Interactive Few-Shot Learning: Limited Supervision, Better Medical Image Segmentation

Ruiwei Feng, Xiangshang Zheng, Tianxiang Gao, Jintai Chen, Wenzhe Wang, *Student Member, IEEE*, Danny Z. Chen, *Fellow, IEEE*, and Jian Wu, *Member, IEEE*

***Abstract*— Many known supervised deep learning methods for medical image segmentation suffer an expensive burden of data annotation for model training. Recently, few-shot segmentation methods were proposed to alleviate this burden, but such methods often showed poor adaptability to the target tasks. By prudently introducing interactive learning into the few-shot learning strategy, we develop a novel few-shot segmentation approach called Interactive Few-shot Learning (IFSL), which not only addresses the annotation burden of medical image segmentation models but also tackles the common issues of the known few-shot segmentation methods. First, we design a new few-shot segmentation structure, called Medical Prior-based Few-shot Learning Network (MPrNet), which uses only a few annotated samples (e.g., 10 samples) as support images to guide the segmentation of query images without any pre-training. Then, we propose an Interactive Learning-based Test Time Optimization Algorithm (IL-TTOA) to strengthen our MPrNet on the fly for the target task in an interactive fashion. To our best knowledge, our IFSL approach is the first to allow few-shot segmentation models to be optimized and strengthened on the target tasks in an interactive and controllable manner. Experiments on four few-shot segmentation tasks show that our IFSL approach outperforms the state-of-the-art methods by more than 20% in the DSC metric. Specifically, the interactive optimization algorithm (IL-TTOA) further contributes ∼10% DSC improvement for the few-shot segmentation models.**

***Index Terms*— Medical image segmentation, few-shot learning, interactive learning, limited supervision.**

Manuscript received January 13, 2021; revised February 11, 2021; accepted February 16, 2021. Date of publication February 19, 2021; date of current version September 30, 2021. The work of Jian Wu was partially supported by the National Research and Development Program of China under grant No. 2019YFB1404802, No. 2019YFC0118802, and No. 2018AAA0102102, the National Natural Science Foundation of China under grant No. 61672453, the Zhejiang University Education Foundation under grants No. K18-511120-004, No. K17-511120-017, and No. K17-518051-02, the Zhejiang public welfare technology research project under grant No. LGF20F020013, the Medical and Health Research Project of Zhejiang Province of China (No. 2019KY667), the Wenzhou Bureau of Science and Technology of China (No. Y2020082), and the Key Laboratory of Medical Neurobiology of Zhejiang Province. D. Z. Chen's research was supported in part by NSF Grant CCF-1617735. *(Ruiwei Feng, Xiangshang Zheng, and Tianxiang Gao contributed equally to this work.) (Corresponding author: Jian Wu.)*

Ruiwei Feng, Xiangshang Zheng, Tianxiang Gao, Jintai Chen, Wenzhe Wang, and Jian Wu are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: ruiwei_feng@zju.edu.cn; xszheng@zju.edu.cn; gaotianxiang@zju.edu.cn; jtigerchen@zju.edu.cn; wangwnezhe@zju.edu.cn; wujian2000@zju.edu.cn).

Danny Z. Chen is with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: dchen@nd.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMI.2021.3060551, provided by the authors.

Digital Object Identifier 10.1109/TMI.2021.3060551

I. Introduction

SEGMENTING objects in medical images is one of the most fundamental tasks in medical image analysis, helping clinical diagnosis and treatment planning. Recently, deep learning methods have been shown to be highly effective for organ/tissue segmentation [1]. Yet, most of these methods were conducted in a fully supervised manner, requiring extensive training data and detailed annotations. It is time-consuming and expensive to annotate large amounts of pixel level ground truth for segmentation tasks. Moreover, in practical scenarios, a medical task may face the dilemma that very few images are available for model training (e.g., for some rare diseases). Thus, it is important to study how to alleviate the heavy annotation burden dependence of deep learning models and address these issues.

Few-shot learning is a machine learning method for learning a task from limited images and supervision [2], which helps address the aforementioned issues. Segmentation based on few-shot learning is an emerging research topic. Many existing few-shot segmentation models followed the network structure of the first work [3], comprising a support branch [4]–[7] or a prototype learner [8], [9], and a query branch. Fig. 1 illustrates the common idea of these models: the support branch (or prototype learner) receives support images, and the query branch receives query images; the specific connections between these two branches convey support information to guide feature extraction and segmentation of the query images. Hence, when using such models for few-shot segmentation tasks, researchers generally followed a specific mechanism [3]: training a model on multiple tasks by adopting the mode of receiving annotated support-query image pairs; then using it on an unseen target task for query segmentation, with very few annotated samples as the support input.

Although the few-shot segmentation models achieved notable success in reducing the annotation burden and supervision, they still have considerable drawbacks, especially the lack of further optimization on the target task.

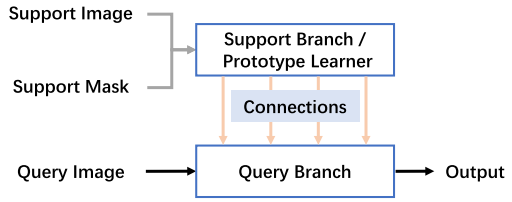

Fig. 1. Illustrating the common idea of the known few-shot image segmentation models.

As mentioned above, few-shot segmentation models are usually trained on multiple tasks, in which the models accumulate a general ability of utilizing support images to guide query segmentation. This ability enables a few-shot segmentation model to attain certain performance on an unseen target task, even when only very few annotated samples are available [9]. Yet, it is a critical issue that this general ability is possibly insufficient for dealing with some unique characteristics of the target task.

This issue may be less severe when the models are used for natural images, since they can be pre-trained on various tasks (e.g., ImageNet [10]) to acquire lots of helpful features and strong prior knowledge. But, this issue is more serious on medical images for two main reasons: 1) the lack of pre-trained models specifically for medical images [11], and 2) the substantial differences between natural and medical images (e.g., the specific organ localization and the unique characteristics of medical images [12]) may compromise the effectiveness of simply transferring the knowledge of natural images to medical imaging tasks [13], [14]. Facing such challenges, the authors in [11] proposed the first few-shot segmentation model, sSENet, for dealing with few-shot medical image segmentation tasks without any pre-training. sSENet adapted the common idea of the few-shot segmentation models and built strong connections between the two branches [11], instead of a single connection as in previous models. This structure achieved state-of-the-art performances for several few-shot medical image segmentation tasks. However, the segmentation performances were not satisfactory enough. Furthermore, this method still suffered the aforementioned drawbacks of the common idea of the existing few-shot segmentation models, which is lack of further optimization on the target task.

In this paper, we propose a novel **I**nteractive **F**ew-**S**hot **L**earning (IFSL) approach for medical image segmentation (shown in Fig. 2), to address the challenge of annotation burden and the aforementioned drawbacks. Our IFSL approach aims to tackle the following specific issues: How to obtain a stronger general ability of utilizing limited support information to guide segmentation; how to strengthen the obtained ability for the target task with the support of limited supervision. For the former issue, we develop a new structure for few-shot segmentation, called Medical Prior-based Few-shot Learning Network (MPrNet). For the latter issue, we introduce the idea of interactive learning into the few-shot learning strategy and propose an Interactive Learning-based Test Time Optimization Algorithm (IL-TTOA).

Our MPrNet has an Encoder-Decoder structure. In the encoder, two branches (a Spatial Branch and a Texture Branch) are constructed to extract features of the support images, and a Query Branch is constructed for query images. The Spatial Branch is designed to capture more spatial features of the target, with large receptive fields. These features are then conveyed to the Query Branch for spatial attention by introducing the "Channel Squeeze & Spatial Excitation" (sSE) blocks. The Texture Branch pays attention to detailed texture features of the support images and builds strong interactions with the Query Branch through weight-sharing operations. The decoder consists of two branches (Decoder-S and Decoder-Q), followed from the network in [11]. In this structure with multi-branches and strong interactions, two priors of some medical images (relatively fixed organ location and general texture nature of medical images) are considered to make up for the disadvantage of the lack of pre-training, which was rarely explored in previous studies. When using MPrNet for the target task, we first apply our IL-TTOA to interactively optimize MPrNet (trained on other tasks), directed by very little additional human supervision and corrections. In this process, several gradient descent steps on a small dataset of the target task are conducted, with the guidance of a specific obtained-ability aware loss function ($\mathcal{L}_{\mathcal{O}}$). The $\mathcal{L}_{\mathcal{O}}$ loss aims to balance the impacts of the obtained ability during training and the additional supervision of the target task.

In this manner, on one hand, our MPrNet can directly utilize the limited supervision of very few annotated samples by taking these samples as the support input of the target task (like most known few-shot learning models [11], [15]). On the other hand, our IL-TTOA can dynamically utilize additional human supervision to strengthen MPrNet for better performance on the target task. In clinical applications, our IFSL approach can help doctors annotate/segment objects with very light burden (only few samples with annotated full pixels are required), and enable dynamic supervision and correction (additionally annotating a small number of pixels) through interactive corrections. Besides, our IL-TTOA can be adapted to enhance any trained models for better performance, and enable humans to be included in the deep learning based clinical pipeline to conduct segmentation for improvement.

In summary, there are four major contributions in our work:

- A new structure, MPrNet, of multiple branches with strong connections for few-shot segmentation of medical images.
- A new algorithm, IL-TTOA, to address the deficiency of the known few-shot segmentation methods through human interactive corrections.
- Experiments show that our IFSL approach achieves promising performance on four few-shot segmentation tasks. The main idea of our IFSL approach (interactively optimizing few-shot segmentation models on the target task by IL-TTOA) further yields notable improvements and shows great potential for strengthening any few-shot segmentation methods.
- Our IFSL approach, especially IL-TTOA, is of clinical significance: (1) alleviating the burden of medical experts on annotating images, and (2) enabling humans to be included in the deep learning based clinical pipeline to strengthen the models on the target tasks (not restricted to few-shot learning models).

