# An Annotation Sparsification Strategy for 3D Medical Image Segmentation via Representative Selection and Self-Training

**Hao Zheng, Yizhe Zhang, Lin Yang, Chaoli Wang, Danny Z. Chen**
Department of Computer Science and Engineering, University of Notre Dame
Notre Dame, IN 46556, USA
{hzheng3, yzhang29, lyang5, chaoli.wang, dchen}@nd.edu

## Abstract

Image segmentation is critical to lots of medical applications. While deep learning (DL) methods continue to improve performance for many medical image segmentation tasks, data annotation is a big bottleneck to DL-based segmentation because (1) DL models tend to need a large amount of labeled data to train, and (2) it is highly time-consuming and label-intensive to voxel-wise label 3D medical images. Significantly reducing annotation effort while attaining good performance of DL segmentation models remains a major challenge. In our preliminary experiments, we observe that, using partially labeled datasets, there is indeed a large performance gap with respect to using fully annotated training datasets. In this paper, we propose a new DL framework for reducing annotation effort and bridging the gap between full annotation and sparse annotation in 3D medical image segmentation. We achieve this by (i) selecting representative slices in 3D images that minimize data redundancy and save annotation effort, and (ii) self-training with pseudo-labels automatically generated from the base-models trained using the selected annotated slices. Extensive experiments using two public datasets (the HVSMR 2016 Challenge dataset and mouse piriform cortex dataset) show that our framework yields competitive segmentation results comparing with state-of-the-art DL methods using less than $\sim 20\%$ of annotated data.

## Introduction

3D image segmentation is one of the most important tasks in medical image applications, such as morphological and pathological analysis (Lee et al. 2015b; Hou et al. 2019), disease diagnosis (Pace et al. 2015), and surgical planning (Kordon et al. 2019). Recently, 3D deep learning (DL) models have been widely used in medical image segmentation and achieved state-of-the-art performance (Ronneberger, Fischer, and Brox 2015; Yu et al. 2017; Liang et al. 2019), most of which were trained with fully annotated 3D image stacks. The performance of DL models (when applied to testing images) is highly dependant on the amount and variety of labeled data used in model training. However, obtaining medical image annotation data is highly difficult and expensive, and full annotation of 3D medical images is a monotonous, labor-intensive, and time-consuming job. For example, a typical 3D abdominal CT scan is of size $300 \times 512 \times 512$, and would take hours of a medical expert to label certain objects of interest in it. How to reduce annotation effort (e.g., cost, time, and available experts) while attaining the best possible performance of DL models remains a challenging problem for 3D medical image segmentation.



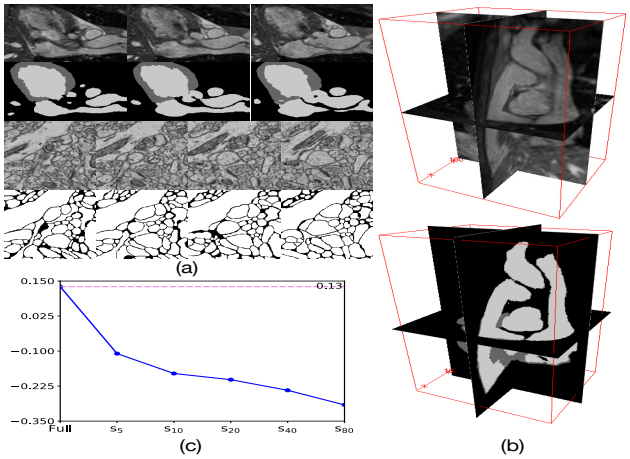


Figure 1: (a) Examples showing similarity in consecutive slices of the HVSMR 2016 heart dataset and of the neuron dataset of mouse piriform cortex. (b) Sparse annotation in a 3D image (top: image, bottom: annotation); only selected slices are manually annotated to train deep learning models. (c) Performance on the HVSMR 2016 dataset using different amounts of annotated training data. Let $\mathrm{s}_k$ denote the setting of selecting slices at an equal distance (i.e., label one out of every $k$ slices). The segmentation performance drops drastically as the annotation ratio $\mathrm{s}_k$ decreases.

A common method to alleviate annotation burden is *sparse 3D fully convolutional networks (FCNs)* (Çiçek et al. 2016). As shown in Fig. 1(a), there can be a great deal of redundancy in consecutive 2D slices along an axis of a 3D image, and it is unnecessary to annotate each and every one of them. (Çiçek et al. 2016) showed that a small number of annotated 2D slices could be used as supervision (see Fig. 1(b))

to train a 3D FCN, and satisfactory segmentation performance was obtained. Compared with conventional 3D FCN models, when calculating the loss, sparse 3D FCN models take only annotated voxels into consideration and perform back-propagation to optimize the networks. However, there are two major issues. (1) The more sparsely one annotates the data, the worse the performance becomes. In our preliminary experiments, we use *equal-interval annotation* (EIA) as a baseline. Although unseen testing stacks can be segmented during inference, the performance decreases drastically if fewer slices are annotated compared with FCNs trained with full annotation (see Fig. 1(c)). (2) Which slices are most valuable for annotation? This is not well addressed. A subset of selected slices should be both *informative* and *diverse* so that the subset would cover typical patterns/topology of 3D objects and reduce redundancy. Although a series of sample selection based methods (Yang et al. 2017; Zhou et al. 2017; Zheng et al. 2019a) were proposed to deal with 2D image segmentation, for 3D images, this is not well studied.

Another line of related approaches is based on semi-supervised learning (SSL) (Zhang et al. 2017; Zhou et al. 2019), where abundant and easily-obtainable unannotated data are utilized for training to boost performance. However, the focus of conventional SSL-based methods is somewhat different from our goal to reduce annotation effort: SSL has an underlying assumption that annotated data should be representative enough to cover the true data distribution, but which data samples should be selected for annotation is neglected in previous work. Besides, selected 3D stacks still need dense voxel-wise annotation. Our aim is complementary to SSL-based approaches; we can further reduce annotation effort, and SSL could in turn improve performance by adding more unannotated data in a later stage.

