive field concept, meanwhile H-PDCR could be extended to all the layers in each part of the network. We investigate where should PDCR be plugged in, by taking the last layer of each block as well as every single layer as candidate sites and ablating PDCR layer-by-layer.

Fig. 5 shows the ablation results for RF-PDCR. It can be clearly seen that if RF-PDCR is adopted in a single layer, the performance will be improved no matter adopted in which layer. However, when adopting RF-PDCR in multiple layers simultaneously, we notice that adopting in layer 2,3,4 of the encoder performs the best.

TABLE I: Details of tasks and datasets used in experiments.

Dataset	Domain	Task	Number of category	Dimension	Modality	Evaluation Protocol
ISIC 2016 [36]+PH2 [37]	Skin	Lesion seg	2	2D	Dermoscopy	900 train(ISIC2016), 200 test(PH2) (Official split)
ISIC 2016 [36]	Skin	Lesion seg	2	2D	Dermoscopy	900 train, 379 test (Official split)
ISIC 2017 [38]	Skin	Lesion seg	2	2D	Dermoscopy	2000 train, 600 test (Official split)
MC [39]	Lung	Lung seg	2	2D	X-Ray	80 train, 58 test
DigestPath [40]	Pathology	Lesion seg	2	2D	Pathology	952 train, 5-fold cross-validation
Decathlon [41]-Task01	Brain	Tumour seg	4	3D	MRI	454 train, 5-fold cross-validation (Official split)
Decathlon [41]-Task02	Heart	Heart seg	2	3D	MRI	20 train, 5-fold cross-validation (Official split)
Decathlon [41]-Task05	Prostate	Prostate seg	3	3D	MRI	32 train, 5-fold cross-validation (Official split)

TABLE II: Results of the proposed model against state-of-the-art methods on a broad scope of medical image segmentation tasks.

ISIC 2016+PH2 (Skin)			ISIC 2016 (Skin)				ISIC 2017 (Skin)				MC (Lung)		
Methods	DI	JA	Methods	DI	JA	AC	Methods	DI	JA	AC	Methods	JA	AC
MSCA [42]	81.57	72.33	DFCN [43]	91.20	84.70	95.50	FCN+SSP [44]	85.7	77.3	93.8	FCN [15]	90.53	97.35
SSLS [45]	78.38	68.16	MSFCN [46]	91.18	84.64	95.51	Bi et al. [47]	85.66	77.73	94.08	UNet [16]	91.64	97.82
FCN [15]	89.40	82.15	ECDN [48]	91.30	84.90	95.70	SLSDeep [49]	87.8	78.2	93.6	M-Net [50]	91.95	97.96
Bi et al. [46]	90.66	83.99	Bi et al. [47]	91.77	85.92	95.78	MBDCNN [51]	87.8	80.4	94.7	Multi-task [52]	92.24	98.13
SBPS [2]	91.84	84.30	biDFL [53]	93.33	88.12	96.75	biDFL [53]	88.54	81.47	94.65	ETNet [54]	94.20	98.65
Ours	**93.51**	**86.74**	Ours	**94.51**	**88.94**	**97.71**	Ours	**89.63**	**82.00**	**95.64**	Ours	**95.30**	**99.68**

DigestPath (Pathology)				Decathlon								
					Brain Tumor				Heart	Prostate		
Methods	DI	JA	AC	Methods	ED	NE	EN	Avg	LA	PE	TR	Avg
FCN [15]	77.98	64.07	94.32	U-ResNet [55]	79.10	58.38	77.37	71.61	91.48	48.37	79.17	63.77
UNet [16]	77.33	63.17	94.29	nnUNet NoDA [22]	81.27	60.92	77.90	73.36	92.85	58.61	83.61	71.11
Dilated-Net [56]	77.34	63.45	94.27	nnUNet [22]	81.68	61.29	77.97	73.65	93.21	63.14	86.53	74.84
DeeplabV3+ [17]	77.92	63.88	94.33	SCNAS [57]	80.41	59.85	78.50	72.92	91.91	53.81	82.02	67.92
-	-	-	-	ASNG [58]	81.94	61.85	79.35	74.38	93.27	67.65	87.04	77.35
Ours	**79.44**	**65.42**	**95.78**	Ours	**83.15**	**65.82**	**85.09**	**78.02**	**94.38**	**68.83**	**88.61**	**78.72**

TABLE III: Standard deviation and t-test against other methods. The improvement is statistically significant.

