ABSTRACT
Capsule network is a recent new deep network architecture that has been applied successfully for medical image segmentation tasks. This work extends capsule networks for volumetric medical image segmentation with self-supervised learning. To improve on the problem of weight initialization compared to previous capsule networks, we leverage self-supervised learning for capsule networks pre-training, where our pretext-task is optimized by self-reconstruction. Our capsule network, SS-3DCapsNet, has a UNet-based architecture with a 3D Capsule encoder and 3D CNNs decoder. Our experiments on multiple datasets including iSeg-2017, Hippocampus, and Cardiac demonstrate that our 3D capsule network with self-supervised pre-training considerably outperforms previous capsule networks and 3D-UNets. Code is available at here.

Index Terms— Capsule network, self-supervised learning, medical image segmentation, less labeled data

1. INTRODUCTION

Since the introduction of UNet [1, 2], UNet-based neural networks have achieved impressive performance in various modalities of medical image segmentation (MIS), e.g. brain tumor [3, 4, 5], infant brain [6, 7], liver tumor [8], optic disc [9], retina [10], lung [11], and cell [12], etc. Recently, capsule networks [13] have also been applied successfully for MIS [14, 15, 11]. Despite such, there remains a wide range of challenges: (1) Most methods are based on supervised learning, which is prone to many data problems like small-scale data, low-quality annotation, small objects, ambiguous boundaries, to name a few. These problems are not straightforward to overcome: labeling medical data is laborious and expensive, requiring an expert’s domain knowledge. (2) Capsule networks for medical segmentation does not outperform CNNs yet, even though the performance gap gets significantly closer [11].

To address such limitations and inspired by the recent success of capsule networks, in this work, we develop SS-3DCapsNet, a self-supervised capsule network for volumetric MIS. Our SS-3DCapsNet is built upon a state-of-the-art 3D capsule network that leverages both 3D Capsule blocks and CNN blocks for encoder and decoder architecture, respectively, which accounts for temporal relations in volumetric slices in learning contextual visual representation. We introduce self-supervised learning (SSL) to our 3D capsule network, which results in a UNet-like architecture that contains three pathways, i.e., visual representation, encoder, and decoder. The first path consists of dilated convolutional layers, which were pre-trained by SSL techniques. The encoder path is built upon 3D Capsule blocks, whereas the decoder path is built upon 3D CNNs blocks. Compared to 2D-SegCaps [15], which is highly dependent on some random phenomena such as sampling order or weight initialization, our SS-3DCapsNet learns visual representation better as well as having a more robust weight initialization thanks to self-supervised learning. Compared to 3D-UCaps [11], we show that self-supervised learning results in additional gain in segmentation accuracy while keeping the same network complexity at test time.

Our contributions are: (1) An effective self-supervised 3D capsules network for volumetric image segmentation. Our network architecture inherits the merits from 3D Capsule block, 3D CNN blocks, and self-supervised learning for better visual representation learning; and (2) A suite of experiments with ablation studies that empirically demonstrates the effectiveness of self-supervised 3D capsules network for MIS.

2. RELATED WORKS

Medical Segmentation. Among various DL architectures [16, 17], an encoder-decoder like UNet [1] and its extension have achieved impressive performance among semantic segmentation approaches. Since the seminal work of UNet [2] for MIS, there have been numerous subsequent works in this task. As shown in a recent survey [18], MIS can be divided into two main DL groups: supervised learning and weakly supervised learning techniques.

The first group includes CNN-based supervised learning methods such as FCN [19], UNet [20], CC-3D-FCN [21], RLS [22], ACRes [4], DenseVoxNet [23], Flow-based [24], VoxResNet [25], 3D DR-UNet [26], Recurrent Level Set [22], AtrousNet [4], Offset Curves Loss [7, 10], Point-Unet [5] as notable methods. The second group includes weakly supervised learning methods such as transfer learning [27], domain adaptation [28], interactive segmentation [29]. To address the issue of data limitation for training, Generative Adversarial Network
(GAN) [30] has been incorporated into CNNs [31, 32, 35, 24]. Training with imperfect datasets with scarce annotations and weak annotations has also been considered recently [34].

**Capsule Networks.** Capsule networks [35] (CapsNet) is a new network architecture concept that strengthens feature learning by retaining more information at the aggregation layer for pose reasoning and learning the part-whole relationship, which makes it a potential solution for semantic segmentation and object detection tasks. In CapsNet, a capsule aims to represent an entity: capsule norm indicates the probability that entity is present and capsule direction indicates the configuration that entity is in. CapsNet is recently made practical [13] in a CNN that incorporates two layers of capsules with dynamic routing.

