You Need Glimpse Before Segmentation: Stochastic Detector-Actor-Critic for Medical Image Segmentation

Zhenghua Xu, Yunxin Liu, Di Yuan, Bo Li, Weipeng Liu, Thomas Lukasiewicz

***Abstract*— Medical images often contain more redundant background areas than natural images, potentially introducing noise and degrading image segmentation performance. Inspired by doctors' diagnostic processes, where they identify the lesion area before conducting a detailed analysis, we introduce a novel Stochastic Detector-Actor-Critic (SDAC) framework to tackle this challenge. SDAC initially glimpses the entire image using a detector network and policy gradient algorithms to filter out irrelevant background regions and focus on crucial, smaller areas for segmentation. The Actor-Critic algorithm then dynamically creates segmentation masks pixel by pixel without user intervention or coarse masks, forming a robust segmentation module. Both processes are trained jointly to reduce error propagation and ensure stability and ease of implementation. Our experiments on two commonly used medical image segmentation datasets demonstrate that SDAC achieves competitive results comparable to state-of-the-art methods while using 10x fewer parameters than the best-performing baseline in terms of DICE and IoU metrics. We also conduct detailed ablation studies to enhance understanding and facilitate practical use. Furthermore, SDAC performs well in low-resource settings (i.e., 50-shot or 100-shot), making it ideal for real-world scenarios. Its lightweight design make SDAC an excellent baseline for medical image segmentation tasks.**

***Index Terms*— Deep reinforcement learning, Medical image segmentation, Detector Actor Critic.**

I. Introduction

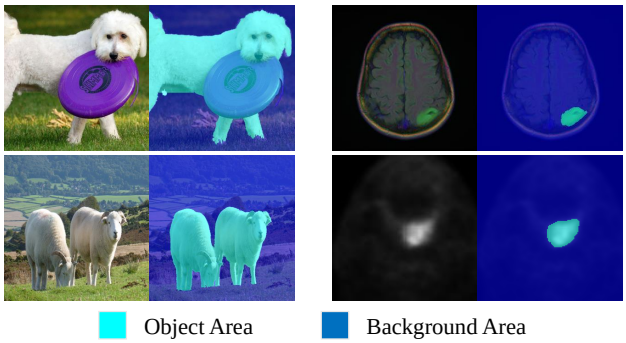
This work was supported by the National Natural Science Foundation of China under the grant 62276089, by the Natural Science Foundation of Tianjin, China, under the grants 24JCJQJC00200 and 24JCQNJC01230, by the Natural Science Foundation of Hebei Province, China, under the grant F2024202064, by the Science Research Project of Hebei Education Department, China, under the grant BJ2025004, by the Ministry of Human Resources and Social Security, China, under the grant RSTH-2023-135-1, and by the S&T Program of Hebei under the grant 24464401D. (Corresponding author: Zhenghua Xu; Di Yuan.)(Zhenghua Xu and Yunxin Liu have contributed equally to this work and share the first authorship.)

This work did not involve human subjects or animals in its research.

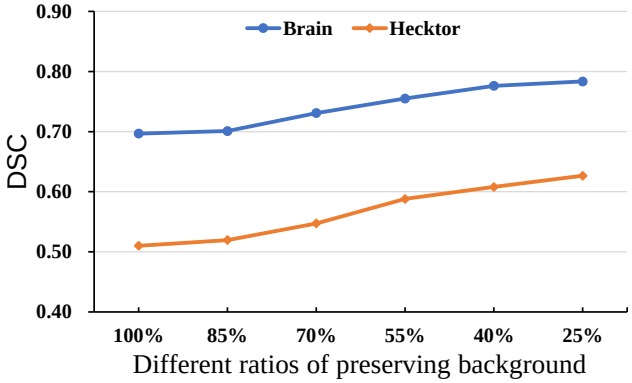
Zhenghua Xu, Yunxin Liu, Di Yuan and Bo Li are with State Key Laboratory of Reliability and Intelligence of Electrical Equipment, School of Health Sciences and Biomedical Engineering, Hebei University of Technology, Tianjin, China (e-mail: zhenghua.xu@hebut.edu.cn; liudearbreeze@gmail.com; yuandi.hn@gmail.com; deepblue.lb@gmail.com).

Weipeng Liu is with School of Artificial Intelligence, Hebei University of Technology, Tianjin, China (e-mail:liuweipeng@hebut.edu.cn).

Thomas Lukasiewicz is with Department of Computer Science, University of Oxford, Oxford, United Kingdom, and the institute of Logic and Computation, Vienna University of Technology, Vienna, Austria (e-mail:thomas.lukasiewicz@cs.ox.ac.uk).



(a) Comparing natural (left) and medical (right) images in terms of content.



(b) Comparing the model segmentation performance of removing different ratios of background information from medical images.

Fig. 1. The demonstrates the significant redundancy present in medical images. (a) Compared to natural images, medical images contain a lot of irrelevant background information. (b) As the background information of medical images is reduced, the segmentation performance can be gradually improved.

