



Interactive medical image segmentation with self-adaptive confidence calibration*#

Chuyun SHEN^{†1}, Wenhao LI^{†1}, Qisen XU¹, Bin HU², Bo JIN¹, Haibin CAI³,
 Fengping ZHU², Yuxin LI², Xiangfeng WANG^{†‡1}

¹School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

²Huashan Hospital, Fudan University, Shanghai 200040, China

³Software Engineering Institute, East China Normal University, Shanghai 200062, China

[†]E-mail: cyshe@stu.ecnu.edu.cn; 52194501026@stu.ecnu.edu.cn; xfwang@cs.ecnu.edu.cn

Received July 13, 2022; Revision accepted Feb. 20, 2023; Crosschecked Aug. 24, 2023

Abstract: Interactive medical image segmentation based on human-in-the-loop machine learning is a novel paradigm that draws on human expert knowledge to assist medical image segmentation. However, existing methods often fall into what we call interactive misunderstanding, the essence of which is the dilemma in trading off short- and long-term interaction information. To better use the interaction information at various timescales, we propose an interactive segmentation framework, called interactive MEDical image segmentation with self-adaptive Confidence CALibration (MECCA), which combines action-based confidence learning and multi-agent reinforcement learning. A novel confidence network is learned by predicting the alignment level of the action with short-term interaction information. A confidence-based reward-shaping mechanism is then proposed to explicitly incorporate confidence in the policy gradient calculation, thus directly correcting the model's interactive misunderstanding. MECCA also enables user-friendly interactions by reducing the interaction intensity and difficulty via label generation and interaction guidance, respectively. Numerical experiments on different segmentation tasks show that MECCA can significantly improve short- and long-term interaction information utilization efficiency with remarkably fewer labeled samples. The demo video is available at <https://bit.ly/mecca-demo-video>.

Key words: Medical image segmentation; Interactive segmentation; Multi-agent reinforcement learning; Confidence learning; Semi-supervised learning

<https://doi.org/10.1631/FITEE.2200299>

CLC number: TP391.4

1 Introduction

Medical image segmentation is a crucial task in computer-assisted medical diagnosis. However, traditional convolutional neural networks (CNNs) (Wang et al., 2018, 2019; Liao et al., 2020) used in segmentation algorithms often struggle to meet clinical standards due to the variability of pathological conditions, the presence of dark lesion areas, and the inconsistent quality of training data. Factors such as discrepancies in imaging scanners, operators, and annotators make the task of medical image segmentation more challenging. To improve accuracy,

[‡] Corresponding author

* Project supported by the Science and Technology Commission of Shanghai Municipality, China (No. 22511106004), the Postdoctoral Science Foundation of China (No. 2022M723039), the National Natural Science Foundation of China (No. 12071145), and the Shanghai Trusted Industry Internet Software Collaborative Innovation Center, China

Electronic supplementary materials: the online version of this article (<https://doi.org/10.1631/FITEE.2200299>) contains supplementary materials, which are available to authorized users

ORCID: Chuyun SHEN, <https://orcid.org/0009-0001-3622-1193>; Wenhao LI, <https://orcid.org/0000-0003-2985-1098>; Xiangfeng WANG, <https://orcid.org/0000-0003-1802-4425>

© Zhejiang University Press 2023

interactive image segmentation algorithms (Lin et al., 2016; Xu et al., 2016; Rajchl et al., 2017; Bredell et al., 2018; Wang et al., 2018, 2019; Liao et al., 2020; Ma et al., 2021) have been developed using interactive correction information, such as clicks, scribbles, or bounding boxes, to refine the results.

Fig. 1 illustrates the general interactive segmentation process, which contains two modules, i.e., an interactive module and a utilization module. In the interactive module, users or experts provide interactive correction information, such as clicks and scribbles. The utilization module then uses this interactive correction information to efficiently refine the previous result. Interactive segmentation algorithms have the potential to outperform traditional algorithms by incorporating additional interactive correction information. The interaction process can be viewed as a continuous collaboration between the model and a human expert. Therefore, an effective interactive segmentation model should be able to incorporate and respond to expert feedback to improve its performance through collaboration.

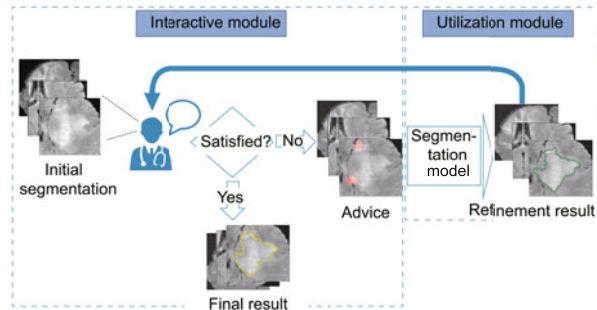


Fig. 1 Interactive segmentation process. In the interactive module, the expert observes the current (or initial) segmentation and provides correction information (the red hint); in the utilization module, new segmentation is refined based on the correction information. References to color refer to the online version of this figure

Methods such as BIFSeg (Wang et al., 2018) and DeepIGeoS (Wang et al., 2019) model user interactions as hard constraints through conditional random fields. However, these models focus more on one-step interaction, and refine segmentation after the first step cannot be efficiently used (Liao et al., 2020; Ma et al., 2021). Thus, these models cannot use long-term interaction information efficiently. The follow-up InterCNN (Bredell et al., 2018) and other methods (Liao et al., 2020; Ma et al., 2021)

improve DeepIGeoS and BIFSeg, and model the problem as an iterative interaction problem, which is more focused on multi-step interaction. Therefore, these methods can cover the data distribution corresponding to different interaction levels in the training phase to effectively use long-term interactions, while ignoring the stochasticity or uncertainty of the model, which makes it difficult for them to use short-term interaction information effectively.

Existing interactive methods cannot effectively use short- and long-term interaction information simultaneously, which leads to interactive misunderstanding. Fig. 2 presents this phenomenon when implementing the popular interactive segmentation algorithm InterCNN (Bredell et al., 2018) on the BraTS2015 dataset. The algorithm ignores the expert's correction information (Fig. 2a), and is even adversely affected (Fig. 2b). These inconsistencies between hint information and the refinement results indicate that existing algorithms still face the critical challenge of inefficient utilization of interactive correction information. The reason for the segmentation refining failure is that, at the end of the interaction process, the total loss of the model will guarantee the main area's priority and ignore some small but challenging areas, such as edges. The regions that are insensitive to interactive correction information could be considered hard-to-segment regions (Shrivastava et al., 2016; Nie et al., 2019), leading to suboptimal refinement of the segmentation process. If these regions are not given additional attention during the training phase, the segmentation model may prioritize easy-to-segment areas at the expense of these more challenging regions. This problem is more severe for medical images, where the hard-to-segment regions usually are tumor boundaries and are very important for clinical diagnosis and surgery. Therefore, it has become urgent to improve the utilization of correction information, especially for the hard-to-segment regions.

Another significant challenge in the development of interactive segmentation algorithms is the need for a large number of labeled images. Accurate annotation of medical images can be time-consuming and costly, making it difficult to obtain a sufficient quantity of high-quality data. On the other hand, it is much more cost-effective to obtain a larger number of unlabeled images, but existing interactive segmentation methods do not use these low-cost resources.

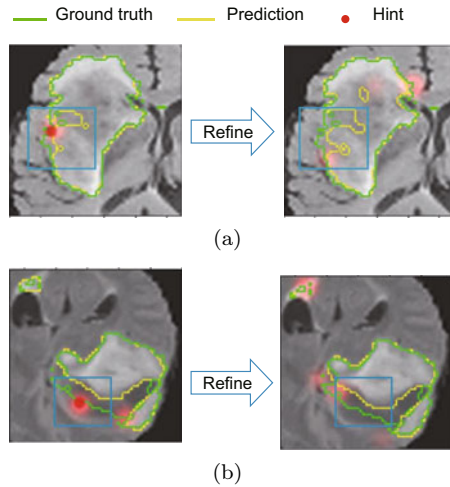


Fig. 2 Segmentation refining failure after long-term interactions on BraTS2015 with InterCNN: (a) the model cannot fully understand or ignore the hint information; (b) the model could misunderstand the correction, which results in a worse result

As a result, it is important to explore how to use the benefits of unlabeled data to reduce the dependence on expert-annotated images.

In this paper, we propose a novel interactive segmentation algorithm for three-dimensional (3D) medical images, called interactive MEDical image segmentation with self-adaptive Confidence CALibration (MECCA). MECCA combines an action-based confidence learning network with the multi-agent reinforcement learning (MARL) framework (Zhang KQ et al., 2021). The action-based confidence network can evaluate the corrective action quality by directly calculating the confidence. Unlike learning the confidence of the overall segmentation result, the confidence of actions could be used to better evaluate whether the segmentation model has correctly used the experts' interaction information. We formulate the iterative interaction process as a Markov decision process (MDP) to model the dynamic process and further introduce the MARL technique (Lee and Song, 2018). Further, instead of setting each image or each patient (a series of images) as the agent, we consider each voxel (the smallest unit in the 3D space) as an agent, while each agent aims to learn the segmentation policy and makes its decision (Furuta et al., 2020). Specifically, after receiving the interactive correction information, each agent will modify its label by changing (increasing or decreasing) the category probability. The novel action-based confidence network will directly evaluate each agent's

corrective action obtained from the MARL corrective policy. Using this action-based confidence network, two additional techniques are further proposed to improve the utilization efficiency of the interactive correction information: (1) compared with former manual rewards, a self-adaptive reward function of each action is constructed which could provide more meticulous feedback under a flexible framework; (2) a simulated label generation mechanism is established using the interactive correction information as the weakly supervisory signal. Combining the confidence network, the simulated label generation mechanism can approximately generate the labels for unsupervised images and reduce over-reliance on labeled data.