In this paper, we propose a new framework to adapt an annotation sparsification strategy into semi-supervised segmentation. For an unannotated 3D image, we select effective slices with *high influence and diversity* using a representative selection algorithm, which allows a considerable relief of manual annotation. Then we train light-weight networks using sparsely annotated data to perform segmentation on the remaining, unannotated slices and obtain pseudo-labels, which fills the annotation gap in the 3D image. Finally, we use these pseudo-labels as *dense* supervision to conduct self-training with the original training data. To achieve this goal, we need to address three vital challenges: (1) How to provide useful clues about the most influential and diverse slices for manual annotation? (2) How to make the most out of the sparse annotation and generate high quality pseudo-labels? (3) How to conduct self-training using dense pseudo-labels?

For the first challenge, we leverage a pre-trained network to extract image features, and devise a max-cover based method to select the most representative slices. For the second challenge, we observe that the generated pseudo-labels (PLs) by an FCN with sparse annotation contain noise, and different types of FCNs possess different characteristics. For example, inferred PLs from 2D FCNs along the three axes may be inconsistent with one another, but 2D FCNs have a quite large field of view thus large structures could be recognized. In contrast, inferred PLs from 3D FCNs are much smoother since 3D image information could be utilized, but some regions-of-interest may be missing due to their limited field of view. Hence, we adopt the predictions of both 2D and 3D FCNs as supervision for better knowledge distillation. Such heterogeneous predictions are likely to get closer to the correct labels of unannotated slices, and thus the performance gap can be reduced accordingly. For the third challenge, we utilize a self-training based network to combine the merits of multiple sets of PLs, which offers the benefits of weakening noisy labels and reducing over-fitting.

In summary, our contribution in this work is three-fold. (a) We propose a new training strategy based on representative slice selection and self-training for 3D medical image segmentation. (b) The most representative slices are selected for manual annotation, thus saving annotation effort. (c) Self-training using heterogeneous pseudo-labels bridges the performance gap with respect to full annotation. Extensive experiments show that using only less than $20\%$ annotated slices, our model achieves comparative results as fully-supervised methods.

## A Brief Review of Related DL Techniques

**3D Medical Image Segmentation.** An array of 2D (Ronneberger, Fischer, and Brox 2015; Wolterink et al. 2017; Shen et al. 2017) and 3D (Çiçek et al. 2016; Yu et al. 2017; Liang et al. 2019; Zheng et al. 2019b) FCNs has been developed that significantly improved segmentation performance on various 3D medical image datasets (Pace et al. 2015; Shen et al. 2017). Scale-level (Ronneberger, Fischer, and Brox 2015) and block-level (He et al. 2016; Huang et al. 2017) skip-connections allow substantially deeper architecture design and ease the training by alleviating the vanishing gradient problem. Other advances such as batch normalization (Ioffe and Szegedy 2015) and deep supervision (Lee et al. 2015a) also help network training and optimization. In this study, we utilize these advanced techniques in our 2D and 3D FCNs for segmentation.

**Sparse Medical Image Annotation.** Sparse annotation was not well addressed in medical image segmentation until recently. Where to annotate and how to utilize sparse annotation for training are two basic issues. Active learning (AL) based frameworks (Yang et al. 2017; Zhou et al. 2017) reduced annotation effort by incrementally selecting the most informative samples from unlabeled sets and querying human experts for annotation iteratively. Recently, (Zheng et al. 2019a) decoupled these two iterative steps in AL frameworks by applying unsupervised networks to encode input samples and extract latent vectors, and ordering the samples based on their representativeness in one-shot, achieving competitive performance. These approaches succeeded in dealing with 2D images because repeated patterns appear over and over again (e.g., cells, glands, etc), but are not potent enough for a large portion of 3D image datasets which have more complex object topology and fewer samples (see Fig. 1(a)). A pioneer work (Çiçek et al. 2016) shed some light on sparse 3D FCN training using 2D annotated slices and yielded good performance. Our framework combines these previous methods to address the two basic issues for sparse annotation to obtain good segmentation performance.

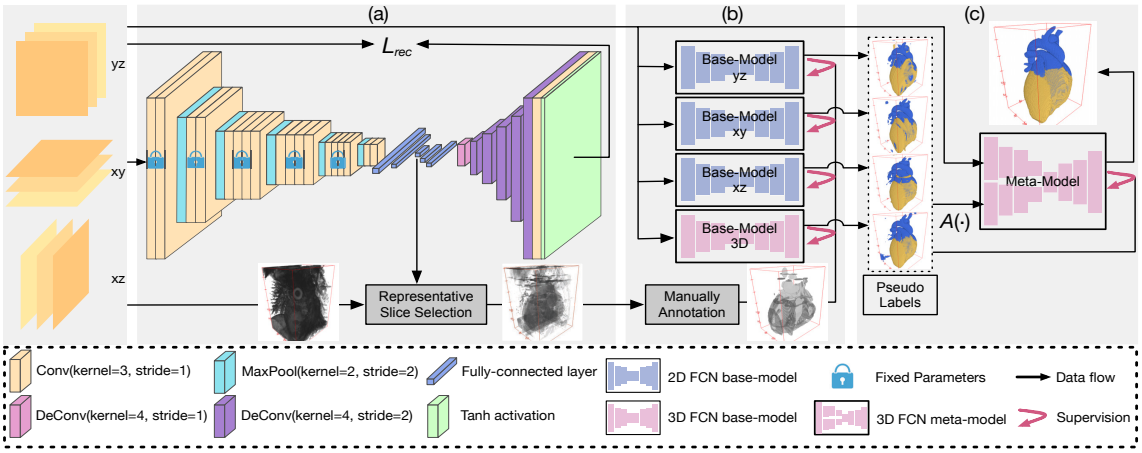


Figure 2: An overview of our proposed framework. (a) Representative slice selection. (b) Manual annotation and Pseudo-label (PL) generation from the base-models using sparse annotation. (c) Meta-model training using PLs.