Dataset	Method (Ours vs)	t-value	p-value	Significant at p <0.05?	Standard deviation
PH2	SBPS (rank2) Bi et al. (rank3)	2.172(DI) 5.053(DI)	0.0161(DI) p<0.00001(DI)	Yes Yes	0.0003
ISIC2016	biDFL (rank2) Bi et al. (rank3)	2.620(DI) 6.459(DI)	0.0087(DI) p<0.0000(DI)	Yes Yes	0.0002
ISIC2017	biDFL (rank2) MBDCNN (rank3)	1.983(DI) 2.787(DI)	0.047(DI) 0.004(DI)	Yes Yes	0.0002
MC	ETNet (rank2) Multi-task (rank3)	6.179(AC) 9.362(AC)	p<0.0001(AC) p<0.0001(AC)	Yes Yes	0.0004
DigestPath	DeeplabV3+ (rank2) Dilated-Net (rank3)	4.1388(DI) 5.7181(DI)	p<0.0001(DI) p<0.0001(DI)	Yes Yes	0.0045
Decathlon (Brain Tumour)	ASNG (rank2) SCNAS (rank3)	3.3401(DI, Avg) 4.6798(DI, Avg)	0.0009(DI, Avg) p<0.0001(DI, Avg)	Yes Yes	0.0019
Decathlon (Heart)	ASNG (rank2) SCNAS (rank3)	3.0085(DI) 6.6946(DI)	0.0072(DI) p<0.0001(DI)	Yes Yes	0.0027
Decathlon (Prostate)	ASNG (rank2) SCNAS (rank3)	2.7813(DI, Avg) 21.9253(DI, Avg)	0.0091(DI, Avg) p<0.0001(DI, Avg)	Yes Yes	0.0032

TABLE IV: Results of ablation studies for different architectures on ISIC 2016 and comparison of computation complexity. A strong and consistent improvement is proved. Besides, both two modules could be implemented in a "lightweight plugin" fashion, which only brings few extra parameters.

Architecture	Parameters	FLOPs	+RF-PDCR	+H-PDCR	+UAFR	JA	DI	AC
UNet	17.3M	120.4G				78.81	86.15	92.05
UNet	17.3M	120.4G	✓			80.04	87.42	92.86
UNet	17.3M	120.4G		✓		80.52	87.94	93.14
UNet	18.8M	128.0G			✓	80.02	87.17	93.20
UNet	18.8M	128.0G		✓	✓	**81.31**	**88.34**	**94.16**
FCN	18.6M	76.5G				80.42	86.46	92.30
FCN	18.6M	76.5G	✓			81.55	87.83	93.65
FCN	18.6M	76.5G		✓		82.37	88.56	94.50
FCN	19.2M	78.3G			✓	82.04	88.32	94.67
FCN	19.2M	78.3G		✓	✓	**83.13**	**89.54**	**95.51**
DeeplabV3+	54.7M	62.7G				86.09	91.17	95.41
DeeplabV3+	54.7M	62.7G	✓			86.52	92.24	96.17
DeeplabV3+	54.7M	62.7G		✓		87.99	93.45	97.01
DeeplabV3+	55.5M	64.1G			✓	87.81	93.50	96.79
DeeplabV3+(Ours)	55.5M	64.1G		✓	✓	**88.94**	**94.51**	**97.71**

On this basis, we contrast H-PDCR with RF-PDCR. We conduct experiments in two aspects: investigate the impact of H-PDCR in each part of DeeplabV3+, including encoder, decoder and cross-layer; compare the performance of H-PDCR and RF-PDCR in the encoder (RF-PDCR could only be adopted in the encoder due to its applicability). The results are shown in Table VI. In Table VI (a), the results indicate that adopting manually assigned patches for H-PDCR in all parts of the network could improve the results. Besides, in Table VI (b), while adopting H-PDCR in the decoder and cross-layer, for the encoder, we investigate whether adopting receptive-field patch (RF-PDCR) or manually assigned patches (H-PDCR) performs better. The results show that manually assigned patch (H-PDCR) performs better, proving that H-PDCR could take more advantage of the multi-scale information and facilitate the network.

On the other hand, we investigate the specific configuration of H-PDCR by adopting it on every single layer. The results are shown in Table VII, indicating that adopting H-PDCR on adjacent level of layers could boost the performance more, which is better than adopting on layers with large spans. The best setting for H-PDCR is adopted in single layer (2,3,4 of the encoder) and across features from different layers (layer 2 and 3 of the encoder). In summary: **it is suggested to adopt RF-PDCR at central layers of the network, where the receptive field is neither too large nor too small. Similarly with H-PDCR. Besides, H-PDCR could be extended to decoder**

and cross-layer.

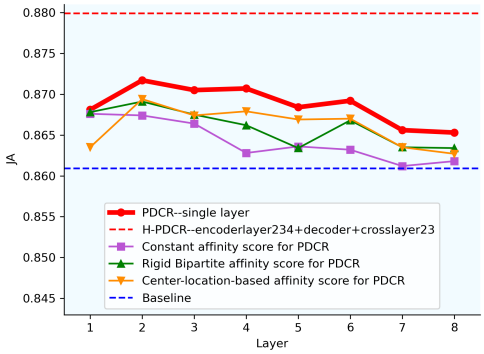

Fig. 5: Ablation study for PDCR on ISIC 2016. We ablate from two aspects: the effect of PDCR in each layer and whether PDCR is better than other similar designs. The red line denotes adding PDCR at each layer in the encoder. The orange, green and purple lines denote adopting three similar designs of affinity score w layer-wisely. The blue dot line denotes using the baseline model and the red dot line denotes the best usage of PDCR. The results show that PDCR is effective for every layer, and performs the best when adding to layer 2,3 and 4. Besides, PDCR outperforms other similar designs.