The main contributions of this work are summarized as follows:

1. A novel framework is proposed for interactive medical image segmentation by combining the action-based confidence learning network with MARL.
2. A self-adaptive feedback mechanism (self-adaptive reward) is constructed with the action-based confidence network to alleviate interactive misunderstanding during the interaction process.
3. Simulated supervisory signals can be generated based on the confidence learning network and actions; hence, much fewer labeled data (ground truth data) are needed to achieve the same performance.

2 Related works

2.1 Deep interactive image segmentation

Xu et al. (2016) segmented images based on CNN interactively. DeepCut (Rajchl et al., 2017) and ScribbleSup (Lin D et al., 2016) both employ weak supervision to establish interactive image segmentation methods. DeepIGeoS (Wang et al., 2019) employs the geodesic distance metric to construct a hint map. The interactive segmentation process can be considered a sequential process which is natural to model by reinforcement learning (RL). Polygon-RNN (Castrejón et al., 2017) fundamentally segments each target as a polygon and iteratively chooses the polygon vertexes through a recurrent neural network (RNN). Polygon-RNN++ (Acuna et al., 2018) employs almost the same idea

as Polygon-RNN but learns to choose vertexes by RL. SeedNet (Lee and Song, 2018) trains an expert interaction generation RL model that obtains new simulated interaction information at each interaction step. IteR-MRL (Liao et al., 2020) and BS-IRIS (Ma et al., 2021) both model the dynamic interaction process as an MDP and employ MARL models to segment images. Some researchers aimed to reduce the annotation cost of interactive image segmentation. IFSL (Feng et al., 2021) introduces interactive learning into the few-shot learning strategy and addresses the annotation burden of medical image segmentation models. IOG (Zhang SY et al., 2020) uses a practical inside-outside guidance approach to minimize the labeling cost. These interactive methods can hardly use experts' short- and long-term interaction information simultaneously, thus creating error correction operations.

2.2 Uncertainty image segmentation

Uncertainty estimation is helpful in the context of deployed machine learning systems because it can detect when a neural network is likely to make an incorrect prediction or when the input may be out of distribution. Recent methods generally adopt the neural network to learn what the uncertainty should be for any given input, i.e., learning-based uncertainty estimation or confidence learning, as demonstrated in Kendall and Gal (2017), Moeskops et al. (2017), DeVries and Taylor (2018b), Hung et al. (2018), Robinson et al. (2018), Jungo and Reyes (2019), and Nie et al. (2019). These methods commonly consist of a segmentation network and a confidence network, and are more computationally efficient than other techniques. Thus, they are better suited when computational resources are limited or when real-time inference is required, such as the interactive segmentation scenario considered in this paper. Specifically, Kendall and Gal (2017) introduced a confidence network to predict the aleatoric and epistemic uncertainties by imitating classic Bayesian tools. The segmentation network of DeVries and Taylor (2018a, 2018b) produces two separate outputs: prediction probabilities and a confidence estimate. Confidence estimate is motivated by interpolation between the predicted probability distribution and the target distribution during training, where the degree of interpolation is proportional to the confidence estimate. A series of

works (Moeskops et al., 2017; Hung et al., 2018; Nie et al., 2019) focus on incorporating uncertainty estimation in the adversarial learning process, where the segmentation network corresponds to the generator, and the confidence network is the discriminator accordingly. Moeskops et al. (2017) first employed generative adversarial networks (GANs) to improve the CNN-based brain magnetic resonance imaging (MRI) segmentation method. The semi-supervised learning technique was used in Hung et al. (2018) to predict trustworthy regions in unlabeled images. Nie et al. (2019) proposed a difficulty-aware attention mechanism to handle those difficult samples or challenging regions. Different from the previous work on learning uncertainty through imitation, joint training, or adversarial learning, a simple but powerful alternative is to introduce an auxiliary task, such as predicting the overlap between a proposed segmentation and its ground truth (Robinson et al., 2018), or to predict the voxel-wise false positive and false negative (Jungo and Reyes, 2019).

2.3 Remark

In our proposed algorithm, the confidence network should evaluate the confidence of calibrating actions instead of the segmentation results, which is the most significant difference between MECCA and previous uncertainty estimation methods. Therefore, we design a novel action-oriented auxiliary task to predict whether the direction of voxel-wise action is consistent with the ground truth.

3 Proposed algorithm

This section introduces the proposed interactive segmentation algorithm, MECCA, which can iteratively evaluate the refinement actions and feedback to the segmentation model. The algorithm framework follows the MARL structure (Sections 3.1 and 3.2), and an action-based confidence learning module (Section 3.3) is introduced to evaluate the confidence of corrective actions. This action-based confidence learning module can be used to establish the self-adaptive reward scheme and simulated label generation mechanism to use the interactive correction information efficiently. The architecture overview of MECCA is depicted in Fig. 3a. The model's state information includes the original 3D

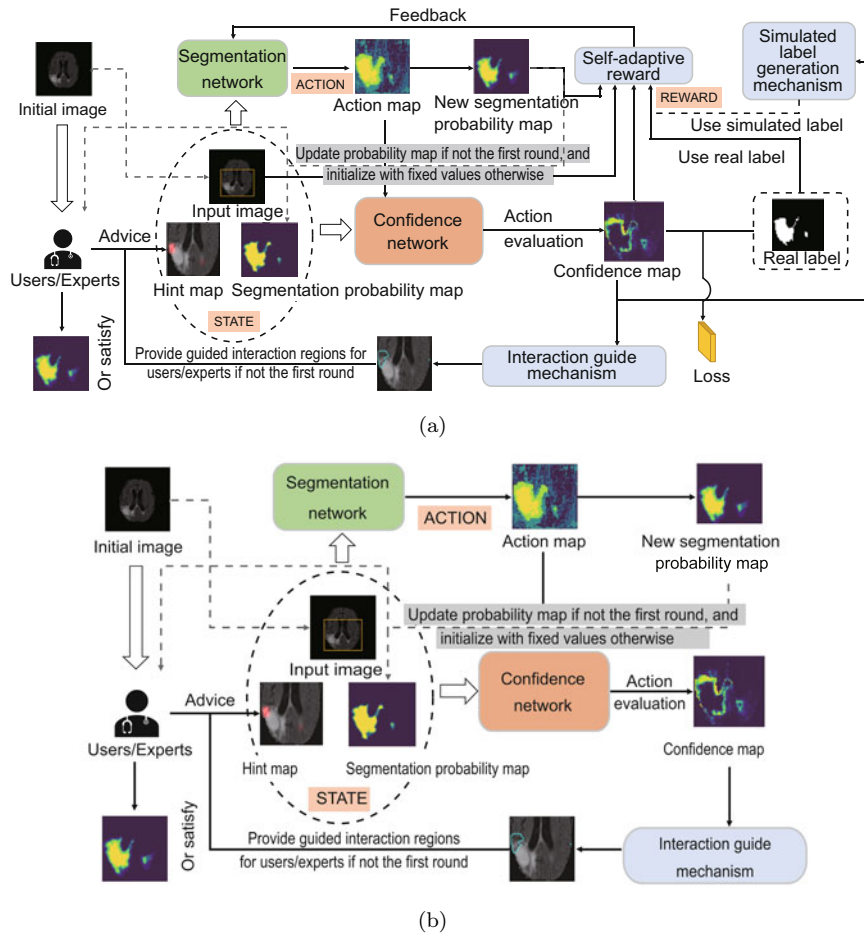


Fig. 3 The architecture (a) and testing stage (b) of MECCA. In (a), the segmentation module outputs actions to change the segmentation probability of each voxel (agent) at each interaction step. Meanwhile, the confidence network will estimate the confidence of the actions, which will generate the self-adaptive reward and simulated label. The confidence map can provide the advice regions of the next interaction step to experts. In (b), all voxels in the segmentation probability map are initialized at a fixed value (set to 0.5 in the experiment). Users or experts randomly mark hint points according to the initialized segmentation probability map

image, the previous segmentation probability, and the hint map generated from interaction and confidence maps. Based on the current state information, the segmentation module gives suggested actions to refine previous segmentation results by adjusting the segmentation probability of each voxel (agent). Further, the state information will be used to evaluate the confidence of the obtained actions through the confidence network, with a confidence map as the output. The self-adaptive reward is designed through a self-adaptive weighting scheme based on the action confidence evaluation (Section 3.4). The self-adaptive reward map can be considered a value map with the same size as the original input image (each agent has its self-adaptive reward). It can re-

fect the performance of the corresponding agent's action. In addition, MECCA will suggest some low-confidence regions for the user to segment during the interaction (Section 3.5). The confidence map can also be used to generate the simulated label by comparison with actions, which will be described in Section 3.6. The newly obtained hint map, the adjusted segmentation probability result, and the original 3D image form a new state. The process described above is repeated until the segmentation result meets the requirements. To emphasize, during the testing stage in Fig. 3b, there is no need to calculate the self-adaptive rewards, and at the same time, we need only the obtained actions and suggested interaction areas.