SINCE the powerful feature learning ability of deep learning [1]–[3], methods based on deep learning have been widely used in image segmentation. However, when it comes to medical image segmentation, most methods are not ideal in directly processing the entire medical image. Since medical images exhibit greater redundancy (i.e., more low-information areas) compared to natural images, as illustrated in Figure 1(a), medical images have smaller objects, larger background areas, and more uniform backgrounds compared to natural images. Additionally, we annotated 200 small sample datasets based on ground truth for training, validation, and testing to compare the impact of removing different proportions of background

information from medical images on the model's segmentation performance. As shown in Figure 1(b), removing excessive redundant information from medical images significantly improves segmentation performance. This confirms our view that such redundancy in medical images is not only useless but can even negatively impact segmentation performance. Therefore, without appropriately reducing this redundancy, it may lead to an excessive focus on background information during the training process, making the object difficult to focus on or even lost, resulting in poor segmentation performance and resource waste [4]–[6].

Certainly, in response to the above issues, some methods have also attempted to first identify key areas to reduce some of the redundancy in the image before processing. However, these methods have several problems: (i) Low constraint: these methods [7], [8] typically employ attention mechanisms to aid model learning, but by merely assigning importance weights to each pixel, they inevitably still focus on irrelevant areas; (ii) High training cost: such methods [4] are based on a detection-segmentation cascade approach, lacking effective integration during the training process; (iii) Non-uniqueness of selected key regions: while these methods [6] often select multiple regions, they can only reduce information redundancy to a certain extent and cannot accurately determine the object's location. Overall, the lack of modeling capabilities for multi-stage behaviors in deep learning leads to suboptimal model performance. Considering that this process aligns well with human behavioral logic. Therefore, based on the typical diagnostic process where doctors first identify lesion points and then perform detailed analysis on these points, we aim for our model to mimic this more rational process. However, the decision-making process of doctors in this procedure is evidently non-differentiable. Hence, we introduce deep reinforcement learning to model the entire process [9].

To this end, we propose a new Stochastic Detector-Actor-Critic (SDAC) framework for medical image segmentation tasks. Specifically, we first employ a Detector network optimized using policy gradient algorithms to "glimpse" the entire image and remove the most irrelevant background regions, thus identifying smaller segmentation areas. This significantly accelerates iterative training and inference speeds while reducing irrelevant background interference, allowing the model to focus more on the segmentation objects, and this process only needs to be done once per image. Subsequently, based on this region, we utilize the Actor-Critic (AC) [10] algorithm to directly select appropriate actions for each agent (i.e., each pixel) to change the current state of the agent (i.e., whether the pixel belongs to the segmentation object or background), dynamically generating masks pixel by pixel. This is an end-to-end process that requires no user intervention or coarse segmentation masks. Additionally, since most irrelevant background regions have been removed from the selected area, only a very lightweight network is needed to accomplish the entire segmentation process. It is worth noting that the entire process of our proposed SDAC framework is trained together, making the entire framework more stable and easy to train. We evaluated our method on two public datasets, where our approach outperforms all baselines while using fewer parameters. Additionally, testing on extremely small datasets demonstrated that our method exhibits stronger robustness.

In summary, this work's main contributions are as follows:

- To address the shortcomings of the current approach of locating before processing in medical image tasks, we propose a novel SDAC framework. To the best of our knowledge, in deep-reinforcement-learning-based medical image segmentation, we are the first to combine the dynamic selection of regions of interest with precise segmentation objects.
- We model the entire general diagnostic process of doctors as a reinforcement learning problem, dividing it into detect and segmentation stages. Based on the different objectives of each stage, we design appropriate rewards and algorithms for optimization. Both stages can be trained and updated together to ensure their cooperation during the training process.
- Extensive experimental studies are conducted using SDAC on two public medical image segmentation datasets to show the effectiveness of SDAC. Specifically, SDAC achieves segmentation performance that outperforms all baselines using a significantly reduced number of parameters. In addition, ablation studies are conducted to demonstrate that the Detector network and PixelSeg are both effective and complement to each other. Finally, we prove that SDAC can still maintain good segmentation performance in extremely small datasets.

II. Related Work

This Section summarizes related works on cascade models based on detection-segmentation solutions and pixel-level deep reinforcement learning for medical image segmentation, concludes where these methods need to be improved, and introduces the advantages of our framework.

Based on cascade models. U-Net [11] can reduce information loss because of its symmetrical U-shaped structure and skip connections. More and more work is based on U-Net to achieve high-precision medical image segmentation [12], [13]. However medical images have obvious information redundancy problems due to the small proportion of segmentation objects. Therefore, Many works use attention mechanisms, such as AttnUNet [7], to make the model pay more attention to effective information and avoid interference information. While these attention mechanisms often significantly improve the model's performance, they all must work on the entire image, which requires resizing the original image to a lower resolution or using sliding windows to extract patches from the image. These methods inevitably lead to information loss and/or high computational costs [14]. However, these only change the proportion of interfering information and do not completely avoid interfering information. Therefore, there are still many works referring to the doctor's diagnosis process, using the option Crop [15], or using a cascade model based on detection-segmentation structure [4], [6], [8], first locating the segmentation object area through the detection network, and then inputting the area into the segmentation network to achieve further accurate segmentation. However, in these

methods, the detection and segmentation networks are trained independently and it requiring the detection network to be trained first, followed by the segmentation network, and then intuitively combining the two. This not only leads to high training costs but also results in suboptimal segmentation performance. In contrast, our method formulates the detection and segmentation stages as a unified deep reinforcement learning problem, allowing both to be updated jointly. This integration makes the entire framework more stable and easier to train, ultimately achieving accurate localization and precise segmentation.