2. Design variations. We ablate other similar designs for PDCR:

(1) **Variations for foreground ratio ϕ in PDCR.** Instead of Eq. (4), we evaluate two other definitions of ϕ for each patch: L1-norm and L2-norm. The results are shown in Table V (a), which prove that our definition performs better.

(2) **Variations for affinity score w in PDCR.** Here, we substantiate 3 variations listed as following:

- **Constant** w: downgrades w to a constant temperature coefficient for each two patches (w = 0.5).
- **Rigid bipartite** w: setting each two patches as rigid positive or negative pairs like in SimCLR, which indicates that w = 1 when a patch with its transformation and w = 0 for each two different patches. Note that the transformation here means the data augmentation, which includes scale, flip, rotate, shift, and shear transformation and keeps the same with all the other experiments in Section V A.
- **Center-location-based** w: w = 1 when the label of the center location of two vectors are the same, 0 otherwise.

According to the results shown in Fig. 5, our design of w outperforms the other three variations.

(3) **Variations of two implementations for H-PDCR.** There are two different implementations for H-PDCR, the first one is to calculate the contrastive loss for each patch pair in the series first, then average the loss, as mentioned in Eq. (9) of Section III. The second one is to calculate the average of foreground ratios for the patches in a series first, then conduct contrastive learning. We evaluate the impact of these two implementations. The results are shown in Table V (b), which reveals that the first one performs better.

(4) **The impact of patch position encoding.** We evaluate the impact of encoding the the location information of each patch(mentioned in Section III B 1). The results are shown in Table V (c), indicating that the location information would benefit the model.

(5) **The variations of measuring the affinity score of patches.** We evaluate two ways to measure the similarity of patches in Eq. (3). The first way is utilizing foreground ratio (Eq. (3)). The second way is to use the dice similarity of each two patches in the segmentation mask to replace the foreground ratio, which makes Eq. (3) as follows:

$$w_{ij} = aff(\boldsymbol{p}_i, \boldsymbol{p}_j) = 1 - \frac{1}{M}\sum_{m=1}^{M} dice(\boldsymbol{p}_i^m, \boldsymbol{p}_j^m) \qquad (16)$$

The results are shown in Table V (d), indicating that the first way performs better. We think this is because the second way relies too much on the location of the foreground (lesion) area. For example, if the two patches have a similar foreground ratio but their foreground area is in a totally different location, the first way tends to regard these two patches as more similar ones. However, the second way will regard them as totally dissimilar patches. This may hurt the performance.

(6) **Comparison with other similar-designed SOTA methods.** In Table V (e), we compare with three image-level contrastive learning methods, including BYOL [6], SimCLR [5], SimSiam [9] on ISIC 2016. They are first used to pre-train the model on ISIC 2016 dataset in a self-supervised fashion, then fine-tuned supervisely. Our method outperforms all of them.

3) Ablation Study for UAFR: Similarly to PDCR, we adopt UAFR and its variations on the backbone to investigate its best configuration.

1. The best configuration. We investigate the configuration of UAFR by adopting it in every single layer of DeepLabV3+. As illustrated in Fig. 6, UAFR is more robust against the plugged-in position and performs roughly at both encoder and decoder. We do not add it at aspp neck("Atrous Spatial Pyramid Pooling") in DeeplabV3+ framework since it is the concatenation of multiple conv layers. Similarly, the benefits are additive and we finally add them to all layers in the encoder and decoder. To summarize: **it is suggested to adopt UAFR at any layer you like in the encoder and decoder.**

2. Design Variations. We ablate other similar designs for UAFR:

(1) Comparison with other similar-designed methods. We compare UAFR with two uncertainty-based methods, including Active FCN [28], UAMT [30] and two popular attention mechanisms, including SENet [60] and CBAM [61], to see the changes brought to the performance. As seen in Fig. 6 and Table VIII, UAFR outperforms four counterparts by a large margin. It is believed that improvements are gained from regions with ambiguous boundaries region as well as heterogeneous textures. Visualization from Fig. 9 also substantiates our intuition.

(2) The impact of different variation for $\mathcal{F}(\cdot)$. We conduct experiments to evaluate the effect of $\mathcal{F}(\cdot)$, including two aspects: whether adopting $\mathcal{F}(\cdot)$ before uncertainty measuring; the impact of different output channel size (M). In

TABLE V: Results of similar designs for PDCR on ISIC 2016. (a) investigates several variations for foreground ratio ϕ. (b) investigates two implementations for H-PDCR. (c) investigates the impact of patch position encoding. (d) compares our method to a variation of measuring the affinity score of patches. (e) compares our method to other similar-designed contrastive learning methods. Our method outperforms other similar designs.