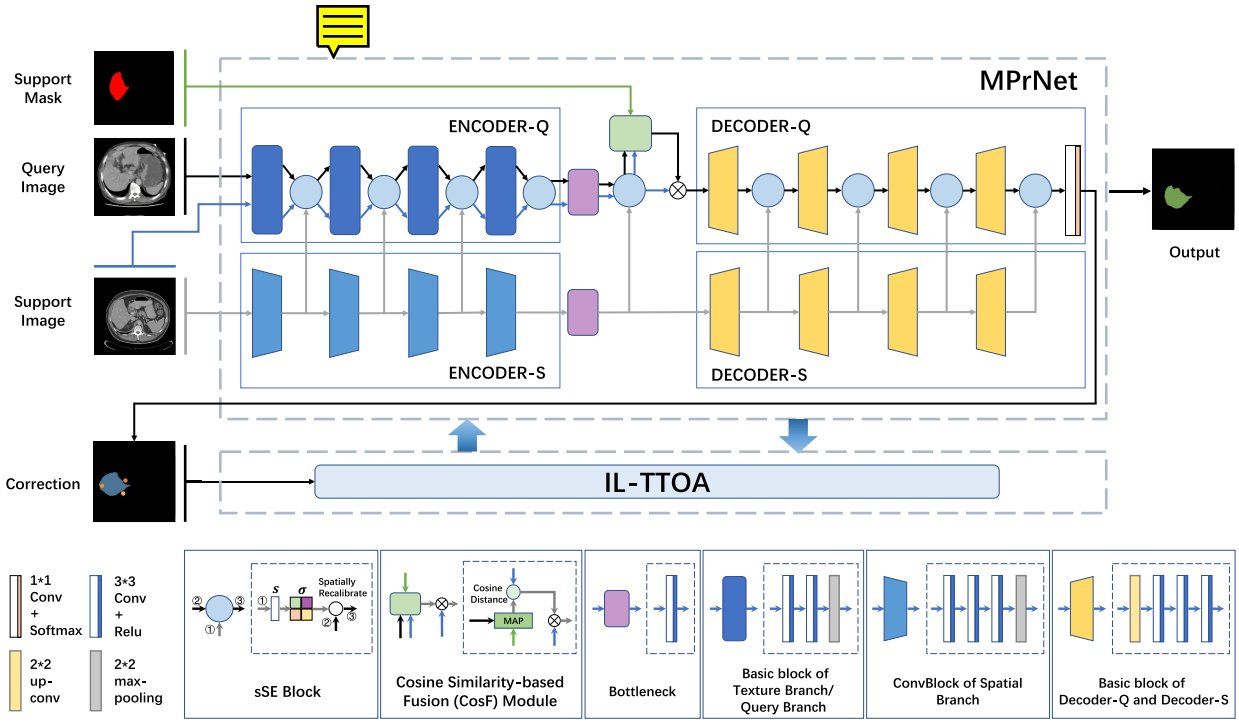

Fig. 2. Illustrating our IFSL approach. The top part is an overview of IFSL, including MPrNet and IL-TTOA. In MPrNet, Encoder-Q represents the Texture Branch and Query Branch (the blue and black lines indicate the data flows of these two branches, respectively), and Encoder-S is the Spatial Branch (the data flow is illustrated by grey lines). IL-TTOA is used to strengthen our MPrNet on the target task, using interactive human corrections on the prediction of MPrNet and optimizing the parameters of MPrNet several times. The bottom boxes show the detailed components of MPrNet. In each box, the left part is an outline of the specific block and the right part is the detailed structure of it.

II. Related Work

Recently, a large body of research work has focused on the problem of alleviating the data and annotation burdens for deep learning methods (e.g., as reviewed in [16], [17]). These studies dealt with different specific issues and mainly worked in different stages of the deep learning pipeline (these stages, as summarized in [18], [19], include sample acquisition, data interpretation and annotation, model training, etc.). In this section, we review these studies and roughly categorize them into three groups according to the specific issues they focused on and the stages they mainly worked with.

1) Methods for the Sample Acquisition Burden: In some medical scenarios, one may face the dilemma that only very few medical images for the target task can be collected (e.g., for some rare diseases). On one hand, some methods (e.g., zero, one, or few shot learning, and meta learning approaches) mainly focused on addressing this dilemma and broadly aimed to learn characteristics of a new target task from only few samples. These methods are desirable when only very few samples can be obtained as training samples, including the extreme cases where the labeled examples are just one or none. However, these methods still performed poorly on medical images (especially for segmentation tasks [11]), although they have worked well on natural images. In [20], the authors reviewed such diverse approaches, including zero, one, and few shot learning approaches. In [21], the authors reviewed different approaches for few-shot learning and categorized these approaches into data-based approaches, model-based approaches, and algorithm-based approaches. So far, known methods of this type focused on classification (e.g., [15], [22], [23]), segmentation (e.g., [11], [24]), and other tasks (e.g., [25], [26]). But, no review or survey paper has specifically covered this type of methods on medical images. Thus, this is a topic that is worth exploring but it is challenging to alleviate the burden of medical image data collection.

On the other hand, some work aimed to resolve this dilemma by expanding and diversifying the training sets by data augmentation (instead of expert annotation). Methods of this type include not only the traditional data augmentation operations such as Gaussian blurring [27], appearance enhancement (e.g., [28], [29]), spatial transformations (e.g., [30], [31]), etc., but also synthetic approaches like CycleGANs [32], conditional GANs [33], SGAN [34], mask-guided GAN [35], etc.

2) Methods for the Annotation Acquirement Burden: In general, it is time-consuming and expert effort-intensive to give the collected medical image samples artificial labels in the annotation stage (especially for pixel level annotation for segmentation tasks). To deal with this issue, on one hand, some methods focused on selecting the few most valuable samples in a more efficient way for annotations and model training, called active learning [16]. Generally, an active learning strategy selects the most valuable samples, labels them, and subsequently adds them to the training set for further training. Many active learning frameworks have been proposed for medical image segmentation (e.g., [36]–[38]), classification and detection (e.g., [39]–[41]). On the other hand, some methods attempted to attach annotation and supervision in an interactive manner, called interactive learning. It can accelerate the annotation process by allowing expert annotators to interactively correct the prediction generated by

a model [17], which may have a significant impact on the segmentation tasks (e.g., [42], [43]). Specifically, the authors in [42] enabled CNNs trained in a fully supervised manner to further adapt to test images based on user interactions (e.g., user providing bounding boxes or scribbles). In [44], the authors reviewed the approaches for interactive medical image segmentation.

3) Methods for Model Training Issues: In the training stage, there is a clear trade-off between model performance and training set scale. Deep learning models can be prone to over-fitting if trained on only few samples, while enlarging the training set may yield better performance but incur expensive time and expert-effort costs. Some studies sought to resolve this conflict by exploring strategies to train deep learning models more efficiently and effectively and to better leverage the limited labeled data in the training stage with small or imperfect labeled training sets. In [45], the authors reviewed approaches of this type in medical image analysis. For instance, methods like transfer learning (including pre-trained operations; e.g., [13], [46]) and multi-task learning approaches [47]–[49] proposed to leverage the characteristics or domain knowledge of other tasks for the target task. These methods may alleviate the data and annotation burden of the target task, as well as accelerating the training process [17]. Besides, some weakly-supervised approaches (e.g., [50], [51]) and semi-supervised approaches (e.g., [52]–[55]) were given to leverage unlabeled data or imperfect data in conjunction with well-labeled data to train high-performance segmentation models.

The aforementioned methods have achieved notable successes on alleviating the burden of data and annotation in medical image analysis, aiming at training more effective models with less data and annotation costs. Some of them can be used in more than one situation, and some may work cooperatively for better performance in certain scenarios. For instance, in [56], the authors integrated active learning and transfer learning to reduce annotation effort. In this paper, we focus on the extreme cases when only very few medical images are available for annotation and model training, in order to explore a more efficient and effective framework for medical image segmentation.

III. Method

In this section, we present our IFSL approach, which consists of a new Medical Prior-based Few-shot Learning Network (MPrNet) and an Interactive Learning-based Test Time Optimization Algorithm (IL-TTOA), as illustrated in Fig. 2.

A. Problem Setup

We follow the formulation of few-shot segmentation in [3]. Given a support set $S = \left\{(I_s^i, Y_s^i(l))\right\}_{i=1}^{K}$, which is a small set of K image and binary mask pairs for a target task, where I_s^i is the i-th image and $Y_s^i(l)$ is the corresponding binary mask of the target semantic class $l \in L_{test}$, the goal is to learn a model $\mathcal{M}$ to generate a binary mask $M_q(l)$ of a query image I_q given the support set S, denoted as $M_q(l) = F(\mathcal{M}; S, I_q)$. To train this model $\mathcal{M}$, we may have access to a large number of image-mask pairs of other semantic classes for training the model, denoted as $D = \left\{(I_d^i, Y_d^i)\right\}_{i=1}^{N}$, where Y_d^i is the binary mask of a training image I_d^i and N is the number of training images. There are maybe multiple semantic classes (L_{train}) in D, with $L_{train} \cap L_{test} = \emptyset$. In this paper, we simply refer to the segmentation process for the target class as the *test stage*, and the process of model training on other tasks as the *training stage*. Table I specifies the notation used in this section.

B. Medical Prior-Based Few-Shot Learning Network (MPrNet)

Our MPrNet is constructed with an Encoder-Decoder structure [1], following the common idea of existing few-shot segmentation models [11], [15] (as shown in Fig. 1). In the encoder, we build a Spatial Branch and a Texture Branch for support images, and a Query Branch for query images. In the decoder, we simply follow the decoder structure of sSENet [11], building two branches (a Decoder-Q branch and a Decoder-S branch). Further, a Cosine Similarity based Fusion (CosF) module is introduced between the encoder and decoder. This structure is elaborately designed by adopting two kinds of strong prior knowledge of medical images [12]: (1) the location of a target organ in medical images is often relatively fixed, and (2) the general texture nature of medical images is distinguishable from that of non-medical images.