**Weakly-/Semi-Supervised Learning.** Weakly-supervised learning (WSL) based methods explore various weak annotation forms (e.g., points (Bearman et al. 2016), scribbles (Lin et al. 2016), and bounding boxes (Khoreva et al. 2017; Zhao et al. 2018; Yang et al. 2018)). But, none of them is suitable for a large portion of 3D medical images. For example, not all cardiovascular substructures are convex and an object could be wrapped by another (e.g., myocardium and blood pool in Fig. 1(a)), or objects are closely packed and are in arbitrary orientation (e.g., neuron cells in Fig. 1(a)). Semi-supervised learning (SSL) based methods exploit additional unannotated images to improve segmentation performance. The self-training approach is the earliest SSL one and recently became popular in DL schemes (Zhang et al. 2017; Radosavovic et al. 2018). It uses the predictions of a model on unlabeled data to re-train the model itself iteratively. Another array of work is based on multi-view learning (Blum and Mitchell 1998) which splits a dataset based on different attributes and utilizes the agreement among different learners. (Zhou et al. 2019) incorporated multi-view learning using multi-view properties of 3D medical data to achieve better performance. However, a major limitation of WSL/SSL based approaches is that they still require annotation of a certain amount of *full* 3D stacks.

We embed a new annotation sparsification strategy into the self-training scheme to address the problem. It further makes use of the underlying assumptions of self-training: the independent and identical distribution of labeled and unlabeled data, and the smoothness of manifold in high-dimensions (Niyogi 2013). Consequently, sparse annotation in each 3D stack would produce accurate pseudo-labels.

## Methodology

We propose a new annotation sparsification approach which saves considerable annotation effort via representative slice selection from each 3D stack and improves segmentation performance via self-training using pseudo-labels (PLs).

**Problem Formulation**: Under the fully-supervised setting, given a set of 3D images, $X = \{\mathcal{X}_i\}_{i=1}^m$, and their corresponding ground-truth $Y = \{\mathcal{Y}_i\}_{i=1}^m$, consider a 3D image $\mathcal{X}_i \in \mathbb{R}^{W\times H\times D}$ with its associated ground-truth $C$-class segmentation masks, $\mathcal{Y}_i \in \{1, 2, \ldots, C\}^{W\times H\times D}$, where $W$, $H$, and $D$ are the numbers of voxels along the $x$-, $y$-, and $z$-axis of $\mathcal{X}_i$ respectively and $\mathcal{Y}_i^{(w,h,d)} = [\mathcal{Y}_i^{(w,h,d,c)}]_c$ provides the label of voxel $(w, h, d)$ as a one-hot vector.

Conventionally, when training a 2D FCN, we can split a 3D volume $\mathcal{X}_i$ along an orthogonal direction. For example, $\{\mathcal{X}_i^V = \{\mathbf{I}_{i,n}^V\}_{n=1}^{N_V}\}_{V\in\{xy,xz,yz\}}$, where $N_V$ is the number of 2D slices obtained from plane $V$ and $\mathbf{I}_{i,n}^V$ is a 2D slice from plane $V$ (e.g., $\mathbf{I}_{i,n}^{xy} \subset \mathbf{R}^{W\times H}$ and $N_V = D$ if $V = xy$). Similarly, $\{\mathcal{Y}_i^V = \{\mathbf{Y}_{i,n}^V\}_{n=1}^{N_V}\}_{V\in\{xy,xz,yz\}}$. If the 3D data are approximate-isotropic, we can split each volume in the $xy$, $xz$, and $yz$ planes respectively, and get three sets of 2D slices. Each set $S = \{(\mathbf{I}_\ell, \mathbf{Y}_\ell)\}_{\ell=1}^L$, where $L$ is the total number of slices. The goal of segmentation is to design a function $\mathcal{H}$ so that $\hat{\mathbf{Y}}_\ell = \mathcal{H}(\mathbf{I}_\ell)$ is close to $\mathbf{Y}_\ell$. The parameters $\theta_\mathcal{H}$ of $\mathcal{H}$ are learned to minimize the segmentation loss $\mathcal{L}_{seg}(\mathbf{I}_\ell, \mathbf{Y}_\ell) = -\sum \mathbf{Y}_\ell \log \hat{\mathbf{Y}}_\ell$ on the whole set $S$. Under the sparse annotation setting, only a subset $S' \subseteq S$ is annotated, and the objective is:

$$\min_{\theta_\mathcal{H}} \frac{1}{|S'|} \sum_{\mathbf{I}_\ell \in S'} \mathcal{L}_{seg}(\mathbf{I}_\ell, \mathbf{Y}_\ell) \tag{1}$$

When training a 3D FCN, the parameters $\theta_\mathcal{H}$ are optimized by minimizing the loss $\mathcal{L}_{seg}(\mathcal{X}_i, \mathcal{Y}_i) = -\sum \mathcal{Y}_i \log \hat{\mathcal{Y}}_i$ over the whole set $\{(\mathcal{X}_i, \mathcal{Y}_i)\}_{i=1}^m$. Under the sparse annotation setting, only a part of all the voxels is annotated. Following (Çiçek et al. 2016), the objective function is:

$$\min_{\theta_\mathcal{H}} \frac{1}{|\mathcal{M}(X)|} \sum_{\mathcal{X}_i \in X} \mathcal{L}_{seg}(\mathcal{X}_i, \mathcal{Y}_i) \cdot \mathcal{M}(\mathcal{X}_i) \tag{2}$$

where $\mathcal{M}(\mathcal{X}_i) = \mathbb{1}_{\Delta(v)}$ and $\Delta(v) = 1$ if and only if a voxel $v$ in $\mathcal{X}_i$ is annotated (otherwise, $\Delta(v) = 0$). Similarly, it is for $\mathcal{M}(X)$ in the dataset. As shown in Fig. 2, our proposed approach consists of three steps:

- **Step I:** Representative Slice Selection. Pre-train an auto-encoder (*AE*) using $\{\mathcal{X}_i^V\}_{i=1}^m$, and extract the compressed vector from *AE* as the feature vector of each input 2D slice $\mathbf{I}_{i,n}^V$. Select image slices according to their representativeness captured by the feature vectors.
- **Step II:** Pseudo-Label (PL) Generation. Train 2D and 3D base-models by Eq. (1) and Eq. (2) using sparsely annotated 2D slices. The trained base-models are applied to $\{\mathcal{X}_i\}_{i=1}^m$ to get corresponding PLs $\{\hat{\mathcal{Y}}_i^V\}_{V\in\{xy,xz,yz,3D\}}$.
- **Step III:** FCN self-training. A 3D FCN is trained with noisy PLs to learn from multiple-views of the 3D medical images.