3.1 MARL-driven interactive segmentation

We employ the MARL structure to formulate the interactive segmentation process and continuously give error-corrective actions at each interaction step. Let $x = (x_1, x_2, \dots, x_N)$ denote the input image, where x_i ($i = 1, 2, \dots, N$) denotes the i^{th} voxel of the image. In MARL setting, every voxel x_i is treated as an agent with its own refinement policy $\pi_i(a_i^{(t)}, s_i^{(t)})$. At time step t , agent x_i obtains action $a_i^{(t)}$ from the segmentation network according to its current state $s_i^{(t)}$. After taking the action, the agent will receive a reward $r_i^{(t)}$ according to the segmentation result.

The state $s_i^{(t)}$ for agent x_i is concentrated by its corresponding image's voxel value b_i , current segmentation probability $p_i^{(t)}$, and value $h_i^{(t)}$ on the hint map. In particular, the segmentation probabilities of all agents are initialized to 0.5 and range from 0 to 1. The hint map $h^{(t)}$ is transformed from the user's hints, which are in the form of edge points. At each step, users click on some edges, which are not correctly predicted, as hints. To let the model receive hint information, we generate a 3D Gaussian (with a kernel size of eight voxels) centered on each of the edge points as the hint map input to the segmentation network.

The action $a_i^{(t)}$ for agent x_i is sampled from its policy and used to adjust its previous segmentation:

$$a_i^{(t)} \sim \pi_\theta(a_i^{(t)} | s_i^{(t)}), \quad (1)$$

$$p_i^{(t+1)} = \text{clip}(p_i^{(t)} + a_i^{(t)}, 0, 1), \quad (2)$$

where $a_i^{(t)} \in \mathcal{A}$ and the clip operation modifies the probability to the interval $[0, 1]$. The action set \mathcal{A} contains actions of different scales, allowing the agent to select the proper action. In our setting, $\mathcal{A} = \{\pm 0.1, \pm 0.2, \pm 0.4\}$. Zero is not used as one of the actions, mainly for the following three reasons: (1) In the early stages of the experiment, we find that using 0 as an action will make the algorithm converge slowly, and it requires more than twice the number of interactions to achieve the performance of baselines. Therefore, we remove 0 from the action space and encourage the agent to explore in the early stage of training. Not only does the algorithm converge faster, but it also requires fewer interactions. (2) To prevent the algorithm's output from exceeding the valid range, we clip the output action

(this is also a common technique in RL to ensure that the output of the policy is in the valid range), as shown in Eq. (2). (3) Due to the existence of the self-adaptive confidence calibration mechanism, the action output by the algorithm will generally change in the direction of the correct label. Combined with the clipping technique used, the algorithm can guarantee effectual output even if there is no 0-action.

The reward $r_i^{(t)}$ is the feedback (positive or negative) of the action and is used to update the refinement policy. We will introduce it in Section 3.4.

3.2 Segmentation network

The segmentation network adopts P-Net (Wang et al., 2019) as the backbone. It has two heads: a policy head and a value head. Two heads share the first three 3D convolutional blocks to extract low-level features. Each of the blocks has two convolution layers, and the size of the convolution kernel is fixed as 3×3 in all these convolution layers. Dilated convolution is employed in all the convolution kernels, which can reduce the loss of resolution. Both heads have another two 3D convolutional blocks to extract specific high-level features. Fig. 4 shows the detailed architecture. Specifically, the policy head outputs policy $\pi_i(a_i^{(t)}, s_i^{(t)})$, which is the distribution of action probabilities under the current state. By taking actions on different scales, the probabilities will be dynamically adjusted. The value head estimates the value of the current state, reflecting how good the current state is and estimating the expected return:

$$V_i^{(t)} = \mathbb{E} \left[R_i^{(t)} | s_i = s_i^{(t)} \right] = \mathbb{E} \left[\sum_{k=t}^T \gamma^{k-t} r_i^{(k)} \right], \quad (3)$$

where T is the terminal time step in the interaction process, and γ is the discount factor. θ_v denotes the parameters of the value head, and the gradient with respect to θ_v is

$$d_{\theta_v} = \nabla_{\theta_v} A^2(s^{(t)}, a^{(t)}), \quad (4)$$

where

$$A(s^{(t)}, a^{(t)}) = \sum_{k=t}^T \gamma^{k-t} \bar{r}^{(k)} - V(s^{(t)}),$$

and $\bar{r}^{(k)}$ is the mean reward of all voxels at time step k . $A(s^{(t)}, a^{(t)})$ represents the advantage function (Mnih et al., 2016).

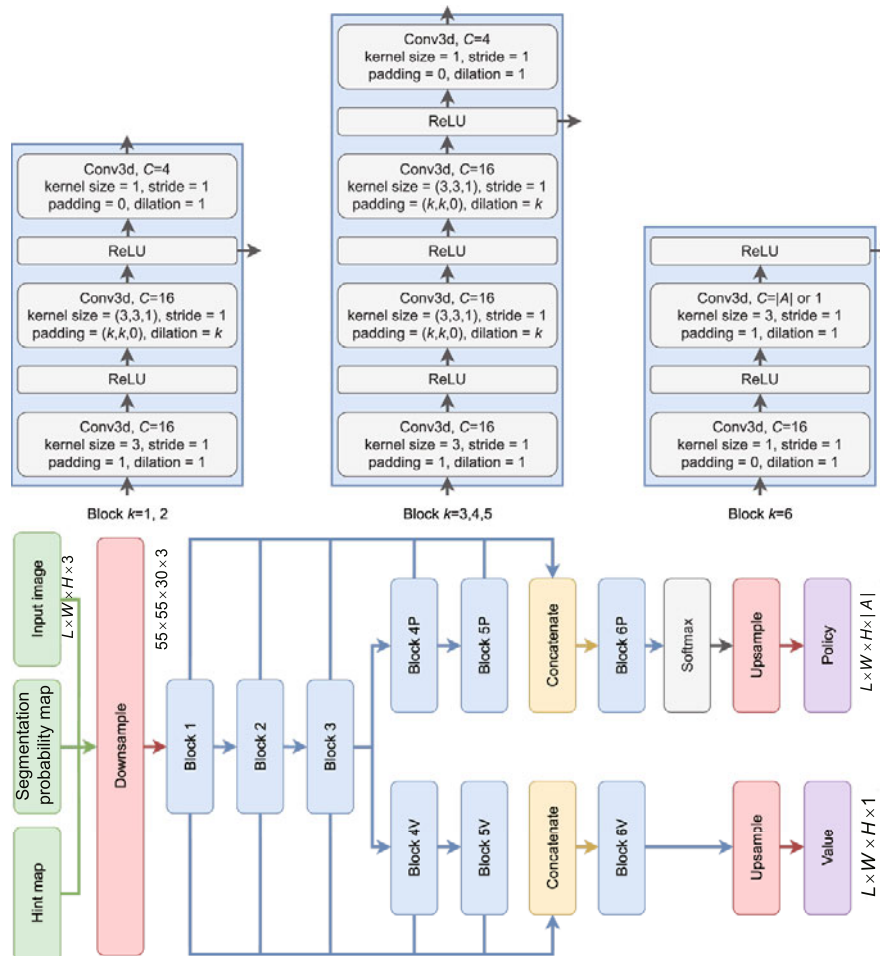


Fig. 4 Architecture of the segmentation network. The confidence network is the same as the value branch of the segmentation network, but the parameters are not shared

The policy head's goal is to maximize the expected return by selecting proper actions in state $s^{(t)}$. We use θ_p to denote the parameters of the policy head, and the gradient for θ_p is denoted as

$$d_{\theta_p} = -\nabla_{\theta_p} \pi(a^{(t)} | s^{(t)}) A(a^{(t)}, s^{(t)}). \quad (5)$$

Usually, the policy head is updated more slowly.

3.3 Action confidence learning

As we mentioned in Section 1, there will be some situations where the segmentation model misunderstands or ignores the hint information. To some extent, these samples (or regions) with the phenomenon of interactive misunderstanding are hard samples (or regions). Although these samples may account for a small portion of the dataset, they are critical for improving generalization and robustness. The most important thing is finding a professional

easy-or-hard representer (Nie et al., 2019) to identify them during the interaction. Focal loss (Lin TY et al., 2017) evaluates the easy-or-hard samples through a predicted probability. Nie et al. (2019) applied adversarial learning to train the easy-or-hard representer. Each representer has its advantages, but all evaluate easy-or-hard samples based on the final segmentation result. As such, these methods cannot be directly applied to interactive segmentation. For example, if the model predicts the category probability of a voxel to be 0.8 ($p_i^{(t)} = 0.8$), and then takes an action $a_i^{(t)} = -0.1$, the next prediction will be 0.7 after interaction ($p_i^{(t+1)} = 0.7$). If $y_i = 1$, the results are both correctly predicted because $p_i^{(t)}$ and $p_i^{(t+1)}$ are both larger than 0.5, whereas for interactive segmentation, the probability is changing in the wrong direction. This change is what we call the interactive misunderstanding phenomenon, and the formal

definition is shown in the following:

Definition 1 (Interactive misunderstanding) For a binary classification problem, the sign of the foreground label $y = 1$ is denoted as positive, and the sign of the background label $y = 0$ is denoted as negative. In an interactive medical image segmentation task (i.e., a voxel-wise binary classification problem), for any voxel i , if the sign of the change of the segmentation probability outputted by the algorithm for two consecutive interaction steps, $\text{sign}(\Delta(p^{(i)}))$, is not equal to $\text{sign}(y_i)$, then this phenomenon is defined as interactive misunderstanding.