While most CapsNet has been proposed for image classification, SegCaps [14, 15] expanded CapsNet for object segmentation. This method functions by treating an MRI image as a collection of slices, each of which is then encoded and decoded by capsules to output the segmentation. However, SegCaps is mainly designed for 2D still images, and it performs poorly when being applied to volumetric data because of missing temporal information. 3D-UCaps [11] is a hybrid network architecture that utilizes both capsules and deconvolutions for feature learning and segmentation output, respectively, which shows that such combination can outperform SegCaps design significantly in the segmentation task while retaining the merits of capsules. Our method further improves upon 3D-UCaps by integrating an efficient pre-training stage.

**Self-supervised Learning.** Self-supervised learning (SSL) is a technique for learning feature representation in a network without requiring a labeled dataset. A common workflow to apply SSL is to train the network in an unsupervised manner by learning with a pretext task in the pre-training stage, and then finetuning the pre-trained network on a target downstream task. In the case of MIS, the suitable pretext tasks can be considered in four categories: context-based, generation-based, free semantic label-based, and cross-modal-based. The first techniques utilize context features of images or videos such as context similarity [36], spatial structure [37], temporal structure [38]. The second techniques have been used in image generation [39] and video generation [40]. The third techniques aim to automatically generate semantic labels and applied into segmentation [41], contour detection [41]. The fourth techniques are applied to multiple modalities data such as visual-audio [42], RGB-Flow [43]. In this work, our pretext task is based on image reconstruction.

3. **SS-3DCapsNet: Self-supervised 3D Capsule Networks**

We draw on the ideas of SegCaps [14] and 3D-UCaps [11] to build our 3D capsule network for the medical segmentation task. Particularly, our network has three stages: (i) Visual representation, (ii) Capsule encoder, and (iii) Convolutional decoder as follows.

(i) **Visual Representation:** This stage is for converting the input to a feature volume that can be consumed by the capsule encoder. Followed the concurrent work, we use three dilated convolution layers with 16, 32, 64 channels, respectively. The kernel size is set to $5 \times 5 \times 5$, with dilate rates of 1, 3, and 3, respectively. The size of the visual features is $H \times W \times D \times 64$.

(ii) **Capsule Encoder:** In this stage, we reshape the feature volume into $H \times W \times D$ capsules, where each capsule is represented by a 64-dimensional vector. Here we consider both spatial and temporal data by using our 3D convolutional capsules to learn a richer representation. The output from a convolution capsule has the shape $H \times W \times D \times C \times A$, where $C$ is the number of capsule types and $A$ is the dimension of each capsule. We follow the concurrent work and set $C$ to (16, 16, 16, 8, 8, 8) for each layer in the capsule encoder, respectively. Note that as the number of capsule types in the last convolutional capsule layer is equal to the number of class labels, we can further supervise this particular layer with a margin loss [13].

(iii) **Convolutional Decoder:** This is the final stage in our network. Here we use the decoder of 3D U-Net [2] which includes deconvolution, skip connection, convolution and BatchNorm layers [44] to generate the segmentation from features learned by capsule layers. Particularly, we reshape the capsules back to tensors of size $H \times W \times D \times (C \times A)$ and pass them to the decoder. The overall architecture can be seen in Fig. 1.

### 3.1. Pretext Task

Our pretext task is self-supervised based on medical image reconstruction. In computer vision, it is common to use pseudo-
Fig. 2. Examples of six transformations for self-supervised learning. (a): original image. (b) from left to right, top to bottom: zeros-green-channel, zeros-red-channel, zeros-blue-channel, swapping (4 swapped patches are shown in yellow boxes), blurring, noisy.

Fig. 3. Our pretext task with reconstruction loss.

Our pretext task procedure is illustrated in Fig. 3. Note that the trivial case can occur when training this pretext task. More concretely, the $F$ function can map any volumetric input data into a constant (e.g., 0). Therefore, the pre-trained model from the pretext task may have no impact on the downstream ones. Our pretext task training is probably stuck at its first local minima so that, luckily, the trivial case cannot occur. Unfortunately, in our accepted version at ISBI2022, we overlooked this issue, so we want to clarify it in this version.