## Representative Selection

Intuitively, one could annotate 3D images by a *sub-volume based* method or a *slice based* method. The former method could be impractical in real-world applications for several reasons: (1) human can only annotate 2D slices well; (2) even if a sub-volume is selected, experts have to choose a certain plane (e.g., the $xy$, $xz$, or $yz$ plane) and annotate consecutive 2D slices one by one, where a lot of redundancy may exist (e.g., see Fig. 1(a)). The latter method, proposed in (Çiçek et al. 2016), trains a *sparse 3D FCN model* with some annotated 2D slices, which is more practical and expert-friendly. Considering that regions-of-interest have various topology shapes and feature patterns in different views of 3D data, we hence propose to select some 2D slices from each orthogonal plane for manual annotation.

**Feature Extractor with a Pre-trained VGG-19.** Auto-encoder (AE) can be used to learn efficient data encoding in an unsupervised manner (Rumelhart, Hinton, and Williams 1986). It consists of two sub-networks: an *encoder* that takes an input sample $\boldsymbol{x}$ and compresses it into a latent representation $\boldsymbol{z}$, and a *decoder* that reconstructs the sample from the latent representation back to the original space.

$$\boldsymbol{z} \sim Enc(\boldsymbol{x}) = q_\phi(\boldsymbol{z}|\boldsymbol{x}),\ \tilde{\boldsymbol{x}} \sim Dec(\boldsymbol{z}) = p_\psi(\boldsymbol{x}|\boldsymbol{z}) \quad (3)$$

where $\{\phi, \psi\}$ are network parameters and the optimization objective is to minimize the reconstruction loss, $\mathcal{L}_{rec}$, on the given dataset $X$:

$$\psi^*, \phi^* = \arg\min_{\psi,\phi} \mathcal{L}_{rec}(\boldsymbol{x},\ (\phi \circ \psi)\boldsymbol{x}). \quad (4)$$

To accelerate the training process and extract rich features, in our implementation, we use the VGG-19 (Simonyan and Zisserman 2014) model pre-trained on ImageNet (Deng et al. 2009) as the backbone network. To further facilitate the customized dataset, we fine-tune the model with our medical images. More specifically, we tile a few fully-connected (FC) layers to the last convolution layer of the VGG-19 network, and add a light-weight decoder to form an AE. The parameters of the convolution layers of the VGG-19 are fixed, and the remaining network is fine-tuned with the combination of images from the three orthogonal planes.

**Representative Slice Selection.** Having trained the feature extractor, we feed an image $I$ to the *encoder* model, and the output feature vector, $I^f$, of the last FC layer can be viewed as a high-level representation of the image $I$. We can measure the similarity between two images $I_i$ and $I_j$ as:

$$\text{sim}(I_i, I_j) = Cosine\_similarity(I_i^f, I_j^f) \quad (5)$$

To measure the representativeness of a set $S_x$ of images for a single image $I$ in another set $S_y$, we define:

$$f(S_x, I) = \max_{I_i \in S_x} \text{sim}(I_i, I) \quad (6)$$

It means $I$ is represented by its most similar image $I_i$ in $S_x$.

In our scenario, we need to find a subset $S_i^V$ of slices from every 3D stack along each plane (i.e., $S_i^V \subset \mathcal{X}_i^V = \{\mathbf{I}_{i,n}^V\}_{n=1}^{N_V}$, where $V \in \{xy, xz, yz\}$) such that $S_i^V$ is the most representative for the corresponding $\mathcal{X}_i^V$. To measure how representative $S_i^V$ is for $\mathcal{X}_i^V$, we define the coverage score of $S_i^V$ for $\mathcal{X}_i^V$ as:

$$F(S_i^V, \mathcal{X}_i^V) = \sum_{I_j \in \mathcal{X}_i^V} f(S_i^V, I_j) \quad (7)$$

This forms a maximum set cover problem which is known to be NP-hard. Its best possible polynomial time approximation solution is based on a greedy method with an approximation ratio $1 - \frac{1}{e}$ (Hochbaum 1997). Therefore, we iteratively choose one image slice from $\mathcal{X}_i^V$ and put it into $S_i^V$:

$$I^* = \arg\max_{I \in \mathcal{X}_i^V \setminus S_i^V} (F(S_i^V \cup \{I\}, \mathcal{X}_i^V) - F(S_i^V, \mathcal{X}_i^V)) \quad (8)$$

This selection process essentially sorts the image slices in $\mathcal{X}_i^V$ based on their representativeness decreasingly. We record the order of the selected slices. The better representative slices have higher priorities for manual annotation.

Under the *equal-interval annotation* (EIA) setting, we select slices at an equal distance, i.e., labeling one out of every $k$ slices, denoted by $s_k$. The number of EIA-selected slices along the $z$-axis is $K = \lfloor D/s_k \rfloor$, where $D$ is the number of voxels along the $z$-axis. Given the same annotation budget, $s_k$, in our *representative annotation* (RA) setting, we select the $K$ most representative slices along the $z$-axis.