The confidence network learns the confidence of the given actions to avoid misunderstanding of hint information and to take accurate actions. We argue that confidential information can be used to regularize action choices and suggest more efficient interaction. The confidence network structure also uses P-Net as the backbone. The confidence network contains six 3D convolutional blocks. Each of the blocks has two convolution layers, and the size of the convolution kernel is fixed as 3×3 in all these convolution layers. The detailed architecture is also shown in Fig. 4.

The confidence network is trained using the previous state and action as input and a confidence map as output. The confidence network is optimized by minimizing the summation of binary cross-entropy loss over actions (shown in Eq. (6)) at each time step t . Here we use C to denote the confidence network, w_C the confidence network parameters, and L_{BCE} the binary cross-entropy loss:

$$L_C(s^{(t)}, a^{(t)}; w_C) = L_{\text{BCE}}(C(s^{(t)}, a^{(t)}), g^{(t)}) + L_{\text{BCE}}(C(s^{(t)}, -a^{(t)}), 1 - g^{(t)}), \quad (6)$$

where

$$g^{(t)} = \begin{cases} 0, & \text{if } a^{(t)} \oplus y^{(t)} = 1, \\ 1, & \text{otherwise,} \end{cases} \quad (7)$$

and $g^{(t)}$ indicates whether the direction of the action is consistent with the label. For $a \oplus b$, the statement is true only if either $a > 0$ or $b > 0$.

One potential issue when training the confidence network is the imbalance of samples. Early in RL training, the error rate of actions outputted by the segmentation network is high, whereas most actions are correct when the network gradually converges. Inspired by discriminator training in GANs, this study introduces symmetric samples into Eq. (6)

to speed up training. An obvious advantage is that it improves sample utilization efficiency because the confidence network can know what a “bad sample” is and obtain a corresponding “good sample.”

3.4 Self-adaptive reward

Essentially, no matter whether focal loss or adversarial learning is applied to train the segmentation model, the objective is to weigh these hard samples to prevent the model from being dominated by easy samples. However, the interactive segmentation task differs from those fully automatic segmentation tasks because the interactive segmentation model needs to cooperate with the user and understand the user’s hint information. The action to refine the segmentation result shows how the segmentation model understands hint information. Therefore, it is necessary to ensure that the hint information is correctly understood and that the correct action is taken.

Specifically, the previously described action-confidence learning can provide the segmentation model with a confidence map to alleviate interactive misunderstanding. Using this confidence map, easy-or-hard samples can be better recognized because the confidence values for these “hard regions” are lower than those for other regions. We formulate this action-aware learning as the self-adaptive reward function, $r^{(t)}$, which is shown in Eq. (8), to adapt this mechanism to MARL training:

$$r^{(t)} = \sum_{i=1}^N \alpha (2 - c_i)^\beta \text{gain}_i^{(t)}, \quad (8)$$

where c_i is the value on the confidence map. The setting of hyperparameters α and β is described in Section 4.1. In addition, $\text{gain}_i^{(t)}$ denotes the relative gain of cross-entropy:

$$\begin{cases} \text{gain}_i^{(t)} = \mathcal{X}_i^{(t-1)} - \mathcal{X}_i^{(t)}, \\ \mathcal{X}_i^{(t)} = -y_i \ln(p_i^{(t)}) - (1 - y_i) \ln(1 - p_i^{(t)}), \end{cases} \quad (9)$$

where $\mathcal{X}_i^{(t)}$ denotes the cross-entropy between the current segmentation probability and ground truth. If an agent obtains a positive reward, its current action is good, and the refined segmentation result is close to the ground truth. With the self-adaptive reward function in Eq. (8), the confidence value c_i of these wrong actions is lower, and they will be punished more when training our segmentation model.

3.5 Interaction guide

Another challenge for interactive image segmentation is that users usually need to decide where to interact in many slices, which is very time-consuming, especially for 3D images. Our framework provides users with an interaction guide mechanism to save user interaction time. After refinement, our framework will suggest some possible areas for users to segment during the interaction. Specifically, our framework will filter out those areas with low action confidence and provide them to users (Fig. 5). First, the original 3D image will be segmented with super voxels; each super voxel can be regarded as a group of voxels that share common characteristics. We use a simple linear iterative clustering (SLIC) (Achanta et al., 2012) technique with spacing = [2, 2, 2], compactness = 0.1 to generate super voxels, and the number of super voxels equals 100 at the beginning and gradually declines during the refinement iterations for training and testing. Second, the proposed algorithm will compute the mean action confidence in each super voxel and rank them in descending order. Finally, the top five super voxels will be marked and recommended to users. Users need to select the best interaction positions from these super voxels.

3.6 Simulated label generation

In medical imaging, there are much more unlabeled data than labeled data due to difficulties in labeling medical images. To address this lack of annotations, the proposed algorithm leverages the action confidence not only to improve the utilization efficiency of hint information but also to generate a simulated label (Fig. 6) for unlabeled data. We define the simulated label as follows:

$$\hat{y}^{(t)} = \begin{cases} 1, & \text{if } a^{(t)} \oplus c^{(t)} = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where $\hat{y}^{(t)}$ is the simulated voxel-level label generated from the confidence map (Fig. 6). Based on this mechanism, we consider using these unlabeled data to assist in training the segmentation network. Specifically, this study divides the dataset into two parts during the training stage: the first part contains voxel-level annotation information and the second part contains only hint information during training. The labeled data are trained as above, while the unlabeled data are trained with the simulated label.

During the training stage, the backward gradients will be masked:

$$\begin{cases} d\theta_v = M\nabla_{\theta_v} A^2(s^{(t)}, a^{(t)}), \\ d\theta_p = M\nabla_{\theta_p} \pi(a^{(t)}|s^{(t)})A(a^{(t)}, s^{(t)}), \end{cases} \quad (11)$$

where $M = I(\max(c^{(t)}, 1 - c^{(t)}) > \delta)$, and $A(s^{(t)}, a^{(t)})$ is the advantage (defined in Mnih et al. (2016)) at time step t of taking $a^{(t)}$ in the condition of state $s^{(t)}$, indicating the actual accumulated reward without being affected by the state and reducing the gradient variance. A mask, M , is used to constrain the training of unlabeled data. The backward gradients of unlabeled data occur only when the action confidence exceeds threshold δ (which is gradually increased during the training process). Unlike traditional pseudo-label training, the supervised signal does not come from the segmentation network but from the confidence network. These filtered data with hint information are more valuable and provide more accurate supervised signals. Generally, the training processes with labeled data and simulated labeled data are carried out simultaneously. When using the simulated label generation mechanism, the pseudocode is as shown in Algorithm 1.

Algorithm 1 Interactive medical image segmentation with self-adaptive confidence calibration

- 1: Initialize the segmentation network S with θ ;
 - 2: Initialize the confidence network C with w ;
 - 3: **for** every sample in labeled datasets **do**
 - 4: Set the segmentation probability of each voxel to 0.5; $s^{(0)} \leftarrow (x, p^{(0)}, h^{(0)})$;
 - 5: **for** every interaction time step t **do**
 - 6: Take action $a^{(t)} \leftarrow S(s^{(t)})$;
 - 7: Obtain the reward $r^{(t)}$ and next state $s^{(t+1)}$;
 - 8: Compute the gradient of S via Eqs. (4) and (5);
 - 9: Compute the gradient of C via Eq. (6);
 - 10: **end for**
 - 11: Obtain a sample (x', y') from the unlabeled dataset;
 - 12: Initialize the state: $s'^{(0)} \leftarrow (x', p'^{(0)}, h'^{(0)})$;
 - 13: **for** every interaction time step t **do**
 - 14: Take action $a'^{(t)} \leftarrow S(s'^{(t)})$;
 - 15: Generate the simulated label $\hat{y}^{(t)}$ by Eq. (10);
 - 16: Observe the reward r'^t and next state $s'^{(t+1)}$;
 - 17: Compute the gradient of S via Eq. (11);
 - 18: **end for**
 - 19: **end for**
-

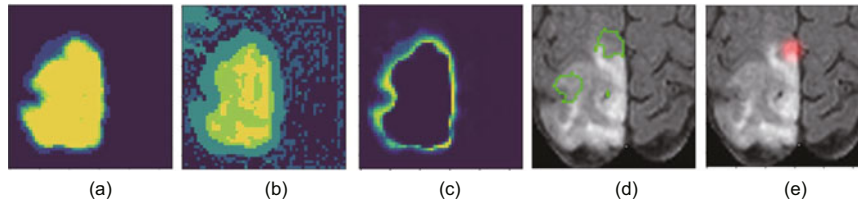


Fig. 5 Interaction guide mechanism: (a) probability map; (b) action map; (c) confidence map; (d) advice region; (e) interaction point. The areas surrounded by the green lines are advice regions provided to users, and the red point in (e) is the real hint information selected from advice regions. The brighter the color (closer to yellow), the larger the positive value; conversely, the darker the color (closer to black), the smaller the negative value. References to color refer to the online version of this figure