(a)

Foreground ratio ϕ	RF-PDCR	H-PDCR	JA	DI	AC
L1-norm	✓		87.89	93.45	96.87
L1-norm		✓	88.43	94.18	97.25
L2-norm	✓		87.52	93.01	96.19
L2-norm		✓	88.04	93.68	96.99
Ours	✓		88.24	93.97	97.21
Ours		✓	**88.94**	**94.51**	**97.71**

(b)

Method	Calculating foreground ratio for H-PDCR	JA	DI	AC
Ours	Average the foreground ratio of patches first	88.75	94.31	97.69
Ours	Calculate the loss of each size of patches first	**88.94**	**94.51**	**97.71**

(c)

Method	With or without patch position encoding	JA	DI	AC
Ours	Without patch position encoding	88.85	94.40	97.67
Ours	With patch position encoding	**88.94**	**94.51**	**97.71**

(d)

Method	Measuring the affinity score of patches	JA	DI	AC
Ours	Dice similarity metric	88.61	94.08	97.22
Ours	Foreground ratio	**88.94**	**94.51**	**97.71**

(e)

CL Method	The level of the component for CL	JA	DI	AC
BYOL	Image-level contrastive learning	86.78	92.75	96.56
SimCLR	Image-level contrastive learning	86.88	93.01	96.75
SimSiam	Image-level contrastive learning	87.11	93.14	96.41
Our proposed	Patch-level contrastive learning	**88.94**	**94.51**	**97.71**

TABLE VI: Results of contrasting H-PDCR with RF-PDCR. (a) investigates the impact of H-PDCR by adopting it in all parts of DeeplabV3+, including encoder, decoder and cross-layer. (b) compares the performance of H-PDCR and RF-PDCR in the encoder. H-PDCR performs better.

(a)

ISIC2016	RF-PDCR (encoder)	H-PDCR (encoder)	H-PDCR (decoder)	H-PDCR (cross-layer)	JA	DI	AC
DeeplabV3+	✓				87.20	92.46	96.18
DeeplabV3+		✓			87.29	92.81	96.22
DeeplabV3+			✓		86.70	92.31	96.32
DeeplabV3+				✓	87.00	92.65	96.30
DeeplabV3+		✓	✓		87.30	93.16	96.61
DeeplabV3+		✓		✓	87.38	93.23	96.62
DeeplabV3+			✓	✓	87.03	92.84	96.54
DeeplabV3+		✓	✓	✓	**87.99**	**93.45**	**97.01**

(b)

ISIC2016	RF-PDCR (encoder)	H-PDCR (encoder)	H-PDCR (decoder)	H-PDCR (cross-layer)	JA	DI	AC
DeeplabV3+		✓	✓	✓	**87.99**	**93.45**	**97.01**
DeeplabV3+	✓		✓	✓	87.83	93.33	96.99

TABLE VII: Results of the configuration for H-PDCR on ISIC 2016, which includes the performance of several layers and the combination of them. The best setting for H-PDCR is adopted in single layer (2,3,4 of the encoder) and across features from different layers (layer 2 and 3 of the encoder). Besides, adopting H-PDCR on layers with large spans will hurt the performance.

ISIC2016	H-PDCR (encoder layer 1)	H-PDCR (encoder layer 2)	H-PDCR (encoder layer 3)	H-PDCR (encoder layer 4)	H-PDCR (encoder layer 5)	JA	DI	AC
DeeplabV3+						86.09	91.17	95.41
DeeplabV3+	✓			✓		87.00	92.66	96.38
DeeplabV3+	✓				✓	86.98	92.64	96.37
DeeplabV3+	✓	✓			✓	87.07	92.71	96.43
DeeplabV3+	✓		✓		✓	87.05	92.73	96.40
DeeplabV3+		✓	✓		✓	87.11	92.76	96.44
DeeplabV3+	✓	✓	✓	✓		87.18	92.79	96.46
DeeplabV3+	✓	✓	✓	✓	✓	87.19	92.80	96.46
DeeplabV3+		✓	✓	✓		87.29	93.09	96.52

DeeplabV3+	H-PDCR (cross encoder layer 1 and 4)	H-PDCR (cross encoder layer 2 and 3)	JA	DI	AC
DeeplabV3+	✓		86.50	91.93	96.01
DeeplabV3+		✓	87.00	92.65	96.30
DeeplabV3+	encoder layer 2 & encoder layer 3 & encoder layer 4 & cross encoder layer 2 and 3		**87.99**	**93.45**	**97.01**

Table IX (a), the results show that $\mathcal{F}(\cdot)$ could effectively learn the semantic information of Q and facilitate the following uncertainty measuring. In Table IX (b), the results show that when M equals 2, which is the segmentation class number, the performance is the best. And when M is close to 2, the performance is also close to which of M=2. But, when M is bigger than 100, the performance drops dramatically. These results show that without supervision, the uncertainty measurement does not demand the channel number to be exactly the same as the segmentation class number, it could be calculated with an arbitrary number of channel. But, when the channel number is close to the segmentation class number, the performance would be better, indicating that the network does learn better feature by measuring the uncertainty according to the information of each class.

(3) convolutional kernel size of $\mathcal{F}(\cdot)$.

We conduct experiments to evaluate the effect of different convolutional kernel sizes for $\mathcal{F}(\cdot)$. The results are shown in Table X, which indicates that 3×3 is the best setting.

D. On Limited Training Data

Both our PDCR and UAFR are designed for robust and effective performance with limited trainset sizes. In this

TABLE VIII: Results of other similar-designed methods for UAFR on ISIC 2016. Our method performs the best.