## Pseudo-Label Generation

After obtaining sparse annotation from human experts, following (Çiçek et al. 2016), we can train a sparse 3D FCN by Eq. (2). Although 3D FCNs can better utilize 3D image information, they adopt a sliding-window strategy to avoid the out of memory problem, thus having a relatively small field of view. Compared with 3D FCNs, 2D FCNs take 2D images as input and can be much deeper and have a larger field of view using the same amount of computational resources. Hence, we propose to utilize 2D FCNs as well (by Eq. (1)), which make the most out of multiple sets of 2D slices to capture heterogeneous features from different views of 3D data. Naturally, we can train three 2D FCNs on three sets of 2D slices separately. The drawbacks are: (1) multiple versions of 2D models are trained, and (2) each 2D model only observes the 3D volume from a specific view and does not explore full geometric distribution of the 3D data. Thus, we treat the three 2D slice

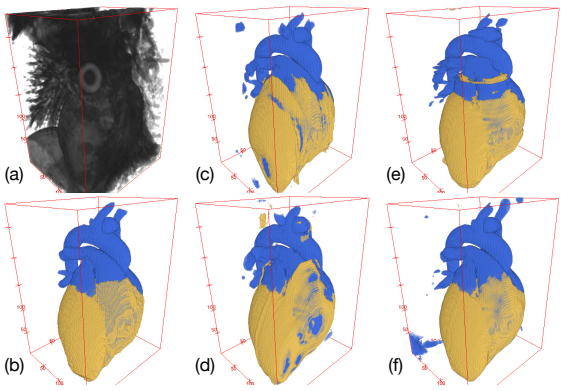


Figure 3: Pseudo-labels generated with an annotation budget $s_{20}$. (a) A raw image $\mathcal{X}_1$; (b) manual annotation $\mathcal{Y}_1$; (c)-(f) $\{\hat{\mathcal{Y}}_1^V\}_{V\in\{xy,xz,yz,3D\}}$, respectively.

sets $\{\{\mathcal{X}_i^V\}_{V\in\{xy,xz,yz\}}\}_{i=1}^m$ equally. In each forward pass of a 2D FCN model, it randomly chooses a stack $\mathcal{X}_i$ and a plane $V$, and crops a patch from a slice as input. This resembles data augmentation that forces the 2D model to learn more from the 3D data. During inference, we apply the trained 2D FCNs to all the sets of 2D slices respectively, and obtain three sets of predictions in the three orthogonal directions respectively, i.e., $\{\{\hat{\mathcal{Y}}_i^V\}_{V\in\{xy,xz,yz\}}\}_{i=1}^m$. Besides, the trained sparse 3D FCN can produce the fourth set of predictions, $\{\hat{\mathcal{Y}}_i^{3D}\}_{i=1}^m$. We use all these as pseudo-labels (PLs) for the next step. As shown in Fig. 3, PLs generated with sparse annotation contain noise, and different types of FCNs possess different characteristics: PLs from the 2D FCNs are inconsistent in the third orthogonal direction, but more structures could be recognized; PLs from the 3D FCN are much smoother, but some regions-of-interest may be missing.

## Self-Training with Pseudo-Labels

In the previous steps, we obtain four sets of PLs, $\hat{Y} = \{\{\hat{\mathcal{Y}}_i^V\}_{V\in\{xy,xz,yz,3D\}}\}_{i=1}^m$ for the training set $X = \{\mathcal{X}_i\}_{i=1}^m$. Here we aim to train a meta-model that summarizes the noisy PLs and attains better prediction accuracy.

Following the practice in (Zheng et al. 2019c), our meta-model is designed as a Y-shape DensVoxNet (Yu et al. 2017) (see Fig. 4), which takes two pieces of input, $\mathcal{X}_i$ and $A(\hat{\mathcal{Y}}_i)$. $A(\cdot)$ is the averaging function that forms a compact representation of $\hat{\mathcal{Y}}_i$ of the PLs. This representation shows the image areas where the PLs hold agreement or disagreement (i.e., average prediction values close to 1 or 0). In addition, using the average of all the PLs of $\mathcal{X}_i$ to form part of the meta-model's input can be viewed as a preliminary ensemble of the base-models and ease the training of the meta-model.

Rather than defining a fixed learning objective for the meta-model training, we train the meta-model in two main stages: (1) Initially, we train the meta-model in order to set up a near-optimal (or sub-optimal) configuration: The meta-model is aware of all the available PLs, and its position in the hypothesis space is influenced by the raw image and the PL

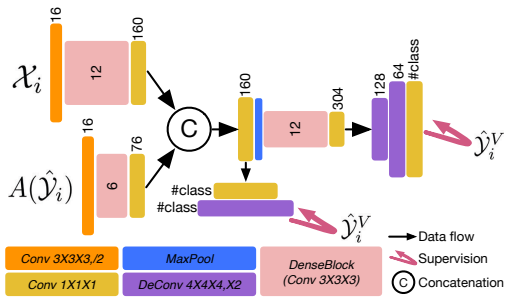


Figure 4: The meta-model structure. For readability, BN and ReLU are omitted, the number of channels is given above each unit, and the number of Conv units in each DenseBlock is shown in the block.

data distribution; (2) In the second training stage, we train the meta-model to fit the nearest PLs to help the training process converge. More technical details are given below.

In the first training stage, we seek to minimize the overall cross-entropy loss for all the image samples with respect to all the PLs:

$$\min_{\theta_{\mathcal{H}}} \sum_{i=1}^{m}\sum_{V} \ell_{mce}(\theta_{\mathcal{H}}(\mathcal{X}_i, A(\hat{\mathcal{Y}}_i)), \hat{\mathcal{Y}}_i^V), \tag{9}$$

where $\theta_{\mathcal{H}}$ is the meta-model's parameters and $\ell_{mce}$ is a multi-class cross-entropy loss. In every training iteration, for one image sample $\mathcal{X}_i$, we randomly choose a set of PLs from $\hat{\mathcal{Y}}_i^V$ ($V \in \{xy, xz, yz, 3D\}$) and set it as the "ground truth" for $\mathcal{X}_i$ in the current training iteration. Randomly choosing PLs for the model to fit ensures the supervision signals not to impose any bias towards any base-model, and allows image samples with diverse PLs to have a better chance to be influenced by other image samples.

In the second training stage, the meta-model itself chooses the nearest PLs to fit (based on its current model parameters), and updates its model parameters based on its current choices. This nearest-neighbor-fit (NN-fit) process iterates until the meta-model fits the nearest neighbors well enough. Since the overall training loss is based on cross-entropy, to make the NN-fit have direct effects on the convergence of the model training, we use cross-entropy to measure the "distance" between a meta-model's output and a PL.

## Experiments

To show the effectiveness and efficiency of our new framework, we evaluate it on two public datasets: the HVSMR 2016 Challenge dataset (Pace et al. 2015) and the mouse piriform cortex dataset (Lee et al. 2015b).