3.2. Downstream Task

After pre-training, we train our SS-3DCapsNet network with annotated data on the medical segmentation task. The total loss function to train this downstream task is a sum of three losses:

$$L_{downstream} = L_{margin} + L_{CE} + L_{reconstruction}. \quad (2)$$

The margin loss is adopted from [13] and it is defined between the predicted label $y$ and the ground truth label $y^*$ as follows:

$$L_{margin} = \begin{cases} 0 & \text{if } y^* \leq 0 \\ y^* \times (\max(0, 0.9 - y))^2 & \text{otherwise} \end{cases} + 0.5 \times (1 - y^*) \times (\max(0, y - 0.1))^2. \quad (3)$$

Particularly, we compute the margin loss ($L_{margin}$) on the capsule encoder output with downsampled ground truth segmentation. We compute the weighted cross-entropy loss ($L_{CE}$) on the convolutional decoder. We also regularize the training with a network branch that aims at reconstructing the original input with masked mean-squared errors ($L_{reconstruction}$) [13][14].

4. EXPERIMENTAL RESULTS

4.1. Implementation Details

We conduct our experiments and comparisons on iSeg [45], Hippocampus, and Cardiac [46] datasets. For iSeg, we follow 3D-SkipDenseSeg [47] to have the training set of 9 subjects and testing set of subject #9. On Hippocampus, and Cardiac [46], the experiments are conducted by 4-fold cross-validation.

Table 1. Comparison on iSeg-2017. 1st group: 3D CNN-based networks and 2nd group: Capsule-based networks.

| Method                  | Depth | Dice Score |
|-------------------------|-------|------------|
|                         |       | WM | GM | CSF | Average |
| Qamar et al. [45]       | 82    | 90.50 | 92.05 | 95.80 | 92.77 |
| 3D-SkipDenseSeg [47]    | 47    | 91.02 | 91.64 | 94.88 | 92.51 |
| VoxResNet [25]          | 25    | 89.87 | 90.64 | 94.28 | 91.60 |
| 3D-U-Net [21]           | 18    | 89.83 | 90.55 | 94.39 | 91.59 |
| CC-3D-FCN [21]          | 34    | 89.19 | 90.74 | 92.40 | 90.79 |
| DenseVoxNet [23]        | 32    | 85.46 | 88.51 | 91.26 | 89.24 |
| 2D SegCaps [13]         | 16    | 82.80 | 84.19 | 90.19 | 85.73 |
| 3D-SegCaps [17]         | 16    | 86.49 | 88.53 | 93.62 | 89.55 |
| 3D-UCaps [11]           | 17    | 90.21 | 91.12 | 94.93 | 92.08 |
| Our SS-3DCapsNet        | 17    | 90.78 | 91.48 | 94.92 | 92.39 |

We implemented our method in Pytorch. We used patch size of $64 \times 64 \times 64$ for iSeg and Hippocampus whereas patch size of $128 \times 128 \times 128$ on Cardiac. Our SS-3DCapsNet was trained without any data augmentation. We used Adam
Table 2. Comparison on Cardiac with 4-fold cross validation.

| Method               | Recall | Precision | Dice  |
|----------------------|--------|-----------|-------|
| Multi-SegCaps (2D)   | 80.76  | 65.65     | 72.42 |
| 3D UCaps [11]        | 81.70  | 80.19     | 80.99 |
| Our SS-3DCapsNet     | 81.84  | 81.49     | 81.59 |

Table 3. Comparison on Hippocampus with 4-fold.

| Method               | Anterior Recall | Anterior Precision | Anterior Dice | Posterior Recall | Posterior Precision | Posterior Dice |
|----------------------|-----------------|--------------------|---------------|------------------|---------------------|----------------|
| Multi-SegCaps (2D)   | 80.76           | 65.65              | 72.42         | 84.46            | 60.49               | 70.49          |
| 3D UCaps [11]        | 81.70           | 80.19              | 80.99         | 80.21            | 79.25               | 79.48          |
| Our SS-3DCapsNet     | 81.84           | 81.49              | 81.59         | 80.71            | 80.21               | 79.97          |

Table 4. Performance of SS-3DCapsNet on iSeg with different network configurations.

| Method                          | WM          | GM          | CSF          | Average      |
|--------------------------------|-------------|-------------|--------------|--------------|
| change number of capsule (set to 4) | 89.02       | 89.78       | 89.95        | 89.58        |
| w/o visual representation       | 89.15       | 89.66       | 90.82        | 89.89        |
| w/o margin loss                  | 87.62       | 88.85       | 92.06        | 89.51        |
| w/o reconstruction loss          | 88.50       | 88.96       | 90.18        | 89.22        |
| w/o pretext task                 | 90.21       | 91.12       | 94.93        | 92.08        |
| SS-3DCapsNet                    | 90.78       | 91.48       | 94.92        | 92.39        |