Architecture	Method Type	JA	DI	AC
Our proposed	Uncertainty-based learning	**88.94**	**94.51**	**97.71**
Active FCN	Uncertainty-based learning	85.01	91.32	95.89
UAMT	Uncertainty-based learning	85.47	91.53	95.70
CBAM	Attention-based	86.48	92.66	96.45
SENet	Attention-based	86.56	92.77	96.23

subsection, we explore the robustness of proposed methods against the data reduction on ISIC 2016 dataset. Specifically, we gradually decrease the amount of training data from 50% of the whole set to 1% with a stride of 5% and see the performance changes along the way. The results are shown in Fig. 7. It is encouraging to see that the performance is less affected by the data size in a broad range from 25% to 100%. Noticeably, the model still performs on par with the whole-set baseline with as little as 20% of overall training data. This property reveals the great potential of proposed methods in handling other especially long-tailed domains where purely big-data-driven methods may not be applicable.

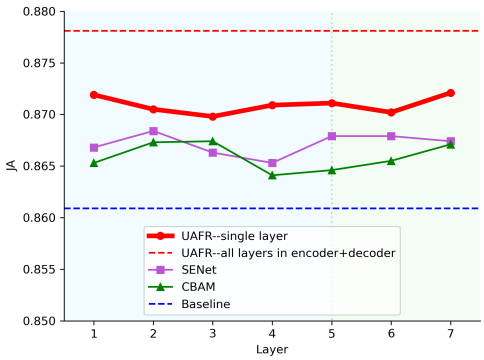

Fig. 6: Ablation study for UAFR on ISIC 2016. We ablate from two aspects: the effect of UAFR in each layer and whether UAFR is better than other similar designs. The red line denotes adding UAFR at each layer in the encoder (layer 1-5) and decoder (layer 6-7). The green and purple lines denote layer-wisely adopting two other similar attention mechanisms: SENet and CBAM. The blue dot line denotes using the baseline model and the red dot line denotes the best usage of UAFR. The results show that UAFR is effective for every layer, and performs the best when added at all layers. Besides, UAFR outperforms SENet and CBAM.

TABLE IX: Results of design variations for UAFR on ISIC 2016. (a) investigate the impact of adopting $\mathcal{F}(\cdot)$ before uncertainty measuring. (b) investigate the impact of different value for the channel number M in A. Our design performs the best.

(a)

Architecture	Whether adopting $\mathcal{F}(\cdot)$	JA	DI	AC
DeeplabV3+	No	87.17	92.91	96.49
DeeplabV3+	Yes	**87.81**	**93.50**	**96.79**

(b)

Architecture	Channel number M of A	JA	DI	AC
DeeplabV3+	$M = 1$	87.75	93.48	96.68
DeeplabV3+	$M = 2$	**87.81**	**93.50**	**96.79**
DeeplabV3+	$M = 5$	87.55	93.35	96.77
DeeplabV3+	$M = 10$	87.62	93.42	96.73
DeeplabV3+	$M = 50$	87.53	93.29	96.73
DeeplabV3+	$M = 100$	87.50	93.26	96.72
DeeplabV3+	$M = 500$	86.96	92.92	96.39
DeeplabV3+	$M = 1000$	86.86	92.90	96.28

E. Visualization Results

Qualitative analysis is performed here to facilitate the understanding. Fig. 9 visually compares the segmentation results of Deeplab with our proposed method (Deeplab-based). First to third rows, fourth to sixth rows, seventh to eighth rows are the results of ISIC2017, DigestPath, MC datasets respectively. As shown in red circles, our method achieves better boundary and structure results. For uncertainty maps of UAFR (Fig. 9(e)), they are generated by using the feature in layer 2 of the encoder of DeeplabV3+. In the uncertainty maps, regions with ambiguous boundaries or heterogeneous textures are clearly highlighted, meaning that they are successfully located by the network and receive lower weights. Besides, the segmentation results are facilitated by suppressing these areas, which are shown in the dice score below each row of Fig. 9(c) and Fig. 9(d).

TABLE X: Results of different convolutional kernel size of $\mathcal{F}(\cdot)$ for UAFR on ISIC 2016. 3×3 kernel size performs the best.

Architecture	Kernel size of $\mathcal{F}(\cdot)$	JA	DI	AC
DeeplabV3+	1×1	87.14	93.13	96.77
DeeplabV3+	3×3	**87.81**	**93.50**	**96.79**
DeeplabV3+	5×5	87.50	93.39	96.67
DeeplabV3+	7×7	87.27	93.23	96.73

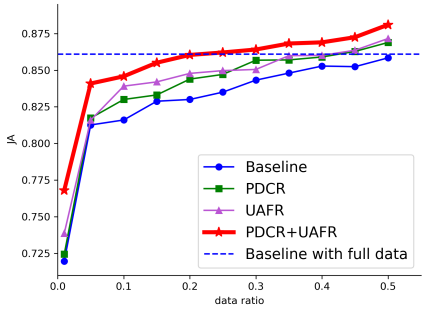

Fig. 7: Comparison of proposed two modules against the baseline with varied training data sizes. Model equipped with both modules surpasses the baseline model trained with all data by using only 25% data.