**3D HVSMR Dataset.** The HVSMR 2016 dataset consists of 10 3D MR images (MRIs) for training and another 10 MRIs for testing. The goal is to segment myocardium and great vessel (blood pool) in cardiovascular MRIs. The ground truth of the testing data is kept secret by the organizers for fair comparison. The results are evaluated using three criteria: Dice coefficient, average distance of boundaries (ADB), and symmetric Hausdorff distance. Finally, an overall score

Table 1: Quantitative results on the HVSMR 2016 dataset. DVN*: For fair comparison, we re-implement it and achieve better performance than what was reported in the original paper, and we use it as the backbone in all our experiments. The up arrows (↑) indicate that higher values are better for the corresponding metrics, and vice versa.

| Model | Annotation budget | Myocardium | | | Blood Pool | | | Overall Score (↑) |
|---|---|---|---|---|---|---|---|---|
| | | Dice (↑) | ADB[mm] (↓) | Hausdorff[mm] (↓) | Dice (↑) | ADB[mm] (↓) | Hausdorff[mm] (↓) | |
| 3D U-Net (Çiçek et al. 2016) | Full | 0.694 | 1.461 | 10.221 | 0.926 | 0.940 | 8.628 | -0.419 |
| VoxResNet (Chen et al. 2018) | | 0.774 | 1.026 | 6.572 | 0.929 | 0.981 | 9.966 | -0.202 |
| Wolterink *et al.* (Wolterink et al. 2017) | | 0.802 | 0.957 | 6.126 | 0.926 | 0.885 | 7.069 | -0.036 |
| DVN (Yu et al. 2017) | | 0.821 | 0.964 | 7.294 | 0.931 | 0.938 | 9.533 | -0.161 |
| DVN* | | 0.809 | 0.785 | 4.121 | **0.937** | 0.799 | **6.285** | 0.13 |
| Sparse DVN* w/ RA | $s_5$ | 0.792 | 1.024 | 6.906 | 0.932 | 0.898 | 7.396 | -0.095 |
| Sparse DVN* w/ RA+ST (Ours) | | **0.830** | **0.678** | **3.614** | **0.937** | **0.770** | 7.034 | **0.166** |

is computed as $\sum_{class}(\frac{1}{2}Dice - \frac{1}{4}ADB - \frac{1}{30}Hausdorff)$ for ranking, which reflects the overall accuracy of the results.

**Mouse Piriform Cortex Dataset.** The mouse piriform cortex dataset aims to segment neuron boundaries in serial section EM images. This dataset contains 4 stacks of 3D EM images. Following the setting in (Lee et al. 2015b; Shen et al. 2017), we split the dataset into the training set (the 2nd, 3rd, and 4th stacks) and testing set (the 1st stack), which are fixed throughout all experiments. Also, as in (Lee et al. 2015b; Shen et al. 2017), the results are evaluated using the Rand F-score (the harmonic mean of the Rand merge score and the Rand split score).

**Implementation Details.** Our feature extractor network is implemented with PyTorch. The decoder is initialized with a Gaussian distribution ($\mu = 0, \sigma = 0.01$) and trained with 2k epochs (with batch size 128; input sizes $128^2$ and $256^2$ for the HVSMR and mouse piriform cortex datasets, respectively). All our FCNs are implemented using TensorFlow. The weights of our 2D base-models are initialized using the strategy in (He et al. 2015). The weights of our 3D base-model and meta-model are initialized with a Gaussian distribution ($\mu = 0, \sigma = 0.01$). All our networks are trained using Adam (Kingma and Ba 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon =$ 1e-10 on an NVIDIA Tesla V100 graphics card with 32GB GPU memory. The initial learning rates are all set as 5e-4. Our 2D base-models decrease the learning rates to 5e-5 after 10k iterations; our 3D base-model and meta-model adopt the "poly" learning rate policy with the power variable equal to 0.9 (Yu et al. 2017). To leverage the limited training data, standard data augmentation techniques (i.e., image flipping along the axial planes and random rotation with 90, 180, and 270 degrees) are employed to augment the training data. Due to large intensity variance among different images, all the images are normalized to have zero mean and unit variance before feeding to the networks.

## Main Experimental Results

Our approach consists of two major components: representative annotation (RA) and self-training (ST). To evaluate the effectiveness of our proposed strategy, we first compare our approach using sparse annotation (denoted by **RA+ST**) with the state-of-the-art methods using full annotation on the two datasets. Then, we demonstrate the robustness of our method under different annotation budgets (e.g., $s_k, k = 5, 10, 20, 40, 80$ for the HVSMR dataset) comparing to the state-of-the-art DenseVoxNet (DVN) (Yu et al. 2017).

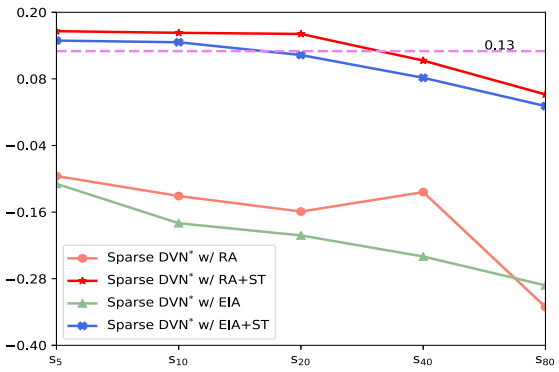


Figure 5: Evaluation of several methods on the HVSMR 2016 dataset with different annotation budgets $s_k$. Given an $s_k$, RA and EIA select different sets of slices for annotation and FCN training. "Sparse DVN* w/ RA" and "Sparse DVN* w/ EIA" are baselines. The dashed line is the performance using the fully supervised DVN*.