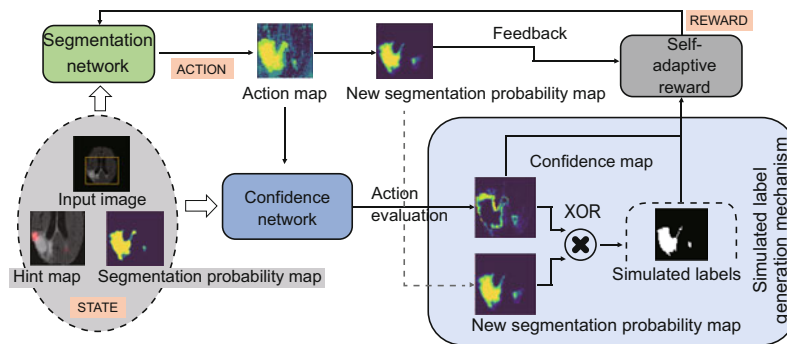


Fig. 6 Illustration of simulated label generation. The simulated label generation mechanism uses the confidence map and the action map to generate the simulated label. The confidence map is used to calibrate the action, and the direction of the calibrated action is the simulated label of each voxel

4 Experiments and results

4.1 Datasets and implementation details

To comprehensively evaluate our proposed method, we apply our algorithm to four 3D medical image datasets. All datasets are divided into two parts: $D_{\text{train}}/D_{\text{test}}$. The details of these datasets are as follows: (1) BraTS2015: Brain Tumor Segmentation Challenge 2015 (Menze et al., 2015) contains 274 (234/40) multi-parametric MRI images (Flair, T1, T1C, T2) from brain tumor patients. In our task, we use only the Flair image and segment the whole brain tumor. (2) BraTS2020: Brain Tumor Segmentation Challenge 2020 (Menze et al., 2015) contains 285 (235/50) multi-parametric MRI images (Flair, T1, T1C, T2) from brain tumor patients. In our task, we use only the Flair image and segment the whole brain tumor. (3) MM-WHS: Multi-Modality Whole Heart Segmentation (Zhuang and Shen, 2016) contains 24 (20/4) multi-modality whole heart images covering the whole heart substructures. In our task, we choose to segment the left atrium blood cavity. (4) Medical Segmentation Decathlon: This is a generaliz-

able 3D semantic segmentation dataset containing different organ segmentation tasks (Simpson et al., 2019). We use the spleen and liver datasets, which provide 61 (41/20) and 106 (96/10) CT images, respectively.

We implement our method with PyTorch (Paszke et al., 2019). The demo video of MECCA is available at <https://bit.ly/mecca-demo-video>. The segmentation and confidence networks are both initialized using the Xavier (Glorot and Bengio, 2010) method, and the learning rate is initialized to $1e-4$. Other parameters are set as follows: $T = 5$, $\gamma = 0.95$, $\alpha = 0.8$, $\beta = 1$. The mask M ranges from 0.85 to 0.99 and increases by 0.000 25 at every epoch. Adam (Kingma and Ba, 2015) is adopted as the optimizer. The original image is cropped by the bounding box based on the ground truth with a random extension in the range of 1 – 11 voxels. Each image is then resized and normalized to $55 \times 55 \times 30$. The data are augmented by random flips and rotations. The proposed algorithm's training time with one Nvidia 2080ti GPU varies from 5 to 13 h for different datasets.

4.2 Experimental settings

During the process of interaction, we adopt the edge points as the hint information for two reasons. On one hand, click operation saves more time than scribbles or other interactive methods. On the other hand, edge points can provide more information about object edges because the bottleneck of medical image segmentation is usually the inaccurate segmentation of object edges. Specifically, we provide each method with 45 points during the whole interaction process. At every step, users will click some edges that are not correctly predicted. We generate a 3D Gaussian (with a kernel size of eight voxels) centered on each of the edge points as the hint map to let networks receive hint information. Then the hint map will be inputted to the segmentation network as part of the state. We use the Dice score and the average symmetric surface distance (ASSD) to evaluate the segmentation results. According to these evaluation metrics, doctors can judge the patient's condition:

$$\text{Dice}(S_p, S_g) = (2|S_p \cap S_g|) / (|S_p| + |S_g|),$$

where S_p and S_g denote the prediction of an algorithm and the ground truth, respectively.

$$\text{ASSD} = (\sum_{i \in S_a} d(i, S_b) + \sum_{i \in S_b} d(i, S_a)) / (|S_a| + |S_b|),$$

where S_a and S_b represent the sets of surface points of the segmentation results predicted by the algorithm and the ground truth, respectively. $d(i, S_b)$ is the shortest Euclidean distance between i and S_b . The Dice and ASSD in all tables are the average values of five algorithm test results.

4.3 Comparisons with baselines

We compare MECCA with four state-of-the-art interactive segmentation methods: DeepIGeoS (Wang et al., 2019), InterCNN (Bredell et al., 2018), Iter-MRL (Liao et al., 2020), and BS-IRIS (Ma et al., 2021). InterCNN is the multi-step version of DeepIGeoS. We also introduce a state-of-the-art medical image segmentation method, U-Net (Ronneberger et al., 2015), as a baseline. Table 1 shows the quantitative comparison of the seven segmentation methods on different datasets. For a fair comparison, except U-Net, all CNN-based methods adopt the same network structure (P-Net), which was proposed in Wang et al. (2019). In particular,

U-Net and P-Net are methods without hint information. We can see that MECCA performs better than or is on par with the state-of-the-art methods on three datasets. We visualize the results in Fig. 7, which shows that our method achieves more accurate results in edge segmentation.

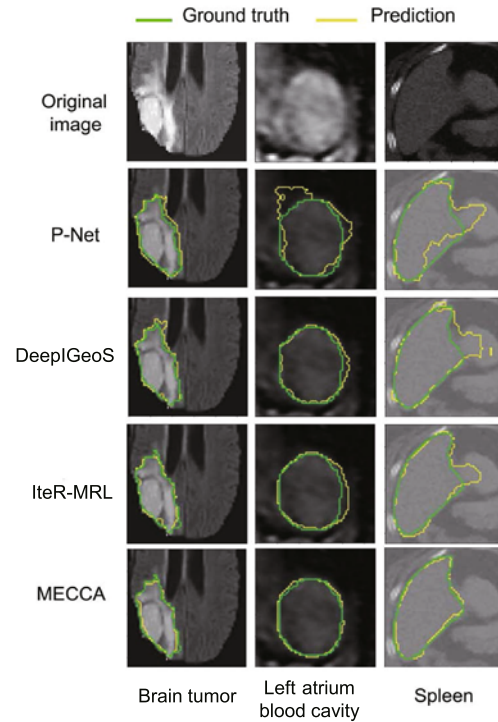


Fig. 7 Visualization of segmentation results by different methods. The green lines represent the boundaries of the ground truth, and the yellow lines represent the predicted boundaries. References to color refer to the online version of this figure

To demonstrate that MECCA can take advantage of hint information more efficiently, we compare the relative improvement of different methods at each interaction step. Due to the different interaction processes of different methods, we set up a unified interaction process: all these methods have five interaction steps and receive 25 identical points at the first step to generate the initial segmentation result. After that, users will click five points at each step in the following four interaction steps (a total of $25 + 5 \times 4 = 45$ points), and these methods will iteratively refine previous segmentation results. The DeepIGeoS method does not model the interaction sequence, and simply combines current and previous hint information to refine the previous segmentation. The results are shown in Table 2 and Fig. 8.

Table 1 Quantitative comparison of different methods

| Method | Dice (%) | | | | |
|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|
| | BraTS2020 | BraTS2015 | MM-WHS | Spleen | Liver |
| P-Net | 83.67±8.35 | 84.00±12.01 | 81.40±1.48 | 88.08±2.25 | 35.89±2.61 |
| U-Net | 84.72±10.42 | 84.66±11.25 | 80.96±1.65 | 87.95±2.87 | 56.00±1.93 |
| DeepIGeoS | 88.54±.97 | 88.32±5.34 | 88.48±0.71 | 91.97±1.51 | 48.57±2.52 |
| InterCNN | 88.39±6.01 | 88.26±7.07 | 87.85±1.15 | 93.52±0.94 | 59.92±2.20 |
| IteR-MRL | 89.22±4.65 | 88.94±4.81 | 89.55±0.87 | 91.50±1.34 | 62.29±1.93 |
| BS-IRIS | 90.47±5.23 | 89.74±3.86 | 89.12±0.98 | 92.35±1.13 | 67.25±2.01 |
| MECCA | 91.02±5.86 | 90.29±5.07 | 90.39±5.89 | 94.96±1.44 | 71.46±1.41 |

| Method | ASSD (pixel) | | | | |
|-----------|------------------|------------------|------------------|------------------|------------------|
| | BraTS2020 | BraTS2015 | MM-WHS | Spleen | Liver |
| P-Net | 5.78±4.01 | 5.10±3.72 | 3.28±0.45 | 4.25±2.07 | 6.05±4.18 |
| U-Net | 4.09±3.89 | 6.17±4.69 | 3.72±0.39 | 5.12±1.09 | 3.93±3.94 |
| DeepIGeoS | 2.11±1.30 | 2.28±1.24 | 1.53±0.18 | 0.93±0.46 | 1.80±0.69 |
| InterCNN | 2.01±1.09 | 1.81±2.09 | 0.80±0.15 | 0.54±0.83 | 1.04±0.56 |
| IteR-MRL | 2.07±0.91 | 1.41±0.22 | 0.90±0.11 | 0.67±0.21 | 0.87±0.59 |
| BS-IRIS | 1.82±0.33 | 1.61±0.42 | 1.19±0.16 | 0.54±0.19 | 0.76±0.24 |
| MECCA | 1.15±0.20 | 1.50±0.33 | 0.80±0.01 | 0.30±0.16 | 0.41±0.20 |