For PDCR, we follow the setting of [26] and use t-SNE [62] to visualize the effect. Specifically, we first take 128 sampled hidden vectors from each 2-4 layers of the network with or without PDCR. Then, t-SNE is used to visualize the distribution for each class. Here the class means the segmentation category of an image. Thus the distribution of different classes could be generated, which reflects whether the feature is discriminative among different classes. As shown in Fig. 8, the comparison clearly shows that PDCR is effective in discriminating different hidden vectors through clustering and it is owing to, as we think, the power of continuously controlled contrastive regularization. Besides, H-PDCR achieves better discrimination results, proving the effectiveness of its extension from RF-PDCR.

F. Limitation of our work

The main limitation of our work is that for measuring the affinity score of patches, the foreground ratio does not consider the spatial distribution information. We try two ways: the first way is foreground ratio, and the second way is dice similarity. The first way performs better, but in few cases, the two patches will share a similar foreground ratio (area ratio between foreground region and background region is similar) but their foreground region has different distribution information (for example scattered v.s. centrally distributed). The spatial distribution information could be better utilized.

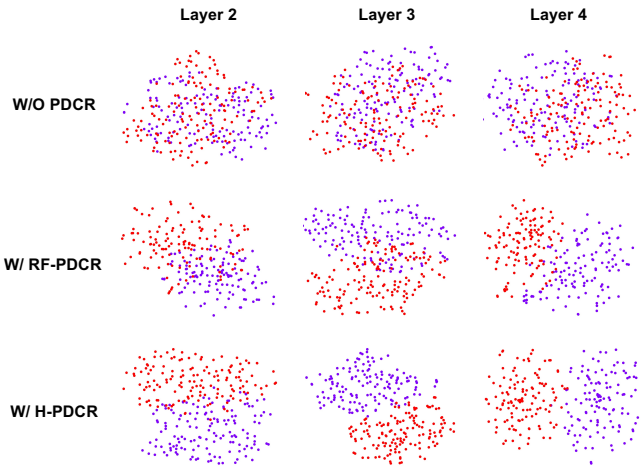

Fig. 8: The distributions of various classes for sampled hidden vectors from 2-4 layers of encoder without and with PDCR. Each color denotes a class. It's convincing that PDCR helps to increase discrimination ability by encouraging clustering. Besides, compared to RF-PDCR, H-PDCR clusters better, which is in line with its better quantitative results.

VI. Conclusion and Future Work

In this work, we propose patch-dragsaw contrastive regularization (PDCR) to generate more discriminate features, which is a key challenge of medical image segmentation tasks. In PDCR, we re-formulate the "pair" of contrastive leaning, including the component and the relationship, to regularize patch-level relations by contrastive constraints. Besides, another module named uncertainty-aware feature re-weighting (UAFR) is introduced to perform feature selection according to the uncertainty modeled by the network itself, to generate more selective representations.

The experiment results verify the effectiveness of PDCR and UAFR. Specifically, PDCR proposes a new formulation of the component (local patches) and the relationship (soft instead of rigid) for contrastive learning. The experiment results of PDCR achieve consistent improvement both quantitatively and qualitatively, and outperform other similar variations. This is highly in line with the motivation, which means that the network could generate more discriminative features. Besides, extending PDCR on all architectures (H-PDCR) is better than only adopting on the encoder of the segmentation framework (RF-PDCR). On the other hand, UAFR models the uncertainty of each pixel and poses different weights on features. The quantitative and qualitative results verify that it successfully generates more selective features and outperforms other similar designs. Other than those, promising results have been shown in the limited-data scenario.

In future work, we will improve on two main aspects: (1) Combine more information to measure the affinity score on the basis of foreground ratio information. For instance, spatial distribution or structure information for every two patches in a pair could be considered and measured. (2) Further explore the functionality of the proposed two modules in un-/semi-supervised learning.