Table 1 gives the segmentation results on the HVSMR 2016 dataset. Note that among the state-of-the-art methods on the leaderboard, DVN achieves the highest Dice score and outdoes others on the overall score. Our re-implementation DVN* of DVN is an enhanced version and outperforms other methods by a large margin. We use DVN* as the baseline for all our experiments, for fair comparison. First, compared with the fully supervised DVN*, we obtain a significant improvement on nearly all the metrics, which demonstrates that our method is more effective. More importantly, if we measure annotation effort using the number of voxels selected as representatives by our method, $s_5$ is equivalent to $\sim 60\%$ of all voxels, which shows the efficiency of our method. Compared with sparse 3D DVN*, our method bridges the performance gap between sparse and full annotations. Second, our approach can further save more annotation effort. We conduct experiments with different annotation ratios; the results are shown in Fig. 5. One can note that the performance gap between the sparse- and fully-annotated 3D DVN* is reduced by our approach with even sparser annotation. Our RA+ST-$s_{40}$ and RA+ST-$s_{20}$ closely approach or outperform the fully supervised DVN*, i.e., our method is able to save up to $\sim 85\%$ of voxel-wise annotation. Some qualitative results are shown in Fig. 6. One can see that our method (RA+ST) achieves superior perfor-

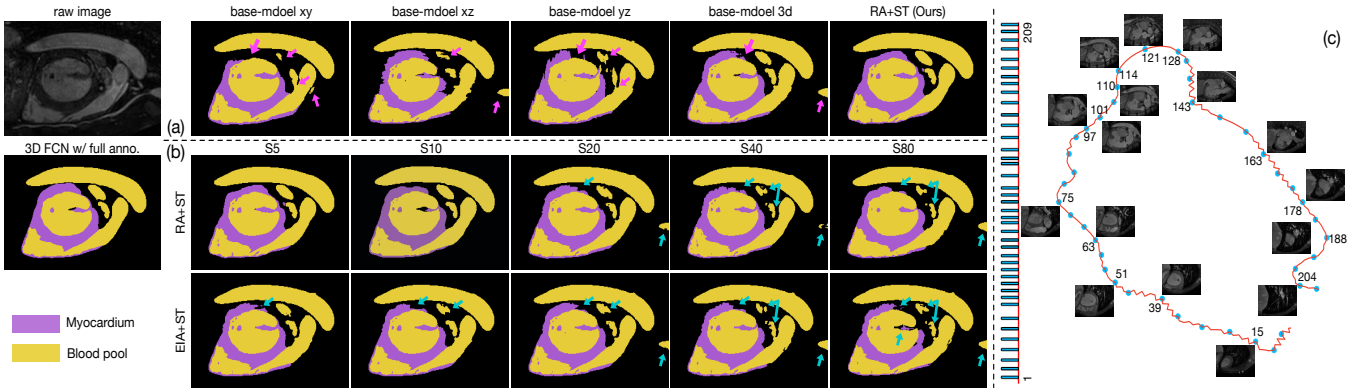


Figure 6: Some visual qualitative results on the HVSMR 2016 dataset (some errors are marked by arrows). (a) Results of the 2D and 3D base-models using annotated slices selected by RA. After self-training using pseudo-labels, our approach produces more accurate results which are comparative to that generated by 3D FCN with full annotation. (b) By comparing our strategy RA+ST (the top row of (b)) with EIA+ST (the bottom row of (b)), using slices selected by RA yields superior performance. (c) We show some slices selected by RA (for an $s_5$ budget) from a 3D stack with the $xy$-plane. After being projected to 2D space by t-SNE, each point represents one selected slice and the consecutive points form a curve. Selected slices are marked with blue dots and those shown along with thumbnails are labeled with their slice IDs. We also indicate the index positions of the slices selected by RA along the $z$-axis, as shown by the vertical line on the left of (c) that represents the $z$-axis of the stack.

Table 2: Quantitative results on the mouse piriform cortex dataset. The up arrow (↑) indicates that higher values are better for the $V^{Rand}_{Fscore}$ metric.

| Method | Anno. budget | $V^{Rand}_{Fscore}$ (↑) |
|---|---|---|
| N4 (Ciresan et al. 2012) | Full | 0.9304 |
| VD2D (Lee et al. 2015b) | | 0.9463 |
| VD2D3D (Lee et al. 2015b) | | 0.9720 |
| M$^2$FCN (Shen et al. 2017) | | 0.9866 |
| DVN$^*$ | | 0.9959 |
| DVN$^*$ | $s_4$ | 0.9970 |
| DVN$^*$ w/ RA+ST (Ours) | | **0.9971** |
| DVN$^*$ | $s_{16}$ | 0.9940 |
| DVN$^*$ w/ RA+ST (Ours) | | **0.9961** |
| DVN$^*$ | $s_{64}$ | 0.9951 |
| DVN$^*$ w/ RA+ST (Ours) | | **0.9957** |

mance than the 2D and 3D base-models, and approaches that of the fully supervised FCN (using more annotation).

We further evaluate our method on the mouse piriform cortex dataset, using similar experimental settings as those for the HVSMR 2016 dataset. Table 2 shows such results. First, we compare our method with an array of 3D FCN-based models, which are all trained with full annotation. Table 2 demonstrates that our method with sparse annotation surpasses each such single 3D FCN with full annotation. Second, one can see that with different annotation ratios, the performance gap is reduced consistently. In particular, our RA+ST-$s_{64}$ $<$ DVN$^*$-Full $<$ RA+ST-$s_{16}$, that is, our method can save up to $\sim 80\%$ of voxel-wise annotation.

## Analysis and Discussions

**On Representative Annotation (RA).** As shown in Fig. 5, we compare our strategy with a different annotation strategy: equal-interval annotation (EIA). One can see that "RA+ST" is better than "EIA+ST", which demonstrates that our representative slice selection algorithm helps select more informative and diverse samples to represent the data (see Fig. 6(c)). Given the same annotation budget, these RA-selected slices are more valuable for expert annotation.

**On Self-Training.** As shown in Fig. 5, by comparing "Sparse DVN$^*$ w/ RA+ST" with "Sparse DVN$^*$ w/ RA", and "Sparse DVN$^*$ w/ EIA+ST" with "Sparse DVN$^*$ w/ EIA", one can see that utilizing pseudo-labels (PLs) for self-training, the performance is significantly improved. It demonstrate that though PLs generated from sparse annotation may be noisy, they fill the spatial gaps of voxel-wise supervision in the 3D stack. Thus our self-training utilizes the PLs and bridges the final performance gap with respect to full annotation.