The best results are in bold

Table 2 Dice scores at each interaction step by different methods

| Method | Dice (%) | | | | |
|-----------|--------------|--------------|--------------|--------------|--------------|
| | 1 | 2 | 3 | 4 | 5 |
| DeepIGeoS | 87.36 | 87.53 | 87.67 | 87.99 | 88.32 |
| | | (+0.17) | (+0.14) | (+0.32) | (+0.33) |
| InterCNN | 87.21 | 88.59 | 88.54 | 88.39 | 88.26 |
| | | (+1.38) | (−0.05) | (−0.15) | (−0.13) |
| IteR-MRL | 84.56 | 85.35 | 88.15 | 88.11 | 88.94 |
| | | (+0.79) | (+2.80) | (−0.04) | (+0.83) |
| MECCA | 86.49 | 88.53 | 89.56 | 89.12 | 90.29 |
| | | (+2.04) | (+1.03) | (−0.44) | (+1.17) |

The value in parentheses represents the improvement relative to the previous step. The best results are in bold

The results show that MECCA performs better under the same amount of hint information and improves the Dice score more after the five steps. Compared with CNN-based methods, the main advantage of RL-based methods is that they can always maintain notable improvement. There are two reasons for this result. The first one is that RL-based methods model the whole interaction process to avoid interaction conflict. The second reason is the relative entropy-based reward which encourages the model to keep refining results. However, we should realize that the RL-based methods still cannot guarantee the high confidence of the corrective actions. As we can see in Fig. 8, the performance of IteR-MRL is not as good as those of other methods at the beginning, which is caused by numerous incorrect

actions. However, after learning the confidence of actions and applying the self-adaptive reward to update the segmentation model, our proposed MECCA can perform well at each step and continue significantly refining the result.

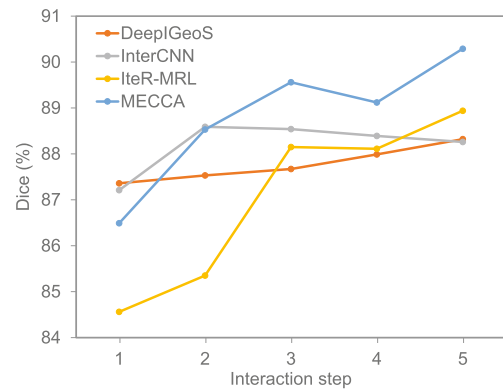


Fig. 8 The performance improvement of different interactive medical segmentation methods at different interaction steps. All these testing results were obtained on the BraTS2015 dataset

4.4 Comparison of different rewards

One main contribution of our work is that the self-adaptive reward mechanism is further proposed based on the confidence map. This mechanism makes different actions have different feedback levels, so the segmentation network can identify wrong actions as

much as possible. To measure the impact of different means of reward weighting on the segmentation result, we compare MECCA with the original IteR-MRL without a weighting reward, Focal-Reward IteR-MRL (Focal-IteR-MRL), and the one with a weighted reward in segmentation error regions (Err-IteR-MRL). Focal-IteR-MRL adopts the idea of focal loss (Lin TY et al., 2017):

$$\text{FocalReward}_i^{(t)} = \mathcal{X}_i^{(t-1)} - \mathcal{X}_i^{(t)}, \quad (12)$$

where

$$\mathcal{X}_i^{(t)} = \begin{cases} -(1-p_i)y_i \ln p_i^{(t)}, & y_i = 1, \\ -p_i(1-y_i) \ln(1-p_i^{(t)}), & y_i = 0. \end{cases} \quad (13)$$

The reward function of Err-IteR-MRL is a linear scaling of the IteR-MRL reward:

$$\text{ErrReward}_i^{(t)} = \begin{cases} (1+\lambda_i)\text{gain}_i^{(t)}, & \text{misclassified,} \\ \text{gain}_i^{(t)}, & \text{otherwise,} \end{cases} \quad (14)$$

where λ_i is a positive real number hyperparameter. The performance of different weighting rewards is shown in Fig. 9. It can be seen that the action-based reward weighting method is more suitable for interactive segmentation. Also, this study visualizes different masks that weigh the primary reward in Fig. 10. As we can see, the focal mask pays more attention to regions with low prediction probabilities, and the error region mask contains all regions segmented incorrectly in the previous result. It is not enough for the focal mask to obtain more structured information by considering only the probability of prediction. As such, the performance of Focal-IteR-MRL is erratic, even the poorest on the left atrium dataset. The performance of Err-IteR-MRL is more stable, but the improvement is not notable. This is because the weighting process of Err-IteR-MRL is based on the segmentation result while the refinement process of interactive image segmentation is based on actions.

4.5 Weakly supervised segmentation

As mentioned in Section 3.6, MECCA can reduce the dependence on voxel-level annotations of images using the simulated label generated from the action confidence map. This study validates MECCA on the BraTS2015 dataset by randomly selecting different proportions of samples as fully labeled data and using the rest of the training images

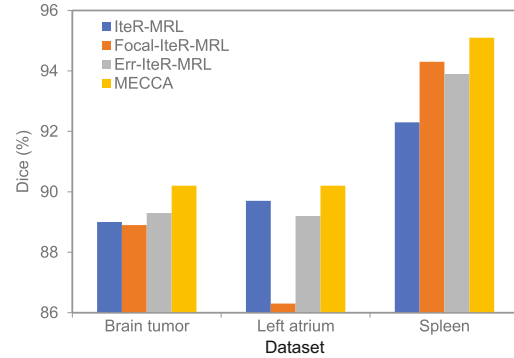


Fig. 9 Average Dice scores of methods with different weighting rewards. The reward function of IteR-MRL is not weighted; the reward function of Focal-IteR-MRL is weighted via Eq. (12), the reward function of Err-IteR-MRL is weighted in segmentation error regions, and the reward function of MECCA is weighted through action confidence

as unlabeled data, which provide only hint information during the interaction. This study compares MECCA against P-Net (without hint), DeepIGeoS, and IteR-MRL (Liao et al., 2020). However, only MECCA uses unlabeled data, and the three baselines use only a fixed proportion of labeled data. This does not fully demonstrate the performance of MECCA on semi-supervised problems. For this reason, we introduce a state-of-the-art semi-supervised method UA-MT (Yu et al., 2019). Table 3 shows the results of different methods. MECCA achieves a Dice score of 87.14% with only 12.5% labeled data, and it performs better than the other methods with 25% labeled data.

Because UA-MT is not an interactive segmentation algorithm, its absolute performance is worse than those of interactive segmentation baselines but better than that of the non-interactive method P-Net. In addition, it can be seen from the last column of Table 3 that UA-MT has the most negligible performance loss with different proportions of labeled data. After MECCA introduces the simulated label generation mechanism, the performance loss is similar to that of UA-MT. Interestingly, DeepIGeoS can maintain low performance loss without using a semi-supervised learning mechanism. Considering the performance loss of IteR-MRL, we can conclude that in the semi-supervised interactive segmentation task, the interactive misunderstanding phenomenon will exacerbate the performance loss caused by missing data. DeepIGeoS and MECCA alleviate the interactive misunderstanding phenomenon through

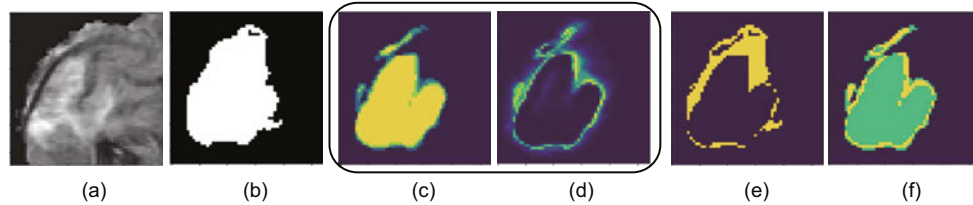


Fig. 10 Visualization of the original image (a), ground truth (b), action map (c), self-adaptive mask (d), error region mask (e), and focal mask (f). All masks are obtained at the first interaction step. The brighter the color (closer to yellow), the larger the positive value; conversely, the darker the color (closer to black), the smaller the negative value. References to color refer to the online version of this figure

Table 3 Quantitative comparison between MECCA and other methods on the BraTS2015 dataset with different portions of labeled data

| Method | Dice (%) | | | | | Improvement* |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
| | 12.5% | 25% | 33% | 50% | 100% | |
| P-Net | 75.86 | 80.83 | 80.88 | 83.02 | 84.00 | 10.73% |
| DeepIGeoS | 85.50 | 85.90 | 87.70 | 87.60 | 88.32 | 3.30% |
| IteR-MRL | 84.36 | 86.54 | 87.38 | 88.66 | 88.94 | 5.43% |
| UA-MT | 83.08 | 84.47 | 84.62 | 84.39 | 84.66 | 1.90% |
| MECCA | 87.14 | 88.23 | 88.31 | 89.17 | 90.29 | 3.61% |

The best results are in bold. * Dice at 100% labeled data compared with that at 12.5% labeled data

hard constraints and self-adaptive confidence calibration, respectively, so both of them can achieve better results in semi-supervised settings. However, note that DeepIGeoS does not consider multi-step interactions or the relationship between consecutive interactions. It cannot fully use long-term interaction information, so there is a large gap with MECCA in absolute performance.