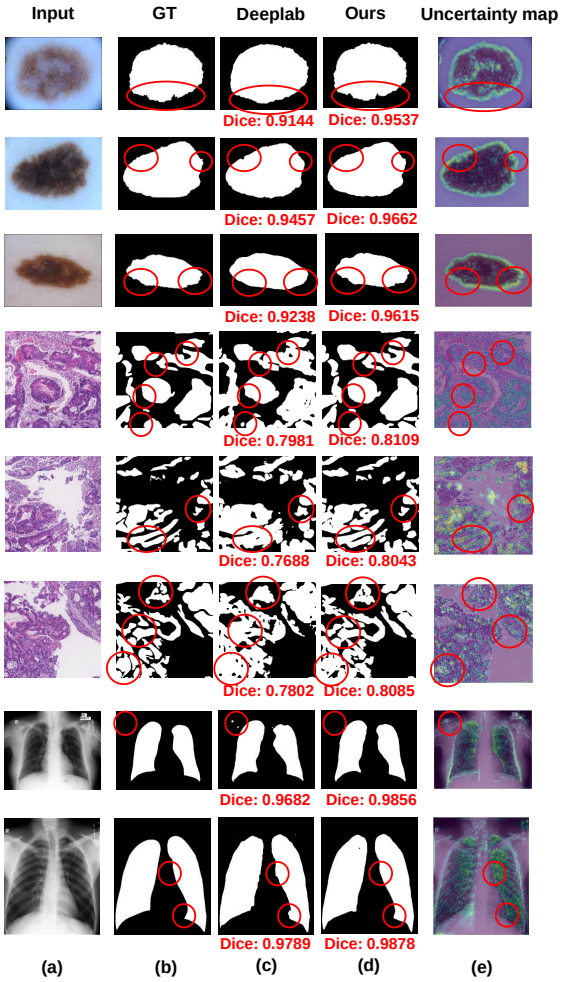

Fig. 9: Visualization results of Deeplab and our method (Deeplab-based) on ISIC 2017 (1-3 rows), DigestPath (4-6 rows) and MC (7-8 rows). (a) is the original images, (b) is the groundtruth mask, (c) is the results of the Deeplab, (d) is the result of our method and (e) is the visualization results of the generated uncertainty map. In (e), regions with lighter color denotes larger uncertainty, and the red circles denote regions with high uncertainty. It can be seen that obscured boundaries and heterogeneous features could be effectively detected, resulting to more selective representations. Besides, quantitatively, the segmentation results are facilitated by suppressing these areas, which are shown in the dice score below each row.

VII. Acknowledgement

This work was partially supported by the Natural Science Foundation of China under contract 62088102 and 62072009, and the Beijing Nova Program (Z201100006820064). We also acknowledge the Clinical Medicine Plus X-Young Scholars Project, and High-Performance Computing Platform of Peking University for providing computational resources.

References

[1] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3d

cnn with fully connected crf for accurate brain lesion segmentation," *Medical image analysis*, vol. 36, pp. 61–78, 2017.

[2] H. J. Lee, J. U. Kim, S. Lee, H. G. Kim, and Y. M. Ro, "Structure boundary preserving segmentation for medical image with ambiguous boundary," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4817–4826, 2020.

[3] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," *arXiv preprint arXiv:1906.05849*, 2019.

[4] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733*, 2020.

[7] J. Li, P. Zhou, C. Xiong, R. Socher, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020.

[8] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *arXiv preprint arXiv:2006.09882*, 2020.

[9] X. Chen and K. He, "Exploring simple siamese representation learning," *arXiv preprint arXiv:2011.10566*, 2020.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[11] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," 2021.

[12] J. Peng, P. Wang, C. Desrosiers, and M. Pedersoli, "Self-paced contrastive learning for semi-supervised medical image segmentation with meta-labels," *arXiv preprint arXiv:2107.13741*, 2021.

[13] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive learning of medical visual representations from paired images and text," *arXiv preprint arXiv:2010.00747*, 2020.

[14] A. Araujo, W. Norris, and J. Sim, "Computing receptive fields of convolutional neural networks," *Distill*, 2019. https://distill.pub/2019/computing-receptive-fields.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[17] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

[18] A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca, "Data augmentation using learned transformations for one-shot medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8543–8553, 2019.

[19] D. Nie, L. Wang, L. Xiang, S. Zhou, E. Adeli, and D. Shen, "Difficulty-aware attention network with confidence learning for medical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1085–1092, 2019.

[20] H. Zheng, Y. Zhang, L. Yang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen, "A new ensemble learning framework for 3d biomedical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5909–5916, 2019.

[21] H. Zheng, Y. Zhang, L. Yang, C. Wang, and D. Z. Chen, "An annotation sparsification strategy for 3d medical image segmentation via representative selection and self-training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6925–6932, 2020.

[22] F. Isensee, Jens, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, *et al.*, "nnU-Net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.

[23] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu, "Contrastive learning of global and local features for medical image segmentation with limited annotations," *arXiv preprint arXiv:2006.10511*, 2020.

[24] M. Wu, C. Zhuang, M. Mosse, D. Yamins, and N. Goodman, "On mutual information in contrastive learning for visual representations," *arXiv preprint arXiv:2005.13149*, 2020.

[25] F. Graf, C. Hofer, M. Niethammer, and R. Kwitt, "Dissecting supervised constrastive learning," in *International Conference on Machine Learning*, pp. 3821–3830, PMLR, 2021.

[26] X. Zhao, R. Vemulapalli, P. Mansfield, B. Gong, B. Green, L. Shapira, and Y. Wu, "Contrastive learning for label-efficient semantic segmentation," 2020.

[27] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *arXiv preprint arXiv:1703.04977*, 2017.

[28] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in *International conference on medical image computing and computer-assisted intervention*, pp. 399–407, Springer, 2017.

[29] S. A. A. Kohl, B. Romera-Paredes, C. Meyer, J. D. Fauw, J. R. Ledsam, K. H. Maier-Hein, S. M. A. Eslami, D. J. Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," 2019.

[30] L. Yu, S. Wang, X. Li, C.-W. Fu, and P.-A. Heng, "Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 605–613, Springer, 2019.

[31] A. Jungo and M. Reyes, "Assessing reliability and challenges of uncertainty estimations for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 48–56, Springer, 2019.