Table 5. Performance of SS-3DCapsNet on Precision (Pre), Recall (Rec) and Dice score (DSC) with and without pretext task on various datasets.

| Method               | iSeg          | Hippocampus  | Cardiac       |
|----------------------|---------------|--------------|---------------|
|                      | Pre | Rec | DSC | Pre | Rec | DSC | Pre | Rec | DSC |
| w/o. SSL             | 92.28 | 91.29 | 92.08 | 79.72 | 80.95 | 80.24 | 84.60 | 95.06 | 89.69 |
| w. SS                | 92.54 | 92.37 | 92.39 | 80.85 | 81.27 | 80.78 | 86.24 | 94.21 | 89.77 |

optimizer with an initial learning rate of 0.0001. The learning rate is decayed by 0.05 if the Dice score on the validation set does not increase for 50,000 iterations. Early stopping is set at 250,000 iterations as in [14].

4.2. Performance and Comparison We compare our SS-3DCapsNet with both SOTA 3D CNNs-based and Capsule-based segmentation methods. 3D-Ucaps [11] has two versions of with and without utilizing MONAI [51]. To conduct a fair comparison, we report the version without MONAI.

The comparison between our proposed SS-3DCapsNet with SOTA segmentation approaches on iSeg dataset [45] is given in Table 1. As can be seen, 3D capsule networks (3D-SegCaps, SS-3DCapsNet) outperform 2D-SegCaps by a wide margin. This performance gap can be explained by the combination of pre-training, Capsule encoder, and Convolutional decoder in SS-3DCapsNet. Our SS-3DCapsNet also outperforms 3D-SegCaps, which contains only a Capsule-based encoder and decoder. Our SS-3DCapsNet also performs comparably to SOTA 3D CNNs, but our network is significantly shallower (17 layers vs. 82 layers in [48]). Our network also has fewer parameters and a better Dice score when compared to SOTA 3D CNNs with similar number of layers, e.g. 3D-UNets [2] (18 layers). In addition to iSeg, we also evaluate our SS-3DCapsNet on Hippocampus and Cardiac, where the results are shown in Table 2 and Table 3.

4.3. Ablation Study We analyze the performance of our method as follows.

i. Network Configuration: We trained SS-3DCapsNet under various settings as shown in Table 4. By following the concurrent work on 3D capsule networks, we use a baseline where the number of capsules of the first layer is reduced to 4 (similar to SegCaps). As can be seen, each component including visual representation, margin loss, reconstruction loss, pre-training contributes to the final performance. Removing any of such components would result in performance drops.

ii. SSL Contribution: We perform experiments on various datasets and turn on/off the self-supervision step in the experiments. The results in Table 5 clearly shows that pre-training plays an important role in our method, which improves the Dice score considerably in iSeg, and slightly in other datasets.

CONCLUSION

In this work, we proposed a capsule network for MIS powered with self-supervised pre-training. Our SS-3DCapsNet can both utilize self-supervised learning and 3D capsules for learning features while retaining the advantage of traditional convolutions in decoding the segmentation results. Even though we use capsules with dynamic routing only in the encoder of a simple Unet-like architecture, we can achieve the competitive result with the SOTA models on iSeg-2017 challenge while outperforming SegCaps [13] on different complex datasets with less labeled annotated data. Future work includes exploring different self-supervised learning methods such as SimCLR [52] for better feature learning and representation.

Acknowledgment: This material is based upon work supported by the National Science Foundation under Award No. OIA-1946391 and NSF Track-2 WVAR-CRESH.

5. REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in MICCAI, pp. 234–241, Springer, 2015.
[2] O. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3d u-net: learning dense volumetric segmentation from sparse annotation,” in MICCAI, pp. 424–432, Springer, 2016.
[3] L. Ngan, R. Gummadi, and M. Savvides, “Deep recurrent level set for segmenting brain tumors,” MICCAI, pp. 646–653, 2018.
[4] N. Le, K. Yamazaki, K. G. Quach, D. Truong, and M. Savvides, “A multi-task contextual atrous residual network for brain tumor detection & segmentation,” in ICPR, pp. 5943–5950, IEEE, 2021.
[5] N.-V. Ho, T. Nguyen, G.-H. Diep, N. Le, and B.-S. Hua, “Point-unet: A context-aware point-based neural network for volumetric segmentation,” in MICCAI, pp. 644–655, 2021.