1) Encoder: In the encoder, we adopt the aforementioned two knowledge priors from two perspectives: individual branches with different structures (Spatial Branch and Texture Branch) for diverse characteristics of the support images; the specific information interaction fashions of the two branches with the Query Branch. For clarity, in Fig. 2, the data flows of the three branches in the encoder (Spatial Branch, Texture Branch, and Query Branch) are illustrated by lines in grey, blue, and black, respectively.

a) Three Branches: First, we build the Spatial Branch with a lightweight structure and large receptive fields, by considering the first knowledge prior of spatial information. This branch consists of four sequential 5×5 naïve ConvBlocks, as illustrated in Fig. 2. The large kernel size (5×5) for large receptive fields enables this branch to be aware of more globally spatial characteristics of the target, while the lightweight structure alleviates the burden of heavy parameters along with the large kernel size. Then, we build the Texture Branch to capture more detailed texture characteristics of the support images. Inspired by the VGGNet [57] structure, the Texture Branch is composed of four repeated convolutional blocks with two convolutions of small kernel size (3×3), each of which is followed by a ReLU and a 2×2 max pooling operation for down-sampling (as shown in Fig. 2). Finally, we build the Query Branch with the same structure as the Texture Branch, to capture the detailed characteristics of the query images that are similar to those of the support images (since the support and query images are for the same target organ and generally show quite similar characteristics). In this structure, we choose the three branches to have symmetric layouts (four blocks each), which help put strong interactions between the matching blocks.

b) Information Interaction Fashions: Information interaction plays an important role in a few-shot segmentation model [11], which determines how the information of the support images could guide the feature extraction and segmentation of the query images. In the encoder, we accomplish information

TABLE I
SUMMARY OF NOTATION USED IN SECTION III

$D = \{(I_d^i, Y_d^i)\}_{i=1}^N$	Training set (other tasks) containing N image (I_d) and binary mask (Y_d) pairs
$S = \{(I_s^i, Y_s^i(l))\}_{i=1}^K$	Support set (target task) containing K image (I_s) and binary mask (Y_s) pairs
$D_o = \{I_e^n\}_{t=1}^{N^*}$	Dataset (target task) containing N^* images for IL-TTOA
I_q, M_q	Query image and prediction mask
$\mathcal{M}$, $\mathcal{M}(\theta)$	The Few-shot segmentation model (MPrNet) and the model with parameters θ
$O(\cdot)$, M_p	Human operation in IL-TTOA and the corresponding operation map
M_g	Artificially corrected mask computed by $M_p \bigoplus M_q$ / pseudo ground-truth mask in IL-TTOA

interaction through three steps. First, we establish directed connections from the Spatial Branch to the Query Branch, by bridging each of the four ConvBlocks in the Spatial Branch to the corresponding block in the Query Branch (as illustrated in Fig. 2). These operations are conducted based on the knowledge prior that a specific target organ in a support-query image pair generally appears in a relatively fixed location. Thus, such one-to-one connections and interaction help convey the captured global spatial features of the support images to guide query feature extraction, from low-level to high-level.

Next, a weight-sharing operation is introduced between the Texture Branch and the Query Branch, based on the texture knowledge prior and the same structures of these two branches. This operation helps the extracted texture features of the support images guide the Query Branch in a more direct and efficient manner. Of course, the extracted features of the query images may also affect the feature extraction of the support images, due to the undirected interactions.

Finally, we introduce the "Channel Squeeze & Spatial Excitation" (sSE) block to facilitate feature interaction between the Spatial Branch and the Query Branch. sSE block is a recently proposed computational unit [11], which squeezes a feature map along the channel for spatial attention. In this study, we use it to squeeze the feature maps of the Spatial Branch and excite the feature maps of the Query Branch, by adding it to the directed connections in the first step. The detailed structure of sSE block is illustrated in the sub-figure labeled as **sSE Block** of Fig. 2, in which the squeeze and excite operations are denoted by s and σ, respectively. That is, the i-th sSE block ($i \in \{1, 2, 3, 4\}$) in the encoder receives features from the i-th ConvBlock in the Spatial Branch (the data flow denoted by grey line ①) and the i-th basic block in the Query Branch (the data flow denoted by black line ②), and delivers the incorporated features to the $(i+1)$-th basic block in the Query Branch (the data flow denoted by black line ③).

2) Cosine Similarity Based Fusion (CosF) Module: To leverage the support masks as an attention to guide query segmentation, we propose a Cosine Similarity based Fusion (CosF) module, and place it after the encoder. This module measures the similarity between the features of the support images and those of the query images, and then uses the similarity as an attention weight for the query features. Specifically, we first conduct masked average pooling for $\hat{M}_{tex}$, which is accomplished by directly masking Y_s over $\hat{M}_{tex}$ (in each channel) to produce foreground and background features as

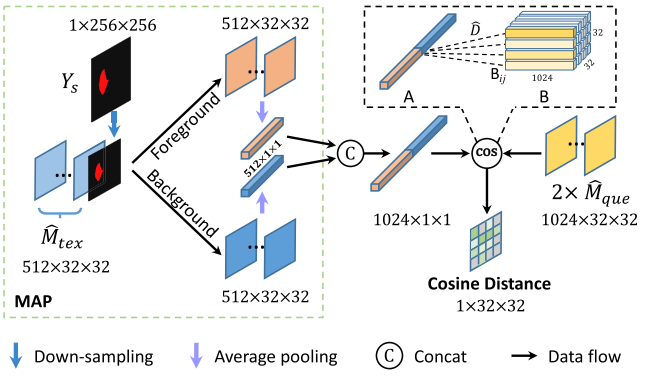

Fig. 3. Illustrating the process of conducting masked average pooling and computing cosine distance. The green dashed box indicates the operation of masked average pooling (MAP). The black dashed box shows the detailed operation of computing cosine distance.

in [6]. Next, we concatenate the foreground and background features to form a whole map and compute its cosine distance with $\hat{M}_{que}$. Intuitively, we illustrate the operations of masked average pooling and computing cosine distance in Fig. 3. The generated cosine distance matrix (of size $1 \times 32 \times 32$) is then used as attention weight for the features of the query images ($\hat{M}_{que}$) by dot product in each channel.

Given a support image I_s, the corresponding ground truth mask Y_s, and a query image I_q, the computing steps in the CosF module are summarized in Eq. (1):

$$\hat{M}_{que}^D = \hat{D}(MAP(\hat{M}_{tex}, Y_s), \hat{M}_{que}) \cdot \hat{M}_{tex} \tag{1}$$

where the features of I_s captured by the Texture Branch are denoted by $\hat{M}_{tex}$, and those of I_q obtained by the Query Branch are denoted by $\hat{M}_{que}$. $\hat{M}_{que}^D$ is the output features of the CosF module, which is also the input of the Decoder-Q. MAP denotes the operation of conducting masked average pooling. $\hat{D}$ represents the process of computing cosine distances, which can be summarized in Eq. (2) below (also illustrated in the black dashed box of Fig. 3):

$$cos_{ij} = \frac{A \cdot B_{ij}}{\sqrt{A^2} \cdot \sqrt{{B_{ij}}^2}}, \quad i = 1, 2, \ldots, 32, \;\; j = 1, 2, \ldots, 32 \tag{2}$$

where cos_{ij} is an element of the aforementioned cosine distance matrix, A is a vector of size 1024 obtained by concatenating the foreground and background features (as illustrated

in Fig. 3), and B_{ij} is a size 1024 component vector of the query features B (denoted by yellow cuboids in Fig. 3). Specifically, B (of size $1024 \times 32 \times 32$) is obtained by copying and concatenating $\hat{M}_{tex}$ (of size $512 \times 32 \times 32$) along the channel dimension. In this manner, the length of each B_{ij} is equal to that of A and the cosine similarity between these two vectors can hence be computed. To further describe the computational details of $\hat{D}$, we may simply denote vector B_{ij} as b. Then, Eq. (2) above can be written as:

$$cos_{ij} = \frac{\sum_{k=1}^{K} A_k \cdot b_k}{\sqrt{\sum_{k=1}^{K} {A_k}^2} \cdot \sqrt{\sum_{k=1}^{K} {b_k}^2}} \tag{3}$$

where A_k and b_k are elements of vectors A and b (i.e., B_{ij}) respectively, and K is the length of vectors A and b ($K = 1024$ in this work). In this stage, we empirically adopt cosine distance to measure the feature similarity, following the work in [9] which showed that cosine distance contributes to a more stable and better performance compared to other distance functions (e.g., the squared Euclidean distance).

3) Structure Comparison and Discussion: Now we discuss the differences between our MPrNet and sSENet [11]. sSENet is regarded as the first few-shot learning segmentation model for medical images, and we use it as our strong baseline. In [11], the authors proved that the strong connections between the support and query branches are valid and the sSE blocks are effective. Therefore, in our MPrNet, we inherit the basic decoder structure of sSENet and also introduce the sSE blocks. Furthermore, we construct a new encoder structure and put the CosF module between the encoder and decoder. In addition, there are three other notable differences. (1) In sSENet, a support image is directly concatenated with its corresponding ground truth mask as the single input of the support/condition branch. However, our MPrNet receives a support image and its mask separately. More specifically, the Spatial Branch and Texture Branch receive the support image, while the CosF module receives the support mask. In this way, our MPrNet could possibly preserve the original characteristics of the images better. Note that these characteristics are different by nature with those of binary masks, and simply concatenating them (as in sSENet) may damage the original characteristics of images [9]. (2) In [11], only one lightweight support branch (denoted as Condition Branch) was used to capture the features of the support input, which might be insufficient for the complicated characteristics of medical images. Unlike sSENet and other previous networks with only one branch for support images, our MPrNet first constructs two branches for support images. These two branches of different structures concentrate on the diverse characteristics of medical images and work complementarily (as shown in the Section IV-C.3). (3) Furthermore, compared to the naive encoder-decoder structure of sSENet, we introduce the CosF module between the encoder and decoder. This module leverages the support masks as an attention to perform query segmentation, and its effectiveness is verified in Section IV-C.