[32] W. Wan, J. Chen, T. Li, Y. Huang, J. Tian, C. Yu, and Y. Xue, "Information entropy based feature pooling for convolutional neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3405–3414, 2019.

[33] J. Gu and C. Dong, "Interpreting super-resolution networks with local attribution maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9199–9208, 2021.

[34] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 447–456, 2015.

[35] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[36] D. Gutman, N. C. Codella, E. Celebi, B. Helba, M. Marchetti, N. Mishra, and A. Halpern, "Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1605.01397*, 2016.

[37] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira, "PH2-a dermoscopic image database for research and benchmarking," in *2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pp. 5437–5440, IEEE, 2013.

[38] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, *et al.*, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)," in *International Symposium on Biomedical Imaging*, pp. 168–172, IEEE, 2018.

[39] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma, "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative imaging in medicine and surgery*, vol. 4, no. 6, p. 475, 2014.

[40] J. Li, S. Yang, X. Huang, Q. Da, X. Yang, Z. Hu, Q. Duan, C. Wang, and H. Li, "Signet ring cell detection with a semi-supervised learning framework," in *International Conference on Information Processing in Medical Imaging*, pp. 842–854, Springer, 2019.

[41] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019.

[42] L. Bi, J. Kim, E. Ahn, D. Feng, and M. Fulham, "Automated skin lesion segmentation via image-wise supervised learning and multi-scale superpixel based cellular automata," in *IEEE International Symposium on Biomedical Imaging*, pp. 1059–1062, IEEE, 2016.

[43] Y. Yuan, M. Chao, and Y.-C. Lo, "Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance," *IEEE transactions on medical imaging*, vol. 36, no. 9, pp. 1876–1886, 2017.

[44] Z. Mirikharaji and G. Hamarneh, "Star shape prior in fully convolutional networks for skin lesion segmentation," in *International Conference*

on Medical Image Computing and Computer-Assisted Intervention, pp. 737–745, Springer, 2018.
[45] E. Ahn, L. Bi, Y. H. Jung, J. Kim, C. Li, M. Fulham, and D. D. Feng, "Automated saliency-based lesion segmentation in dermoscopic images," in *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pp. 3009–3012, IEEE, 2015.
[46] L. Bi, J. Kim, E. Ahn, A. Kumar, M. Fulham, and D. Feng, "Dermoscopic image segmentation via multistage fully convolutional networks," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2065–2074, 2017.
[47] L. Bi, J. Kim, E. Ahn, Ashnil, D. Feng, and M. Fulham, "Step-wise integration of deep class-specific learning for dermoscopic image segmentation," *Pattern recognition*, vol. 85, pp. 78–89, 2019.
[48] Y. Yuan and Y.-C. Lo, "Improving dermoscopic image segmentation with enhanced convolutional-deconvolutional networks," *IEEE journal of biomedical and health informatics*, vol. 23, no. 2, pp. 519–526, 2017.
[49] K. Sarker, H. A. Rashwan, F. Akram, S. F. Banu, A. Saleh, V. K. Singh, F. U. Chowdhury, S. Abdulwahab, S. Romani, P. Radeva, *et al.*, "Slsdeep: Skin lesion segmentation based on dilated residual and pyramid pooling networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 21–29, Springer, 2018.
[50] H. Fu, J. Cheng, Y. Xu, D. W. K. Wong, J. Liu, and X. Cao, "Joint optic disc and cup segmentation based on multi-label deep network and polar transformation," *IEEE transactions on medical imaging*, vol. 37, no. 7, pp. 1597–1605, 2018.
[51] Y. Xie, J. Zhang, Y. Xia, and C. Shen, "A mutual bootstrapping model for automated skin lesion segmentation and classification," *IEEE Transactions on Medical Imaging*, 2020.
[52] H. Chen, X. Qi, L. Yu, and P.-A. Heng, "Dcan: deep contour-aware networks for accurate gland segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2487–2496, 2016.
[53] X. Wang, X. Jiang, H. Ding, and J. Liu, "Bi-directional dermoscopic feature learning and multi-scale consistent decision fusion for skin lesion segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 3039–3051, 2019.
[54] Z. Zhang, H. Fu, H. Dai, J. Shen, Y. Pang, and L. Shao, "Et-net: A generic edge-attention guidance network for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 442–450, Springer, 2019.
[55] T. Estienne, M. Vakalopoulou, S. Christodoulidis, E. Battistela, M. Lerousseau, A. Carre, G. Klausner, R. Sun, C. Robert, S. Mougiakakou, *et al.*, "U-resnet: Ultimate coupling of registration and segmentation with deep nets," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 310–319, Springer, 2019.
[56] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.
[57] S. Kim, I. Kim, S. Lim, W. Baek, C. Kim, H. Cho, B. Yoon, and T. Kim, "Scalable neural architecture search for 3d medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 220–228, Springer, 2019.
[58] J. Xu, M. Li, and Z. Zhu, "Automatic data augmentation for 3d medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 378–387, Springer, 2020.
[59] J. Xu, M. Li, and Z. Zhu, "Automatic data augmentation for 3D medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 378–387, Springer, 2020.
[60] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.
[61] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.
[62] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.