## Conclusions

In this paper, we proposed a new annotation sparsification strategy for 3D medical image segmentation based on representative annotation and self-training. The most valuable slices are selected for manual annotation, thus saving annotation effort. Heterogeneous 2D and 3D FCNs are trained using sparse annotation, which generate diverse pseudo-labels (PLs) for unannotated voxels in 3D data. Self-training utilizing PLs further improves the segmentation performance and bridges the performance gap with respect to full annotation. Our extensive experiments on two public datasets show that using less than $20\%$ annotated data, our new strategy obtains comparative results with fully supervised training.

## Acknowledgments

This research was supported in part by NSF grants CCF-1617735, CNS-1629914 and IIS-1455886, and NIH grant R01 DE027677-01.

## References


Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2016. What's the point: Semantic segmentation with point supervision. In *ECCV*, 549–565.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, 92–100.

Chen, H.; Dou, Q.; Yu, L.; Qin, J.; and Heng, P.-A. 2018. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage* 170:446–455.

Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S. S.; Brox, T.; and Ronneberger, O. 2016. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *MICCAI*, 424–432.

Ciresan, D.; Giusti, A.; Gambardella, L. M.; and Schmidhuber, J. 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, 2843–2851.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and L, F.-F. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*, 248–255.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV*, 1026–1034.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hochbaum, D. S. 1997. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In *Approximation Algorithms for NP-hard Problems*. Boston, MA, USA: PWS Publishing Co. 94–143.

Hou, L.; Agarwal, A.; Samaras, D.; Kurc, T. M.; Gupta, R. R.; and Saltz, J. H. 2019. Robust histopathology image analysis: To label or to synthesize? In *CVPR*, 8533–8542.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *CVPR*, 2261–2269.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M.; and Schiele, B. 2017. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 876–885.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kordon, F.; Fischer, P.; Privalov, M.; Swartman, B.; Schnetzke, M.; Franke, J.; Lasowski, R.; Maier, A.; and Kunze, H. 2019. Multi-task localization and segmentation for X-ray guided planning in knee surgery. *arXiv preprint arXiv:1907.10465*.

Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; and Tu, Z. 2015a. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, 562–570.

Lee, K.; Zlateski, A.; Ashwin, V.; and Seung, H. S. 2015b. Recursive training of 2D-3D convolutional networks for neuronal boundary prediction. In *NIPS*, 3573–3581.

Liang, P.; Chen, J.; Zheng, H.; Yang, L.; Zhang, Y.; and Chen, D. Z. 2019. Cascade decoder: A universal decoding method for biomedical image segmentation. In *16th IEEE International Symposium on Biomedical Imaging (ISBI)*, 339–342.

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J. 2016. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 3159–3167.

Niyogi, P. 2013. Manifold regularization and semi-supervised learning: Some theoretical analyses. *The Journal of Machine Learning Research* 14(1):1229–1250.

Pace, D. F.; Dalca, A. V.; Geva, T.; Powell, A. J.; Moghari, M. H.; and Golland, P. 2015. Interactive whole-heart segmentation in congenital heart disease. In *MICCAI*, 80–88.

Radosavovic, I.; Dollár, P.; Girshick, R.; Gkioxari, G.; and He, K. 2018. Data distillation: Towards omni-supervised learning. In *CVPR*, 4119–4128.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 234–241.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 318–362.

Shen, W.; Wang, B.; Jiang, Y.; Wang, Y.; and Yuille, A. 2017. Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection. In *ICCV*, 2391–2400.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Wolterink, J. M.; Leiner, T.; Viergever, M. A.; and Išgum, I. 2017. Dilated convolutional neural networks for cardiovascular MR segmentation in congenital heart disease. In *Reconstruction, Segmentation, and Analysis of Medical Images*, 95–102.

Yang, L.; Zhang, Y.; Chen, J.; Zhang, S.; and Chen, D. Z. 2017. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *MICCAI*, 399–407.

Yang, L.; Zhang, Y.; Zhao, Z.; Zheng, H.; Liang, P.; Ying, M. T.; Ahuja, A. T.; and Chen, D. Z. 2018. BoxNet: Deep learning based biomedical image segmentation using boxes only annotation. *arXiv preprint arXiv:1806.00593*.

Yu, L.; Cheng, J.-Z.; Dou, Q.; Yang, X.; Chen, H.; Qin, J.; and Heng, P.-A. 2017. Automatic 3D cardiovascular MR segmentation with densely-connected volumetric ConvNets. In *MICCAI*, 287–295.

Zhang, Y.; Yang, L.; Chen, J.; Fredericksen, M.; Hughes, D. P.; and Chen, D. Z. 2017. Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In *MICCAI*, 408–416.

Zhao, Z.; Yang, L.; Zheng, H.; Guldner, I. H.; Zhang, S.; and Chen, D. Z. 2018. Deep learning based instance segmentation in 3D biomedical images using weak annotation. In *MICCAI*, 352–360.

Zheng, H.; Yang, L.; Chen, J.; Han, J.; Zhang, Y.; Liang, P.; Zhao, Z.; Wang, C.; and Chen, D. Z. 2019a. Biomedical image segmentation via representative annotation. In *AAAI*, volume 33, 5901–5908.

Zheng, H.; Yang, L.; Han, J.; Zhang, Y.; Liang, P.; Zhao, Z.; Wang, C.; and Chen, D. Z. 2019b. HFA-Net: 3D cardiovascular image segmentation with asymmetrical pooling and content-aware fusion. In *MICCAI*, 759–767.

Zheng, H.; Zhang, Y.; Yang, L.; Liang, P.; Zhao, Z.; Wang, C.; and Chen, D. Z. 2019c. A new ensemble learning framework for 3D biomedical image segmentation. In *AAAI*, volume 33, 5909–5916.

Zhou, Z.; Shin, J.; Zhang, L.; Gurudu, S.; Gotway, M.; and Liang, J. 2017. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *CVPR*, 7340–7351.

Zhou, Y.; Wang, Y.; Tang, P.; Bai, S.; Shen, W.; Fishman, E. K.; and Yuille, A. L. 2019. Semi-supervised multi-organ segmentation via deep multi-planar co-training. In *WACV*, 121–140.