C. Interactive Learning-Based Test Time Optimization Algorithm

As illustrated in Fig. 2, we propose an Interactive Learning-based Test Time Optimization Algorithm (IL-TTOA) to strengthen our trained MPrNet on the target task, in order to address the drawbacks of the known few-shot segmentation models discussed in Section I. First, we select a few unannotated samples of the target task to form a small dataset, $D_o = \{I_e^i\}_{i=1}^{N^*}$. The value of the sample number N^* may depend on the specific task. Next, we strengthen the trained MPrNet ($\mathcal{M}(\theta^0)$) with our IL-TTOA on D_o, and obtain an optimized model $\mathcal{M}(\theta)$ for the final usage on the target task. In IL-TTOA, we interactively add human supervision as guidance or corrections, and conduct several times the gradient descent steps on D_o, until the segmentation reaches a desired quality level or anthropogenic interruption. In this process, we face two major issues. (1) How to make better use of the additional human supervision, so that the trained model can acquire improved ability to capture the unique characteristics of an unseen target task. (2) How to enable the trained model to maintain the general power obtained in the training process. For these two considerations, we design a specific obtained-ability aware loss function ($\mathcal{L}_{\mathcal{O}}$) to balance these two abilities.

1) Workflow of IL-TTOA: Starting from a trained MPrNet $\mathcal{M}(\theta^0)$, with parameters θ^0, we conduct a basic loop processing for each sample $I_e^n \in D_o$: "*Prediction — Correction — Modification*", which is more precisely illustrated in Algorithm 1. For each sample I_e^n, we conduct *Iter* iterations of this loop. Fig. 6 gives an example with 20 slices in D_o and 10 iterations. *Prediction* means using the current model $\mathcal{M}$ to predict a binary mask M_q for I_e^n. *Correction* means requesting a human expert to correct the predicted mask (M_q) by pointing out any error regions. This operation is denoted as $O(\cdot)$ and the added correction information is denoted as M_p. M_p is a matrix of values in $\{1, 0, -1\}$, indicating that each pixel is corrected to what label. Specifically, pixels corrected as positive are set to 1 and as negative are set to -1. The remaining ones with no correction are set to 0. *Modification* means updating the parameters (θ^*) in model $\mathcal{M}$ by running a gradient descent step to minimize our $\mathcal{L}_{\mathcal{O}}$ loss. In this process, we utilize a pixel-wise addition ($\bigoplus$) to combine the predicted map M_q and correction map M_p, and generate the corrected mask M_g. Thus, the false negative pixels in M_q are corrected as 1 in M_g (corresponding to true positive), while the false positive pixels in M_q are corrected as 0 in M_g (corresponding to true negative). Then, M_g is taken as a pseudo ground-truth mask to compute a Dice loss for the predicted mask M_q. The Dice loss is part of our $\mathcal{L}_{\mathcal{O}}$ loss to guide the parameter optimization.

2) Obtained-Ability Aware Loss Function ($\mathcal{L}_{\mathcal{O}}$): Our $\mathcal{L}_{\mathcal{O}}$ loss is defined in Eq. (4) below. It combines a naive Dice loss ($L_{dice}(\theta^*)$) and a regularizer. The former mainly measures the loss caused by additional supervision and corrections in IL-TTOA. The latter penalizes the changes to parameters according to their importance ($\Omega_{i,j}$) for the previous tasks. The changes to parameters that are deemed important can be severely penalized, so that important knowledge of the previous tasks can be prevented from being overwritten [58]. λ is a hyper-parameter playing a trade-off role between L_{dice} and the regularizer.

$$\mathcal{L}_{\mathcal{O}}(\theta^*) = L_{dice}(\theta^*) + \lambda \sum_{i,j} \Omega_{i,j} (\theta^*_{i,j} - \theta_{i,j})^2 \tag{4}$$

The regularizer adapts the Memory Aware Synapses (MAS) proposed in [58], which were originally for classification tasks

Algorithm 1: IL-TTOA Interactive Learning-Based Test Time Optimization Algorithm

Input: A trained model $\mathcal{M}(\theta^0)$ with parameters θ^0, a support set S, a dataset $D_o = \{I_e^i\}_{i=1}^{N^*}$, and a number *Iter* of interactive iterations.
Output: $\mathcal{M}(\theta)$
$\mathcal{M} \leftarrow \mathcal{M}(\theta^0)$ ▷ Initialize model
for each n in N^* **do**
  ▷ Modify $\mathcal{M}$ *Iter* times for each image in D_o
  $I_s \leftarrow I_s^n$, $I_q \leftarrow I_e^n$
  ▷ Input query image and support image
  for each a in *Iter* **do**
    $M_q^n = F(\mathcal{M}; I_s, I_q)$
    ▷ Predict mask using the current model
    $M_p^a = O(M_q^n, I_q)$
    ▷ Artificially correct the prediction
    $M_g^a = M_p^a \bigoplus M_q^n$
    $\theta^* \leftarrow \text{Minimize}(\mathcal{L}_{\mathcal{O}}(M_q^n, M_g^a; \theta^*))$
    ▷ Update parameters
    $\mathcal{M} \leftarrow \mathcal{M}(\theta^*)$
$\mathcal{M}(\theta) \leftarrow \mathcal{M}$

in lifelong learning strategies. As discussed in [58], it measures the importance of each parameter in the network, according to the sensitivity of the predicted output to the change of the parameter. In this work, we adapt MAS for our segmentation tasks, by treating segmentation as a task of assigning a consistent semantic class label (organ or background) to each pixel. Other than regarding an image as a data point in [58], we treat a pixel as a data point instead. Thus, following [58], we denote the gradient of the learned function with respect to parameter $\theta_{i,j}$ for each pixel x_{pi} as $g_{i,j}(x_{pi})$. θ is the "old" network parameters (the network trained on several tasks) whose values are relatively fixed, and θ^* is the "new" network parameters updated in each iteration. The importance weight $\Omega_{i,j}$ for parameter $\theta_{i,j}$ can be obtained by:

$$\Omega_{i,j} = \frac{1}{N \times W \times H} \sum_{1}^{N} \sum_{pi=1}^{W \times H} \left\| g_{i,j}(x_{pi}) \right\| \tag{5}$$

Here, we use N images to compute $\Omega_{i,j}$ and each image contains $W \times H$ pixels. In practical applications, considering the time and space costs of such computation, we may down-sample the images by $4\times$ in the case of similar results. For every target task, Ω is calculated only once on the training set. In this way, we allow the parameters to change with the help of additional supervised information on the target task, while important knowledge learned from the previous tasks may be protected. Thus, our IL-TTOA likely provides a new point of view for few-shot segmentation models in clinical scenarios, which allows medical doctors to inform the models to be aware of the missed characteristics of the target task in an interactive and real-time fashion.

IV. Experiments and Evaluations

A. Datasets

Public datasets: The KiTS19 challenge dataset [59] and LiTS challenge dataset [60]. Both KiTS19 and LiTS are training sets used in organ segmentation challenges. KiTS19 contains 209 3D CT stacks with slice-level annotations for the kidney. LiTS contains 131 3D CT stacks with annotations for the liver. Fig. 4 gives an example of each dataset.

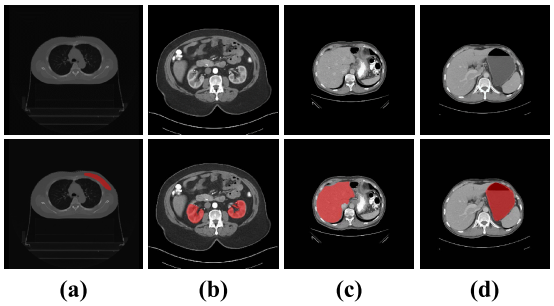

Fig. 4. Examples of CT slices of four organs or tissues. (a) CTV for BC, (b) kidney, (c) liver, and (d) stomach. The top row gives the original slices and the bottom row shows the slices with segmentation masks (marked in red).

In-house datasets: The *Stomach* dataset and *Breast* dataset. The *Stomach* dataset contains 153 3D CT stacks with slice-level annotations for the stomach. The *Breast* dataset is for the right-sided clinical target volume (CTV) for breast cancer (BC) radiotherapy, containing 114 3D CT stacks with the corresponding CTV annotations. CTV segmentation is an essential step for successful treatment delivery, since the CTV usually includes tissues with potential tumor spread or sub-clinical diseases. However, automatic segmentation of the CTV is very challenging, due to the low contrast visibility without clear delineation [61] (as shown in Fig. 4).

All the CT stacks are of size $256 \times 256 \times C$ and $C \in [160, 220]$ is the number of slices in a 3D stack (after uniform resampling with a thickness of $2mm$). In this work, we conduct all the experiments on 2D images (i.e., with slices of the stacks as input).

B. Experimental Setups

In this study, we consider each target organ/tissue as a semantic class with only a few labeled images, while the datasets of the other tasks are packed into a training set D for training the models. For convenience, in the experiments below, the four segmentation tasks are denoted as Breast, Kidney, Liver, and Stomach. For instance, when we conduct experiments with Liver as the target task, the other three datasets are used as the training set D. Following the benchmark setup of the slice sampling strategy in [11], all the CT stacks used in the following experiments consist of the central CT slices in which the organs of interest lie, of size $256 \times 256 \times C$ ($C \in \{[50, 96], [47, 91], [20, 119], [31, 106]\}$, respectively for the Breast, Kidney, Liver, and Stomach tasks).

1) Constructing Support-Query Image Pairs: Specifically, we propose two different ways to form the support-query image pairs for the training tasks and the target task, respectively.