We also evaluate MECCA with different numbers of interactions and percentages of labeled data. Based on the experimental results (Table 4), we conclude the following: (1) if there are enough labeled data, MECCA can achieve good results after a few interactions; (2) the number of interactions required by MECCA to achieve the same performance is roughly inversely proportional to the portion of labeled data; (3) although MECCA requires more interactions (about 2–3 times) when only part of the labeled data is available, it can approach the performance of the algorithm trained with all data being labeled in the end.

4.6 Ablation studies

Table 5 shows the impact of different mechanisms on MECCA. It is clear that the simulated label generation mechanism and self-adaptive reward mechanism significantly improve the algorithm's

Table 4 Dice scores with different numbers of interactions and percentages of labeled data

| Percentage of labeled data (%) | Dice (%) | | | | |
|--------------------------------|----------|-------|-------|-------|-------|
| | 3 | 4 | 5 | 6 | 7 |
| 25 | 84.03 | 86.18 | 88.23 | 87.32 | 88.24 |
| 33 | 85.32 | 87.60 | 88.31 | 87.99 | 88.53 |
| 50 | 85.39 | 87.07 | 89.17 | 88.93 | 89.13 |
| 100 | 89.93 | 90.40 | 90.29 | 89.85 | 90.79 |

performance. The simulated label generation mechanism can achieve a 1.70% and 2.26% gain over the initial algorithm without any mechanism for the BraTS2015 dataset and the Liver dataset in Medical Segmentation Decathlon, respectively. The main reason for this significant improvement is the utilization of these unlabeled data.

Table 5 Ablation study of our proposed algorithm on BraTS2015 and Liver datasets with 25% labeled data

| Self-adaptive reward | Interaction guide | Simulated label generation | Dice (%) | |
|----------------------|-------------------|----------------------------|--------------|--------------|
| | | | BraTS2015 | Liver |
| | | | 86.54 | 66.43 |
| ✓ | | | 87.21 | 67.45 |
| | ✓ | | 86.56 | 66.56 |
| | | ✓ | 88.01 | 67.93 |
| ✓ | ✓ | | 87.24 | 67.59 |
| ✓ | ✓ | ✓ | 88.23 | 69.99 |

The best results are in bold

Table 6 shows the comparison of interactive segmentation methods in computational and interaction time. The results show that MECCA takes only about half the time of other methods when performing an interaction. The main contribution to the reduction of interaction time is the interaction guide mechanism mentioned. Furthermore, to explore the efficiency of MECCA in authentic tasks, we design an interactive software platform. We ask oncologists to use it for natural interactive segmentation, and the results are shown in Table 7. It can be seen from the results that the MECCA algorithm also takes only about half the time of the baselines due to interaction guidance. In addition, as the segmentation results become more and more accurate, the interaction time required gradually decreases.

Table 6 Comparison of computational and interaction time among different interactive segmentation methods

| Method | Total inference time per iteration (ms) | Interaction time per iteration (ms) |
|-----------|---|-------------------------------------|
| DeepIGeoS | 510 | 226 |
| InterCNN | 532 | 237 |
| IteR-MRL | 869 | 235 |
| BS-IRIS | 990 | 250 |
| MECCA | 631 | 123 |

The best results are in bold

Table 7 Quantitative comparison of interaction time among different interactive segmentation methods with different numbers of interactions on a realistic segmentation platform operated by experts

| Method | Interaction time (s) | | | | |
|-----------|----------------------|-----------|-----------|----------|----------|
| | 1 | 2 | 3 | 4 | 5 |
| DeepIGeoS | 70 | 15 | 15 | 8 | 8 |
| InterCNN | 76 | 19 | 18 | 8 | 7 |
| IteR-MRL | 75 | 17 | 17 | 8 | 8 |
| BS-IRIS | 80 | 20 | 20 | 11 | 10 |
| MECCA | 41 | 12 | 11 | 5 | 5 |

Initially there are 25 points, and during each next interaction five new points are added. The best results are in bold

5 Conclusions

We present a novel action-based confidence learning method for interactive 3D image segmentation. Specifically, we propose a method for learning the confidence of actions that continuously refine the segmentation result during the interaction

process, so that hint information can be used more effectively. Based on this process, a self-adaptive reward is proposed for the segmentation module, which can prevent the misunderstanding phenomenon during the interaction and help reduce the time cost of interaction by providing users with the advice regions with which they should interact next. In addition, the confidence map can replace the ground truth to generate feedback for the unlabeled samples for the segmentation module. These samples, without voxel-level annotations, can also be used to model training. By integrating these components, MECCA is demonstrated to improve medical segmentation efficiently using fewer annotated samples.

Although our method can greatly enhance the utilization of short- and long-term interaction information with significantly fewer labeled samples, there are also certain limitations that we need to address. One of the primary challenges in medical image segmentation is the high dimension and resolution of medical images, leading to a requirement for numerous interactive clicks. Specifically, Wang et al. (2018) provided the actual use time collected from real physicians. Thus, there is a need for more efficient interactive methods, and Aljabri et al. (2022) introduced the currently available annotation tools for medical imaging. In future work, we may incorporate more efficient annotation methods to minimize the cost of physician interactions.

Further, RL from human feedback (RLHF), which is used in ChatGPT (OpenAI, 2022), is increasingly popular. This approach collects comparison data and trains a reward model, where the data are human rankings of the model's outputs from best to worst. RLHF is another application that benefits from human feedback, and in future work, we may consider another interactive method in which radiographers rank the different outputs as reward signals. Another limitation in the field of medical image segmentation is the presence of multiple centers, where data are collected from different devices, resulting in varying distributions. With such non-independent and identically distributed (non-IID) data, the generalization ability of deep learning based models is limited (Li et al., 2021; Ye et al., 2022). To overcome this, our proposed MECCA method may require additional design; e.g., Aljabri et al. (2022) introduced a novel robust weakly supervised learning paradigm.

Finally, as technology continues to advance,

changes in patient distribution and the introduction of new medical equipment may necessitate the development of online learning methods to ensure that the algorithm remains robust. Popular methods such as iCaRL (Rebuffi et al., 2017), GDumb (Prabhu et al., 2020), and iCaRL (Rebuffi et al., 2017) maintain a buffer to store the most important samples for the model. Specifically, in the RL area, we can borrow the ideas and theoretical justifications from lifelong RL, in which setting agents must solve a series of related tasks drawn from a task distribution rather than a single, isolated task (Abel et al., 2018). Abel et al. (2018) explored which knowledge should be transferred in lifelong RL through two simple families of results. Xie et al. (2020) leveraged latent variable models to learn a representation of the environment from experience and performed off-policy RL with this representation.

Contributors

Chuyun SHEN, Wenhao LI, and Qishen XU designed the research and conducted the experiments. Bin HU, Fengping ZHU, and Yuxin LI ensured the validity of the experiments. Bo JIN, Haibin CAI, and Xiangfeng WANG offered support across various experimental aspects. Chuyun SHEN drafted the paper. All the authors revised and finalized the paper.

Compliance with ethics guidelines

Chuyun SHEN, Wenhao LI, Qisen XU, Bin HU, Bo JIN, Haibin CAI, Fengping ZHU, Yuxin LI, and Xiangfeng WANG declare that they have no conflict of interest.

Data availability

The demo video is available at <https://bit.ly/meccademo-video>. The other data that support the findings of this study are available from the corresponding author upon reasonable request.