In the training stage, we adapt the benchmark setup of the slice sampling strategy [11] to form support-query pairs. A distinct extension is that the support images and query images are selected from two different CT stacks (of the same task), which may be more efficient for practical applications.

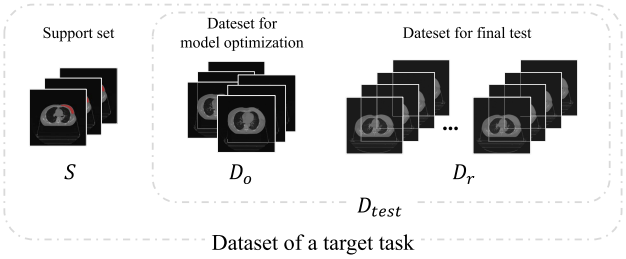

Fig. 5. Illustrating the data split for a target task.

TABLE II
THE NUMBERS OF CT STACKS AND SLICES FOR THE FOUR TASKS. *Img* AND *Slc* DENOTE THE NUMBERS OF CT STACKS AND SLICES IN EACH SUBSET OF THE DATA SPLIT FOR EACH TASK, RESPECTIVELY. *Total* IS THE TOTAL NUMBER OF CT STACKS IN EACH DATASET

Task	$Total$	S $Img\ (Slc)$	D_o $Img\ (Slc)$	D_r Img
Breast	114	1 (10)	20 (20)	93
Kidney	209	1 (10)	20 (20)	188
Liver	131	1 (10)	20 (20)	110
Stomach	153	1 (10)	20 (20)	132

Specifically, we divide each CT stack into K equal-size groups of slices, indexed as 1 to K (starting from the top, as in [11]). Every time we form a support-query pair, two CT stacks are randomly selected. For one of them, a randomly selected slice in the i-th group is the query image; for the other, the center slice of its i-th group is taken as the support image.

For the target task, only K slices of one CT stack are used as support images. Specifically, this CT stack is divided into K groups ($K = 10$ in this work), and the center slice of each group is annotated to form the support set $S = \{(I_s^i, Y_s^i(l))\}_{i=1}^{K=10}$ (indexed from the top, like operations in last paragraph). The other CT stacks of this task form the test set D_{test}. Moreover, we split D_{test} into a small subset $D_o = \{I_e^i\}_{i=1}^{N^*}$ and a subset D_r (as shown in Fig. 5). D_o, with N^* slices randomly selected from N^* CT stacks, is used by our IFSL to optimize MPrNet. D_r, is used for the final test. The data sizes of D_o and D_r for each task are shown in Table II. During both the optimization stage and final test stage, we form the support-query pairs with two steps. First, each CT stack in D_o and D_r is divided into $K = 10$ groups. Thus, given a query image from a CT stack, we can determine exactly to which group of the stack it belongs. Suppose a query image is from the i-th group of a stack (indexed from the top). Then we select from S the i-th support image I_s^i to form a support-query pair. In this fashion, the few-shot segmentation models can work as they were trained, by receiving support-query pairs from the matching groups of different CT stacks. In clinical applications, additional manual interaction may be needed to indicate the start and ending slices of a target organ in a CT stack.

2) Implementation Details: For the training tasks, we use a naive Dice loss to train our MPrNet, for 50 epochs with a batch size 1. With an initial learning rate of le-3, we reduce it using a polynomial decay learning rate policy. We use the mini-batch Stochastic Gradient Descent (SGD) as the optimizer, with momentum = 0.9 and weight decay = le-4.

During model optimization using our IL-TTOA, we simply simulate user corrections as in [62], which assumes the user clicking on the largest error region. We determine this error region by comparing the model predictions with the ground-truth and selecting its center pixel. Then we expand it with a 4×4 kernel. In this work, for all the optimization experiments with our IL-TTOA, we optimize the trained models sequentially on $T = 20$ slices in D_o, and uniformly conduct $Iter = 10$ iterations on each slice without breaking (20 slices are randomly ordered). This means that 10×20 artificial points are interactively added as additional supervision and corrections (as shown in Fig. 6). For our $\mathcal{L}_{\mathcal{O}}$ loss, we down-sample the predicted masks by 4×4 maxpooling before calculating the values of Ω, and the hyper-parameter λ is 0.2 (since the comparison experiments indicate that $\lambda = 0.2$ works best). The Dice-Sørensen Coefficient (DSC) metric is used to measure the performance of all the methods in the experiments.

C. Results and Analysis

1) Comparison With State-of-the-Art Few-Shot Segmentation Models: Table III compares our MPrNet and IFSL approach with several state-of-the-art few-shot segmentation models. For fair comparison, all the experiments split data and form support-query pairs following the setups discussed above. All the compared few-shot segmentation models are implemented as in the original papers.

The first observation on Table III is that our MPrNet and our IFSL approach outperform all the state-of-the-art few-shot segmentation models on each task, and yield higher average DSC on the four tasks. Despite the success of Co-FCN [5] and SG-One [6] in few-shot segmentation on natural images, they show poor performance on our medical datasets. This is possibly due to the big difference in nature between non-medical and medical images and the lack of pre-trained models on medical images [13]. For the Liver and Kidney segmentation tasks (also the target tasks in [11]), the performances of sSENet are not exactly the same as the results presented in the original paper, due to the different training tasks and the datasets that we use. Second, we notice that our MPrNet performs much better on some tasks (Liver and Stomach), but a little poorly on the others (the DSC values are lower than 60% in DSC). In all likelihood, this is inevitable since there is no overlap between the training and target tasks, so that the ability and knowledge that MPrNet obtains in the training stage may not match well with the target task. For instance, MPrNet may obtain an ability to pay more attention to texture features during training, but spatial features are more important for the target task. Hence, we think that the performance of MPrNet will be further improved if we could balance the impacts of the Texture Branch and Spatial Branch in accordance with the target data. In this case, IL-TTOA performs as a complement to assist a trained model to adapt to the target task. As shown in Table III, with the help

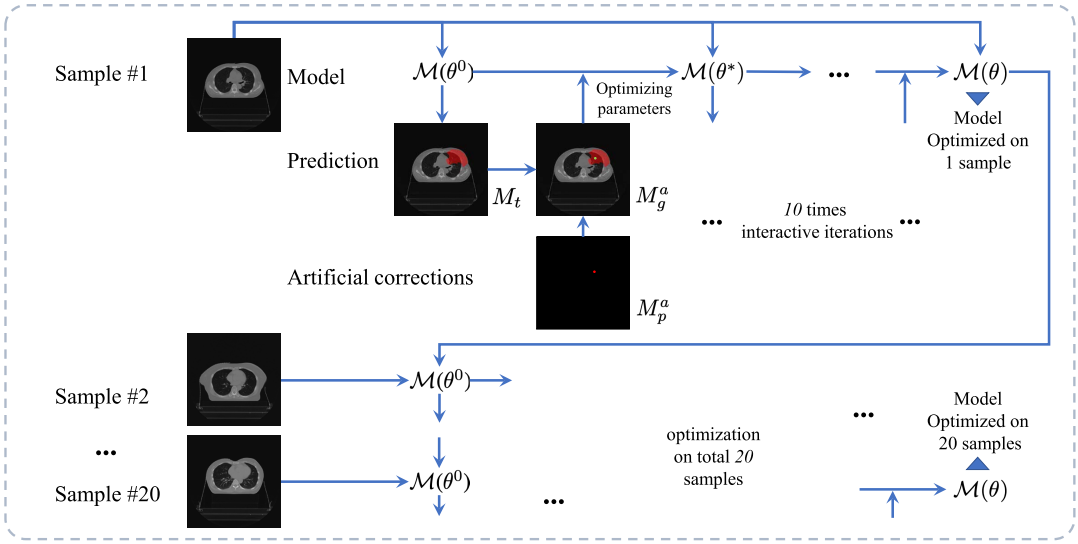

Fig. 6. Visualization of how IL-TTOA works, following Algorithm 1. We perform this algorithm on 20 slices in D_o to strengthen MPrNet on the target task. For each sample, we conduct gradient descent 10 times to optimize the trained model, directed by interactive human corrections.

TABLE III

COMPARISON OF OUR IFSL AND REPRESENTATIVE KNOWN MODELS ON FOUR FEW-SHOT SEGMENTATION TASKS (PER-TASK AND MEAN DICE SCORE). OUR MPRNET OUTPERFORMS ALL THESE FEW-SHOT SEGMENTATION MODELS, AND IFSL ACHIEVES STATE-OF-THE-ART PERFORMANCE

Method	Breast (%)	Kidney (%)	Liver (%)	Stomach (%)	Mean (%)	Annotation Data
Co-FCN [5]	-	5.40	6.00	5.94	4.33	10 slices
SG-One [6]	-	4.90	1.90	4.67	2.87	10 slices
PANet [9]	28.50	38.89	50.88	39.59	39.46	10 slices
sSENet [11]	30.96	31.52	34.23	43.36	35.02	10 slices
MPrNet (Ours)	**47.49**	**49.87**	**65.62**	**60.41**	**55.85**	10 slices
IFSL (Ours)	**56.93**	**58.40**	**75.11**	**67.51**	**64.49**	10 slices + 10 × 20 points
U-Net [1]	80.64	85.00	87.28	84.37	84.32	slice # in [18928, 36120]

of IL-TTOA, MPrNet can be optimized quickly on a small set (e.g., 20 slices) for the target task and achieve a big improvement (nearly 10% in DSC on each task).

Fig. 7 shows several randomly selected samples (in dataset D_r), with the ground truth masks and segmentation results by different models. For better visualization, we perform simple post-processing on all the segmentation results, i.e., dilation, obtaining the largest connected regions, and erosion. Compared to sSENet and PANet, our MPrNet appears to perform better on all the tasks. Obviously, our IFSL approach achieves the most promising performance with respect to the ground truth masks, comparing to all the other models.

Besides, we conduct an experiment as a soft reference to show how a commonly used supervised deep learning model (the model is trained on a large-scale dataset in a fully supervised manner) performs on these four datasets. Since U-Net [1] is broadly used as a strong architecture for medical image segmentation, we conduct the experiment with U-Net in a naive fully supervised manner following [1]. For every segmentation task, we simply train a U-Net model on 80% of the samples of the dataset and test it on the remaining samples (i.e., 20% of the dataset). In other words, four U-Net models are trained on N ($N \in \{91, 168, 105, 122\}$) CT stacks, and each stack consists of n slices (n in the range of [160, 220]), respectively for the Breast, Kidney, Liver, and Stomach tasks (with totally {18928, 36120, 23100, 26230} slices). The rightmost column of Table III reports the annotation data of different methods on the target task. For the few-shot segmentation models, only 10 slices with fully-annotated masks are used, while for the U-Net models, about 20,000 slices are used (the rough slice number 20,000 per task is estimated by multiplying the mean value of N and that of n). Compared to U-Net, our IFSL obtains promising results with greatly less annotation costs on the target tasks, especially Liver and Stomach. On these two tasks, the DSC difference between our approach and the U-Net models is around 12%-17%, while the annotation burden of our approach is far smaller than that of U-Net (almost 1/2000).

Discussion: We should note that in all the scenarios (using few-shot methods, purely training a U-Net for the target segmentation), the models yield the worst performance on Breast among the four tasks. This likely suggests that the Breast task is more difficult than the others. As mentioned above, the specific characteristics of CTV for breast cancer

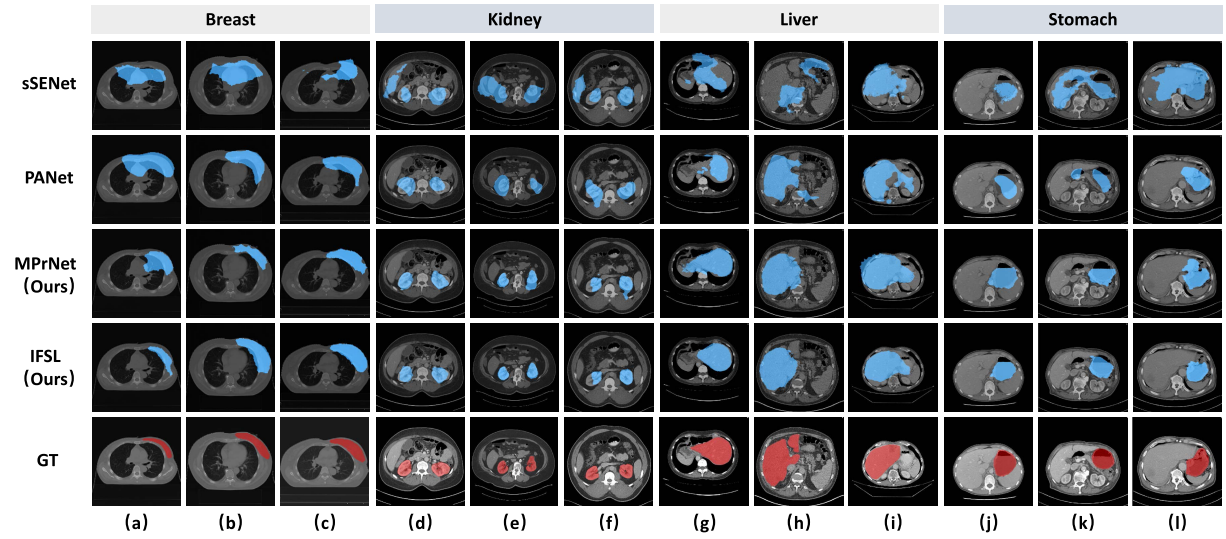

Fig. 7. Segmentation results of different methods on the four tasks. We show three cases for each task, with ground truth mask (GT) and segmentation results by different models: sSENet, PANet, our MPrNet, and our IFSL approach.

radiotherapy (low contrast without clear delineation) make CTV difficult to auto-segment, which would be even tougher with very limited annotation. Certainly, our approach can achieve better performance on this "hard" task, at the cost of slightly increasing the annotation burden on the target task (as reported in the Supplementary Material).

2) Effects of IL-TTOA: In this section, we extend the main idea of our approach, introducing interactive learning into few-shot learning models, to strengthen several well-performing few-shot segmentation models, i.e., PANet and sSENet (respectively denoted by $PANet \oplus IL\text{-}TTOA$ and $sSENet \oplus IL\text{-}TTOA$). Before using the trained models on D_r, we optimize them with our specific IL-TTOA, following the detailed process shown in Fig. 6. We perform the experiments on the four tasks and present the results in Table IV. It is worth noting that with the help of our IL-TTOA, both the models achieve promising improvements. sSENet even attains big improvements of more than 30% in DSC for Liver. However, the contribution of IL-TTOA to our MPrNet is not so outstanding compared to that of these two models (around 7%-10% improvements). We think this may be due to that the basic performance of our MPrNet on these four few-shot segmentation tasks is already much better than the other models. While these promising results demonstrate the effectiveness of our idea, we believe the performance will possibly improve further if we use more data of the target task to optimize MPrNet (as reported in the Supplementary Material).

Fig. 8 presents the performance comparison of the models with and without optimization. For each task, we randomly select one slice from the final test set D_r. Then, we use a trained MPrNet to segment the target in that image and show the segmentation results in the first row. After that, we optimize the model on D_o using IL-TTOA and randomly select the model versions optimized on 1, 8, 16, and 20 slices, to segment the target in the selected slice. The second to the next-to-last rows present the segmentation results. It is worth noting that for each task, the segmentation results are gradually better from top to bottom. These promising results well verify the effectiveness of our IL-TTOA on the four tasks.

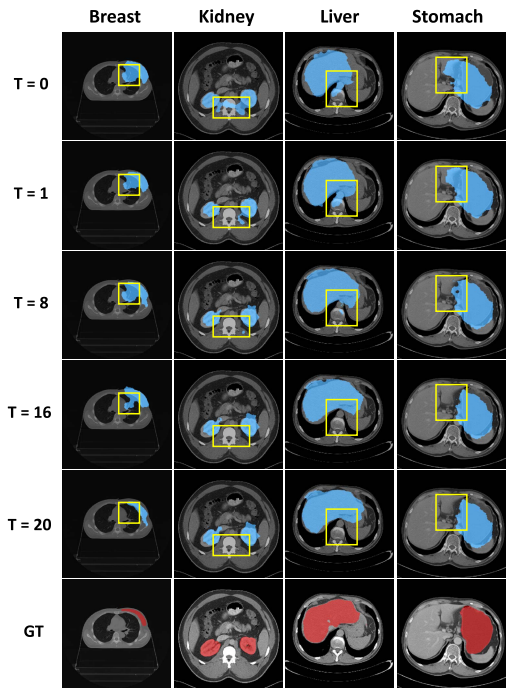

Fig. 8. Visualization of examples randomly selected from the final test set. From top to bottom are prediction results of: trained MPrNet ($\mathcal{M}$), trained MPrNet further strengthened by our IL-TTOA with 1, 8, 16, and 20 slices, and the ground truth masks. The last row is the ground truth masks. From left to right are the examples of the four segmentation tasks: Breast, Kidney, Liver, and Stomach. The yellow boxes indicate some regions that are first incorrectly regressed and then gradually corrected by the strengthened models.

3) Effects of the Individual Components of MPrNet: To examine the effect of each individual component of MPrNet, we conduct ablation study by removing one component each time. We successively pick off the Spatial Branch, Texture

TABLE IV
SEGMENTATION ACCURACY (DSC, %) COMPARISON OF FEW-SHOT SEGMENTATION MODELS WITH AND WITHOUT INTERACTIVE OPTIMIZATION USING OUR IL-TTOA

Method	Breast (%)	Kidney (%)	Liver (%)	Stomach (%)	Mean (%)
PANet	28.50	38.89	50.88	39.59	39.46
PANet $\oplus$ IL-TTOA	40.63	52.44	56.91	61.31	52.82
sSENet	30.96	31.52	34.23	43.36	35.02
sSENet $\oplus$ IL-TTOA	37.85	43.27	64.26	57.28	50.67
MPrNet	47.49	49.87	65.62	60.41	55.85
MPrNet $\oplus$ IL-TTOA	**56.93**	**58.40**	**75.11**	**67.51**	**64.49**

TABLE V
DETAILED PERFORMANCE COMPARISON OF EACH COMPONENT IN OUR MPRNET

Method	Breast (%)	Kidney (%)	Liver (%)	Stomach (%)	Mean (%)
MPrNet $\ominus$ Tex	27.35	40.08	48.80	49.95	41.55
MPrNet $\ominus$ Spa	33.22	37.54	52.14	46.46	42.34
MPrNet $\ominus$ CosF	14.59	34.99	45.86	43.90	34.84
MPrNet (Ours)	**47.49**	**49.87**	**65.62**	**60.41**	**55.85**

Branch, and CosF module to construct three models, denoted by *MPrNet* $\ominus$ *Spa*, *MPrNet* $\ominus$ *Tex*, and *MPrNet* $\ominus$ *CosF* in Table V. Since the experiments in [11] showed the effectiveness of the decoder structure with strong connections, we keep all the models with a decoder structure as the final version of our MPrNet. Note that each component of MPrNet contributes an improvement of 4%-10% to the average Dice score for the four tasks.

Further, we conduct experiments to examine how well the Spatial Branch and Texture Branch concentrate on the diverse characteristics of support images and work complementarily with each other (discussed in Section III-B.3). Fig. 9 gives several examples to visualize the captured features of these two branches using the Gradient-weighted Class Activation Mappings (Grad-CAM) [63]. For each branch, Grad-CAM uses the gradient information flowing into the final convolutional layer (the Bottleneck) to generate a heat map, which highlights the discriminative regions that the branch concerns. More detailed calculation of Grad-CAM was described in [63]. In Fig. 9, the heat maps are overlapped on the original support images and the red regions indicate high attention by the branches.