References

- Abel D, Jinnai Y, Guo SY, et al., 2018. Policy and value transfer in lifelong reinforcement learning. Proc 35th Int Conf on Machine Learning, p.20-29.
- Achanta R, Shaji A, Smith K, et al., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Patt Anal Mach Intell*, 34(11):2274-2282. <https://doi.org/10.1109/TPAMI.2012.120>
- Acuna D, Ling H, Kar A, et al., 2018. Efficient interactive annotation of segmentation datasets with Polygon-RNN++. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.859-868. <https://doi.org/10.1109/CVPR.2018.00096>
- Aljabri M, AlAmir M, AlGhamdi M, et al., 2022. Towards a better understanding of annotation tools for medical imaging: a survey. *Multim Tools Appl*, 81(18):25877-25911. <https://doi.org/10.1007/s11042-022-12100-1>
- Bredell G, Tanner C, Konukoglu E, 2018. Iterative interaction training for segmentation editing networks. Proc 9th Int Workshop on Machine Learning in Medical Imaging, p.363-370. https://doi.org/10.1007/978-3-030-00919-9_42
- Castrejón L, Kundu K, Urtasun R, et al., 2017. Annotating object instances with a polygon-RNN. IEEE Conf on Computer Vision and Pattern Recognition, p.4485-4493. <https://doi.org/10.1109/CVPR.2017.477>
- DeVries T, Taylor GW, 2018a. Learning confidence for out-of-distribution detection in neural networks. <https://doi.org/10.48550/arXiv.1802.04865>
- DeVries T, Taylor GW, 2018b. Leveraging uncertainty estimates for predicting segmentation quality. <https://doi.org/10.48550/arXiv.1807.00502>
- Feng RW, Zheng XS, Gao TX, et al., 2021. Interactive few-shot learning: limited supervision, better medical image segmentation. *IEEE Trans Med Imag*, 40(10):2575-2588. <https://doi.org/10.1109/TMI.2021.3060551>
- Furuta R, Inoue N, Yamasaki T, 2020. PixelRL: fully convolutional network with reinforcement learning for image processing. *IEEE Trans Multim*, 22(7):1704-1719. <https://doi.org/10.1109/TMM.2019.2960636>
- Glorot X, Bengio Y, 2010. Understanding the difficulty of training deep feedforward neural networks. Proc 13th Int Conf on Artificial Intelligence and Statistics, p.249-256.
- Hung W, Tsai Y, Liou Y, et al., 2018. Adversarial learning for semi-supervised semantic segmentation. Proc British Machine Vision Conf, p.65.
- Jungo A, Reyes M, 2019. Assessing reliability and challenges of uncertainty estimations for medical image segmentation. Proc 22nd Int Conf on Medical Image Computing and Computer-Assisted Intervention, p.48-56. https://doi.org/10.1007/978-3-030-32245-8_6
- Kendall A, Gal Y, 2017. What uncertainties do we need in bayesian deep learning for computer vision? Proc 3rd Int Conf on Neural Information Processing System, p.5580-5590.
- Kingma DP, Ba J, 2015. Adam: a method for stochastic optimization. Proc 3rd Int Conf on Learning Representations. <https://doi.org/10.48550/arXiv.1412.6980>
- Lee KM, Song G, 2018. SeedNet: automatic seed generation with deep reinforcement learning for robust interactive segmentation. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.1760-1768. <https://doi.org/10.1109/CVPR.2018.00189>
- Li L, Zimmer VA, Schnabel JA, et al., 2021. AtrialGeneral: domain generalization for left atrial segmentation of multi-center LGE MRIs. Proc 24th Int Conf on Medical Image Computing and Computer-Assisted Intervention, p.557-566. https://doi.org/10.1007/978-3-030-87231-1_54
- Liao X, Li WH, Xu QS, et al., 2020. Iteratively-refined interactive 3D medical image segmentation with multi-agent reinforcement learning. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9394-9402. <https://doi.org/10.1109/CVPR42600.2020.00941>
- Lin D, Dai JF, Jia JY, et al., 2016. ScribbleSup: scribble-supervised convolutional networks for semantic segmentation. IEEE Conf on Computer Vision and Pattern Recognition, p.3159-3167. <https://doi.org/10.1109/CVPR.2016.344>

- Lin TY, Goyal P, Girshick R, et al., 2017. Focal loss for dense object detection. Proc IEEE Int Conf on Computer Vision, p.2999-3007. <https://doi.org/10.1109/ICCV.2017.324>
- Ma CF, Xu QS, Wang XF, et al., 2021. Boundary-aware supervoxel-level iteratively refined interactive 3D image segmentation with multi-agent reinforcement learning. *IEEE Trans Med Imag*, 40(10):2563-2574. <https://doi.org/10.1109/TMI.2020.3048477>
- Menze BH, Jakab A, Bauer S, et al., 2015. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans Med Imag*, 34(10):1993-2024. <https://doi.org/10.1109/TMI.2014.2377694>
- Mnih V, Badia AP, Mirza M, et al., 2016. Asynchronous methods for deep reinforcement learning. Proc 33rd Int Conf on Machine Learning, p.1928-1937.
- Moeskops P, Veta M, Lafarge MW, et al., 2017. Adversarial training and dilated convolutions for brain MRI segmentation. Proc 3rd Int Workshop on Deep Learning in Medical Image Analysis and 7th Int Workshop on Multimodal Learning for Clinical Decision Support, p.56-64. https://doi.org/10.1007/978-3-319-67558-9_7
- Nie D, Wang L, Xiang L, et al., 2019. Difficulty-aware attention network with confidence learning for medical image segmentation. Proc 33rd AAAI Conf on Artificial Intelligence, 31st Innovative Applications of Artificial Intelligence Conf, and 9th AAAI Symp on Educational Advances in Artificial Intelligence, p.1085-1092. <https://doi.org/10.1609/aaai.v33i01.33011085>
- OpenAI, 2022. ChatGPT: Optimizing Language Models for Dialogue. <https://openai.casa/blog/chatgpt/> [Accessed on July 10, 2022].
- Paszke A, Gross S, Massa F, et al., 2019. PyTorch: an imperative style, high-performance deep learning library. Proc 33rd Int Conf on Neural Information Processing Systems, p.8026-8037.
- Prabhu A, Torr PHS, Dokania PK, 2020. GDumb: a simple approach that questions our progress in continual learning. Proc 16th European Conf on Computer Vision, p.524-540. https://doi.org/10.1007/978-3-030-58536-5_31
- Rajchl M, Lee MCH, Oktay O, et al., 2017. DeepCut: object segmentation from bounding box annotations using convolutional neural networks. *IEEE Trans Med Imag*, 36(2):674-683. <https://doi.org/10.1109/TMI.2016.2621185>
- Rebuffi SA, Kolesnikov A, Sperl G, 2017. iCarL: incremental classifier and representation learning. IEEE Conf on Computer Vision and Pattern Recognition, p.5533-5542. <https://doi.org/10.1109/CVPR.2017.587>
- Robinson R, Oktay O, Bai WJ, et al., 2018. Real-time prediction of segmentation quality. Proc 21st Int Conf on Medical Image Computing and Computer-Assisted Intervention, p.578-585. https://doi.org/10.1007/978-3-030-00937-3_66
- Ronneberger O, Fischer P, Brox T, 2015. U-Net: convolutional networks for biomedical image segmentation. Proc 18th Int Conf on Medical Image Computing and Computer-Assisted Intervention, p.234-241. https://doi.org/10.1007/978-3-319-24574-4_28
- Shrivastava A, Gupta A, Girshick R, 2016. Training region-based object detectors with online hard example mining. IEEE Conf on Computer Vision and Pattern Recognition, p.761-769. <https://doi.org/10.1109/CVPR.2016.89>
- Simpson AL, Antonelli M, Bakas S, et al., 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. <https://doi.org/10.48550/arXiv.1902.09063>
- Wang GT, Li WQ, Zuluaga MA, et al., 2018. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE Trans Med Imag*, 37(7):1562-1573. <https://doi.org/10.1109/TMI.2018.2791721>
- Wang GT, Zuluaga MA, Li WQ, et al., 2019. DeepIGeoS: a deep interactive geodesic framework for medical image segmentation. *IEEE Trans Patt Anal Mach Intell*, 41(7):1559-1572. <https://doi.org/10.1109/TPAMI.2018.2840695>
- Xie AN, Harrison J, Finn C, 2020. Deep reinforcement learning amidst lifelong non-stationarity. <https://doi.org/10.48550/arXiv.2006.10701>
- Xu N, Price B, Cohen S, et al., 2016. Deep interactive object selection. IEEE Conf on Computer Vision and Pattern Recognition, p.373-381. <https://doi.org/10.1109/CVPR.2016.47>
- Ye QH, Gao Y, Ding WP, et al., 2022. Robust weakly supervised learning for COVID-19 recognition using multi-center CT images. *Appl Soft Comput*, 116:108291. <https://doi.org/10.1016/j.asoc.2021.108291>
- Yu LQ, Wang SJ, Li XM, et al., 2019. Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation. Proc 22nd Int Conf on Medical Image Computing and Computer-Assisted Intervention, p.605-613. https://doi.org/10.1007/978-3-030-32245-8_67
- Zhang KQ, Yang ZR, Basar T, 2021. Decentralized multi-agent reinforcement learning with networked agents: recent advances. *Front Inform Technol Electron Eng*, 22(6):802-814. <https://doi.org/10.1631/FITEE.1900661>
- Zhang SY, Liew JH, Wei YC, et al., 2020. Interactive object segmentation with inside-outside guidance. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12231-12241. <https://doi.org/10.1109/CVPR42600.2020.01225>
- Zhuang XH, Shen J, 2016. Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI. *Med Image Anal*, 31:77-87. <https://doi.org/10.1016/j.media.2016.02.006>

List of supplementary materials

- 1 More related works
 - 2 More visualizations
 - 3 Robustness of MECCA
 - 4 Comparison of baseline responses to the same user interaction
- Fig. S1 MECCA segmentation process
 Fig. S2 Qualitative segmentation results of MECCA for the BraTS2015 validation set
 Figs. S3-S5 Results of different methods' responses to the same user interactions according to the same initial segmentation on different testing instances and different channels for the Liver dataset in Medical Segmentation Decathlon
 Table S1 Dice of our method which varies with the number of interactions under different cases
 Table S2 MECCA's tolerance to inaccurate interaction points