From Fig. 9, one can see that the regions of high activation in the Spatial Branch differ largely from those in the Texture Branch. This phenomenon indicates that these two branches focus on different regions in a support image to capture diverse characteristics. Specifically, the Spatial Branch concentrates on relatively fixed regions for a task (i.e., the global spatial features of the task), especially in the case of using the same support image. For instance, the examples in Fig. 9(i) and Fig. 9(j) use the same slice as the support image. Observe that, despite the apparent difference of the target shapes in the two query images, the Spatial Branch pays attention to nearly the same regions (indicated in the red boxes). The same can be observed on the examples of Fig. 9(m) and Fig. 9(n). On the contrary, the Texture Branch pays attention to no specific regions, even when having the same support image (e.g., see the examples of Fig. 9(e) and Fig. 9(f)).

As discussed in Section III-B, the Texture Branch and Query Branch have undirected interactions with the weight-sharing operations. Thus, the characteristics of the query images can largely affect feature extraction in the Texture Branch, as indicated by the green boxes in Fig. 9(e). Another example is the yellow boxes in Fig. 9(o); the top-right yellow box possibly indicates that the Texture Branch notices the boundary of the target in the query image, while it seems that the highlighted areas are out of the regions in the support image. Due to the specific structures and interaction fashions of these two branches, the two branches may work complementarily to capture the features of the support images for guiding query segmentation. For instance, in the example of Fig. 9(o), the target shape in the support image is quite different from that in the query image. In this case, the Spatial Branch concentrates on the global spatial features of the support image (as discussed above), while the Texture Branch captures the discriminate and particular support features relevant to the query image (e.g., the region or boundary of the target in the query image), as indicated in the yellow boxes.

4) Sensitivity of the Hyper-Parameter λ: To examine the sensitivity of λ, we empirically conduct the experiments by increasing the value of λ from 0 to 1.5, at a step size of 0.1. Following [58], the effect of λ is estimated in two perspectives: 1) model performance, and 2) model forgetting. For the former, we conduct experiments for our IFSL approach following the aforementioned setups in Section IV-B with different λ values. The mean DSCs for the four tasks are reported in Fig. 10, denoted by Avg.Per. For the latter, we follow the common studies of forgetting estimation in lifelong learning [64] and treat the training tasks as the "previously encountered tasks". Specifically, the forgetting of our IFSL approach on a target task (simply denoted as $Forgetting$) is defined as how much the DSC degrades on the "previously encountered tasks". That means that we obtain the $Forgetting$ value by evaluating MPrNet twice on the "previously encountered tasks", before and after optimized by IL-TTOA on a target task, and computing the difference between these two performances. These performances

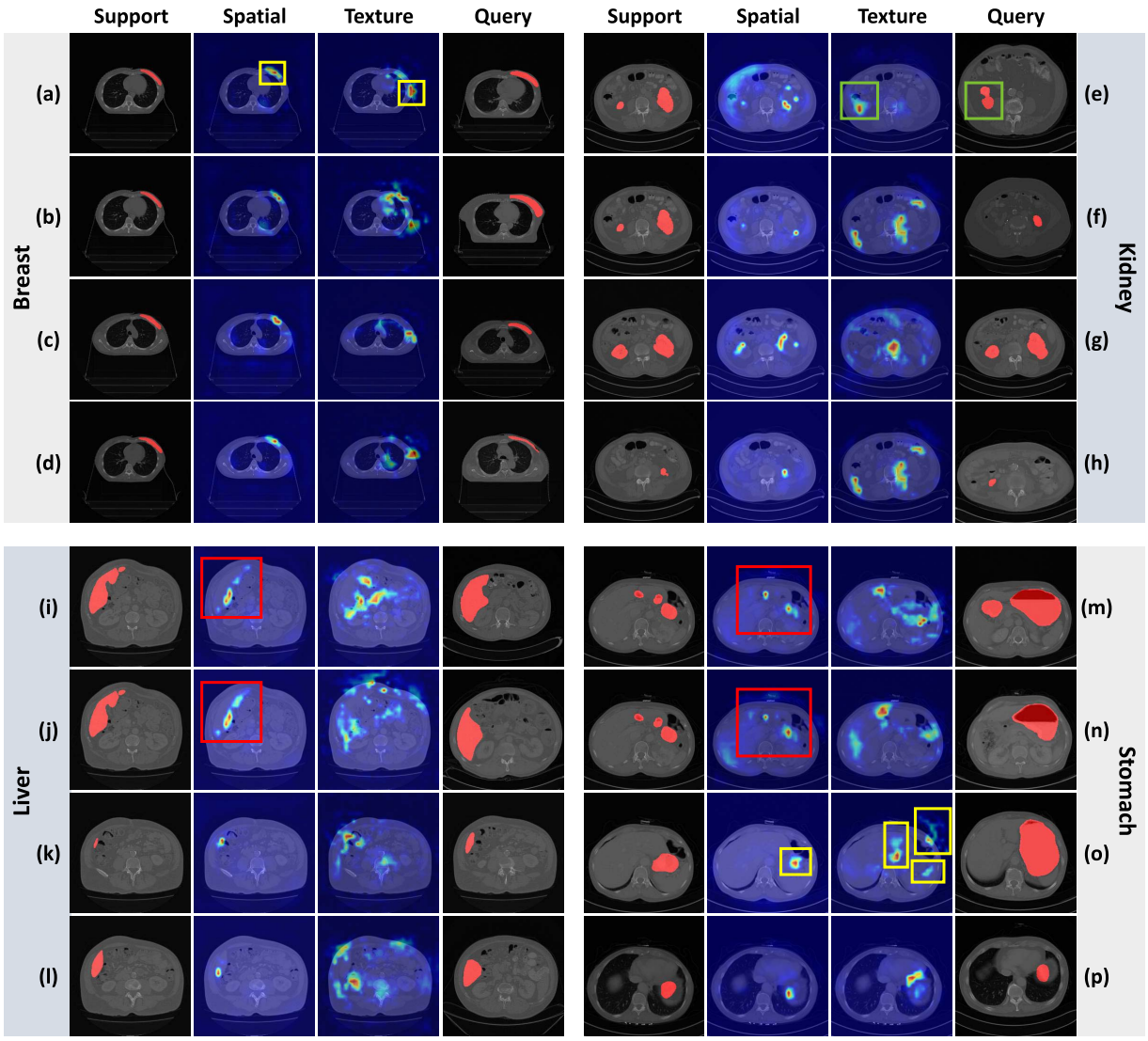

Fig. 9. Visual comparison of the Spatial Branch and Texture Branch (four examples are given for each task). (a)-(d) Breast, (e)-(h) Kidney, (i)-(l) Liver, and (m)-(n) Stomach. The first four columns: the support images, the feature heat maps of the Spatial Branch and Texture Branch, and the query images. The last four columns are for the same. For example, in (a), the support image in column 1 is used to guide the segmentation of the query image in column 4, and the feature heat maps in columns 2 and 3 are obtained by Grad-CAM [63].

are denoted by Per.Old and Per.New, respectively, and the difference is computed by |Per.Old−Per.New|. For a comprehensive estimation, we average the *Forgetting* values of the four tasks to obtain the mean forgetting (denoted as Avg.For in Fig. 10). More detailed setups and experiments are reported in the Supplementary Material.

As shown in Fig. 10, the average performance (Avg.Per) gets better first and then decreases as the λ value increases. The best performance occurs at $\lambda = 0.2$ (DSC = 64.49%). This could be explainable that $\lambda = 0.2$ possibly attains the best balance between the power obtained in the training stage and the impact of additional supervision in IL-TTOA. Thus, we choose $\lambda = 0.2$ in all the experiments in this work. Another observation is that, the mean Forgetting value decreases as λ becomes larger. This phenomenon seems consistent with the theoretical prediction of our $\mathcal{L}_{\mathcal{O}}$ (in Section III-C.2) that

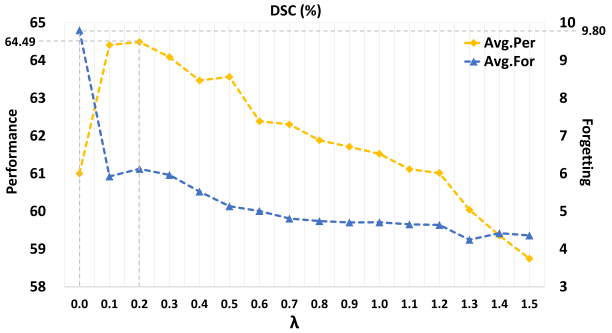

Fig. 10. The performance (left) and forgetting (right) of our IFSL approach with different λ values. The yellow curve indicates the average model performance and the blue curve is for the mean forgetting.

the larger λ is, the more penalty on the parameter changes and the less obtained knowledge allowed to be forgotten

(in particular, $\lambda = 0$ means that our $\mathcal{L}_{\mathcal{O}}$ is equivalent to a naive Dice loss).

V. Discussion and Conclusion

In this paper, for the first time, we incorporated interactive learning into few-shot segmentation methods, to alleviate the annotation burden of traditional supervised deep learning methods and address the common issues of the known few-shot segmentation methods. Our new Interactive Few-shot Learning (IFSL) approach effectively enables few-shot segmentation models to be strengthened on the target tasks, in an interactive and controllable manner. Experimental results demonstrated the superiority of our IFSL approach, which outperforms all the state-of-the-art few-shot segmentation methods by over 20% in DSC (as shown in Section IV-C.1).

Besides our novel idea and notable performance improvements, we find several critical directions for future research. First, in practice, our interactive strategy (IL-TTOA) possibly incurs additional time costs (compared to the naive few-shot segmentation models), since it needs experts to indicate error regions to modify our MPrNet in several iterations. Hence, developing a more effective way to receive and utilize additional supervision on the target tasks is an important direction, which may yield a better trade-off between performance and time costs. Second, other targets in medical images may possibly appear at different locations (e.g., brain tumors in CT stacks or lesions in pathology images). Applying our current approach to such targets may still face a big challenge which we will consider in our future study. Perhaps, with the help of active learning strategies, carefully selecting the most informative samples as support images or for the refinement stage (with IL-TTOA) would be useful in the above two directions.

To conclude, we identified the drawbacks of existing few-shot segmentation methods which lack optimization on the target tasks, and proposed a novel IFSL approach to remedy such drawbacks. To our best knowledge, we are the first to enable few-shot segmentation models to be strengthened on the target tasks with additional interactive supervision. Further, in clinical applications, our new optimization algorithm (IL-TTOA) can strengthen any few-shot deep learning segmentation models on-the-fly, in a human controllable fashion.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[2] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[3] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," 2017, *arXiv:1709.03410*. [Online]. Available: http://arxiv.org/abs/1709.03410

[4] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine, "Few-shot segmentation propagation with guided networks," 2018, *arXiv:1806.07373*. [Online]. Available: http://arxiv.org/abs/1806.07373

[5] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine, "Conditional networks for few-shot semantic segmentation," in *Proc. ICLR Workshop*, 2018.

[6] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "SG-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3855–3865, Sep. 2020.

[7] T. Hu, P. Yang, Z. Chiliang, G. Yu, Y. Mu, and C. Snoek, "Attention-based multi-context guiding for few-shot semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8441–8448.

[8] N. Dong and E. Xing, "Few-shot semantic segmentation with prototype learning," in *Proc. BMVC*, vol. 3, 2018.

[9] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "PANet: Few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9197–9206.

[10] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[11] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "'Squeeze & excite'guided few-shot segmentation of volumetric images," *Med. Image Anal.*, vol. 59, Jan. 2020, Art. no. 101587.

[12] J. Cho, K. Lee, E. Shin, G. Choy, and S. Do, "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?" 2015, *arXiv:1511.06348*. [Online]. Available: http://arxiv.org/abs/1511.06348

[13] N. Tajbakhsh *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[14] T. Zhou, S. Ruan, and S. Canu, "A review: Deep learning for medical image segmentation using multi-modality fusion," *Array*, vols. 3–4, Sep. 2019, Art. no. 100004.

[15] M. Kim, J. Zuallaert, and W. De Neve, "Few-shot learning using a small-sized dataset of high-resolution FUNDUS images for glaucoma diagnosis," in *Proc. 2nd Int. Workshop Multimedia Pers. Health Health Care*, Oct. 2017, pp. 89–92.

[16] P. Zhang, Y. Zhong, Y. Deng, X. Tang, and X. Li, "A survey on deep learning of small sample in biomedical image analysis," 2019, *arXiv:1908.00473*. [Online]. Available: http://arxiv.org/abs/1908.00473

[17] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation," *Med. Image Anal.*, vol. 63, Jul. 2020, Art. no. 101693.

[18] S. E. Whang and J.-G. Lee, "Data collection and quality challenges for deep learning," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 3429–3432, Aug. 2020.

[19] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1153–1159, May 2016.

[20] S. Kadam and V. Vaidya, "Review and analysis of zero, one and few shot learning approaches," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, 2018, pp. 100–112.

[21] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, Jul. 2020.

[22] A. Medela *et al.*, "Few shot learning in histopathological images:Reducing the need of labeled data on biological datasets," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2019, pp. 1860–1864.

[23] X. Jiang, L. Ding, M. Havaei, A. Jesson, and S. Matwin, "Task adaptive metric space for medium-shot medical image classification," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 147–155.

[24] A. Parida, A. Tran, N. Navab, and S. Albarqouni, "Learn to segment organs with a few bounding boxes," 2019, *arXiv:1909.07809*. [Online]. Available: http://arxiv.org/abs/1909.07809

[25] T. Fechter and D. Baltas, "One-shot learning for deformable medical image registration and periodic motion tracking," *IEEE Trans. Med. Imag.*, vol. 39, no. 7, pp. 2506–2517, Jul. 2020.

[26] S. Puch, I. Sánchez, and M. Rowe, "Few-shot learning with deep triplet networks for brain imaging modality recognition," in *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*. Cham, Switzerland: Springer, 2019, pp. 181–189.

[27] K. Sirinukunwattana *et al.*, "Gland segmentation in colon histology images: The glas challenge contest," *Med. Image Anal.*, vol. 35, pp. 489–502, Jan. 2017.

[28] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic brain tumor detection and segmentation using U-net based fully convolutional networks," in *Proc. Annu. Conf. Med. Image Understand. Anal.* Cham, Switzerland: Springer, 2017, pp. 506–517.

[29] D. J. Ho, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Nuclei segmentation of fluorescence microscopy images using three dimensional convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 82–90.

[30] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8543–8553.

[31] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2016, pp. 424–432.

[32] C. Fu *et al.*, "Three dimensional fluorescence microscopy image synthesis and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 2221–2229.

[33] H.-C. Shin *et al.*, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *Proc. Int. Workshop Simulation Synth. Med. Imag.* Cham, Switzerland: Springer, 2018, pp. 1–11.

[34] Y. Tang *et al.*, "CT image enhancement using stacked generative adversarial networks and transfer learning for lesion segmentation improvement," in *Proc. Int. Workshop Mach. Learn. Med. Imag.* Cham, Switzerland: Springer, 2018, pp. 46–54.

[35] Y.-B. Tang, S. Oh, Y.-X. Tang, J. Xiao, and R. M. Summers, "CT-realistic data augmentation using generative adversarial network for robust lymph node segmentation," *Proc. SPIE*, vol. 10950, Mar. 2019, Art. no. 109503V.

[36] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2017, pp. 399–407.

[37] J. Sourati, A. Gholipour, J. G. Dy, S. Kurugol, and S. K. Warfield, "Active deep learning with Fisher information for patch-wise semantic segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 83–91.

[38] F. Ozdemir, Z. Peng, C. Tanner, P. Fuernstahl, and O. Goksel, "Active learning for segmentation by optimizing content information for maximal entropy," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 183–191.

[39] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, "Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7340–7351.

[40] J. Folmsbee, X. Liu, M. Brandwein-Weber, and S. Doyle, "Active deep learning: Improved training efficiency of convolutional neural networks for tissue classification in oral cavity cancer," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 770–773.

[41] A. Smailagic *et al.*, "MedAL: Accurate and robust deep active learning for medical image analysis," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 481–488.

[42] G. Wang *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE Trans. Med. Imag.*, vol. 37, no. 7, pp. 1562–1573, Jul. 2018.

[43] G. Wang *et al.*, "DeepIGeoS: A deep interactive geodesic framework for medical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1559–1572, Jul. 2019.

[44] F. Zhao and X. Xie, "An overview of interactive medical image segmentation," *Ann. BMVA*, vol. 2013, no. 7, pp. 1–22, 2013.

[45] V. Cheplygina, M. de Bruijne, and J. P. W. Pluim, "Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Med. Image Anal.*, vol. 54, pp. 280–296, May 2019.

[46] H.-C. Shin *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[47] S. Mehta, E. Mercan, J. Bartlett, D. Weaver, J. G. Elmore, and L. Shapiro, "Y-Net: Joint segmentation and classification for diagnosis of breast biopsy images," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 893–901.

[48] Y.-J. Huang *et al.*, "3-D RoI-aware U-net for accurate and efficient colorectal tumor segmentation," *IEEE Trans. Cybern.*, early access, Apr. 1, 2020, doi: 10.1109/TCYB.2020.2980145.

[49] L. Sun, Z. Fan, X. Ding, Y. Huang, and J. Paisley, "Joint CS-MRI reconstruction and segmentation with a unified deep network," in *Proc. Int. Conf. Inf. Process. Med. Imag.* Cham, Switzerland: Springer, 2019, pp. 492–504.

[50] Q. Li, A. Arnab, and P. H. Torr, "Weakly-and semi-supervised panoptic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 102–118.

[51] X. Feng, J. Yang, A. F. Laine, and E. D. Angelini, "Discriminative localization in CNNs for weakly-supervised segmentation of pulmonary nodules," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2017, pp. 568–576.

[52] D. Nie, Y. Gao, L. Wang, and D. Shen, "ASDNet: Attention based semi-supervised deep networks for medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 370–378.

[53] W. Bai *et al.*, "Semi-supervised learning for network-based cardiac MR image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2017, pp. 253–260.

[54] D. Mahapatra, "Semi-supervised learning and graph cuts for consensus based medical image segmentation," *Pattern Recognit.*, vol. 63, pp. 700–709, Mar. 2017.

[55] Y. Zhang, L. Yang, J. Chen, M. Fredericksen, D. P. Hughes, and D. Z. Chen, "Deep adversarial networks for biomedical image segmentation utilizing unannotated images," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2017, pp. 408–416.

[56] Z. Zhou, J. Y. Shin, S. R. Gurudu, M. B. Gotway, and J. Liang, "Active, continual fine tuning of convolutional neural networks for reducing annotation efforts," 2018, *arXiv:1802.00912*. [Online]. Available: http://arxiv.org/abs/1802.00912

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[58] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 139–154.

[59] N. Heller *et al.*, "The KiTS19 challenge data: 300 kidney tumor cases with clinical context, CT semantic segmentations, and surgical outcomes," 2019, *arXiv:1904.00445*. [Online]. Available: http://arxiv.org/abs/1904.00445

[60] P. Bilic *et al.*, "The liver tumor segmentation benchmark (LiTS)," 2019, *arXiv:1901.04056*. [Online]. Available: http://arxiv.org/abs/1901.04056

[61] K. Men *et al.*, "Fully automatic and robust segmentation of the clinical target volume for radiotherapy of breast cancer using big data and deep learning," *Phys. Medica*, vol. 50, pp. 13–19, Jun. 2018.

[62] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 373–381.

[63] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[64] Q. She *et al.*, "OpenLORIS-object: A robotic vision dataset and benchmark for lifelong deep learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4767–4773.