It is worth noting that our method is significantly different from multi-task learning approach. First, although the detection network, policy network, and value network share the same feature extractor, the detection network is not performing a conventional object detection task. Its goal is merely to roughly identify the location of the segmentation target to remove irrelevant background information, rather than to precisely enclose the full volume of the target. Second, during training, the loss functions of the two stages are not combined in a weighted manner but are relatively independent. Finally, the output of the detection network is not a standalone result, it serves as input for the subsequent policy and value networks. There is a clear sequential relationship: the segmentation stage begins only after the detection network completes its localization task, and the accuracy of this localization has a significant impact on the segmentation outcome. Therefore, although our method may appear to involve two tasks, the role of the probing network is to eliminate redundant information in medical images as much as possible to support the subsequent segmentation task. Segmentation remains the sole objective of the entire method.

Based on deep reinforcement learning. Inspired by the successful application of deep reinforcement learning in playing video games [16]–[18], many deep reinforcement learning algorithms applied in medical image analysis [19], [20], especially medical image segmentation [21], [22], have been widely explored and applied. Most works have focused on using an iterative refinement [23] based on deep reinforcement learning to improve the segmentation performance in medical images. For instance, [24] proposes a multi-agent reinforcement learning approach with user interaction to capture voxel dependencies for medical image segmentation while reducing the exploration space to a manageable size. [25] proposes an end-to-end policy strategy to emulate the progressive delineation of a region of interest on medical images, which starts from a coarse result and refines it into a finer result using a set of threshold values (i.e., actions). However, these methods are not only not automatic (requiring user interaction or providing coarse segmentation results), but also the segmentation results of artificially setting the threshold range are not objective and ideal.

Inspired by PixelRL [26], using the asynchronous advantage actor-critic (A3C) [27] as the backbone to achieve the pixel-level denoise, we propose a new pixel-level deep reinforcement learning segmentation model (PixelSeg) using a dynamic iterative update policy. Different from existing methods, we directly input the output of the Detector network into PixelSeg, only design two actions $a \in \{0,\ do\ noting\}$, and directly output the action with a large probability value by the policy network. Therefore, PixelSeg is not only a fully automatic segmentation network but also makes the segmentation results more ideal and accurate by avoiding artificially designed threshold ranges.

III. Stochastic Detector-Actor-Critic

A large amount of work has successfully imitated the process of doctors marking segmentation objects, that is, further segmenting the objects after the detection network detects the key area, thereby achieving accurate segmentation while reducing redundant information in medical images. However, these methods are implemented using independently updated detection and segmentation networks and cannot ideally reduce interference information in medical images. Therefore, we propose a new Stochastic Detector-Actor-Critic framework, SDAC, which can use the Detector network to generate ideal bounding boxes and use PixelSeg for further accurate segmentation. SDAC models this process as deep reinforcement learning, achieving integrated updates of the two networks by designing appropriate rewards for each stage. The framework diagram of SDAC is shown in Figure 2. Based on the effectiveness of U-Net in medical image segmentation, our framework also uses U-Net as the backbone. Because of the significant reduction of redundant information after processing by the Detector network, a lightweight network can be used to implement the entire process. Therefore, in the feature extraction module, Detector network, policy network and value network of PixelSeg, the convolutional layers of the original U-Net are replaced with MobileNetV2 convolutional layers [28]. Since the Detector network is only used for positioning and does not require too much feature information, it does not include the skip connection. The detailed network structure of the Stochastic Detector-Actor-Critic is shown in Table II.

A. Detector Network

Given an original image X, after passing through the symmetric feature extraction module and Detector network, a preliminary bounding box D_{pre} is obtained. The size of D_{pre} is only one-quarter of the original image X. In theory, the time it takes to iterate D_{ori} four times is equivalent to the time it takes to infer about a complete image X, which can reduce interference information and reduce time costs. Following the policy gradient context [29], our Detector network is considered as a policy, the original image X is viewed as the state s, the position of D_{pre} as the action a, and the reward signal R_{det} is output based on the position of D_{pre}. Our goal is to maximize the expected reward $\mathcal{J}(\theta_d)$ and find the optimal policy:

$$\mathcal{J}(\theta_d) = E_d(a;\theta_d)[R_{det}], \tag{1}$$

where $E_d(a;\theta_d)$ is the expected reward after taking action a, and θ_d is the model's parameter. The policy $\pi(a|s)$, represents a policy (i.e., probability distribution) for taking action a in state s, and is learned through back-propagation, which

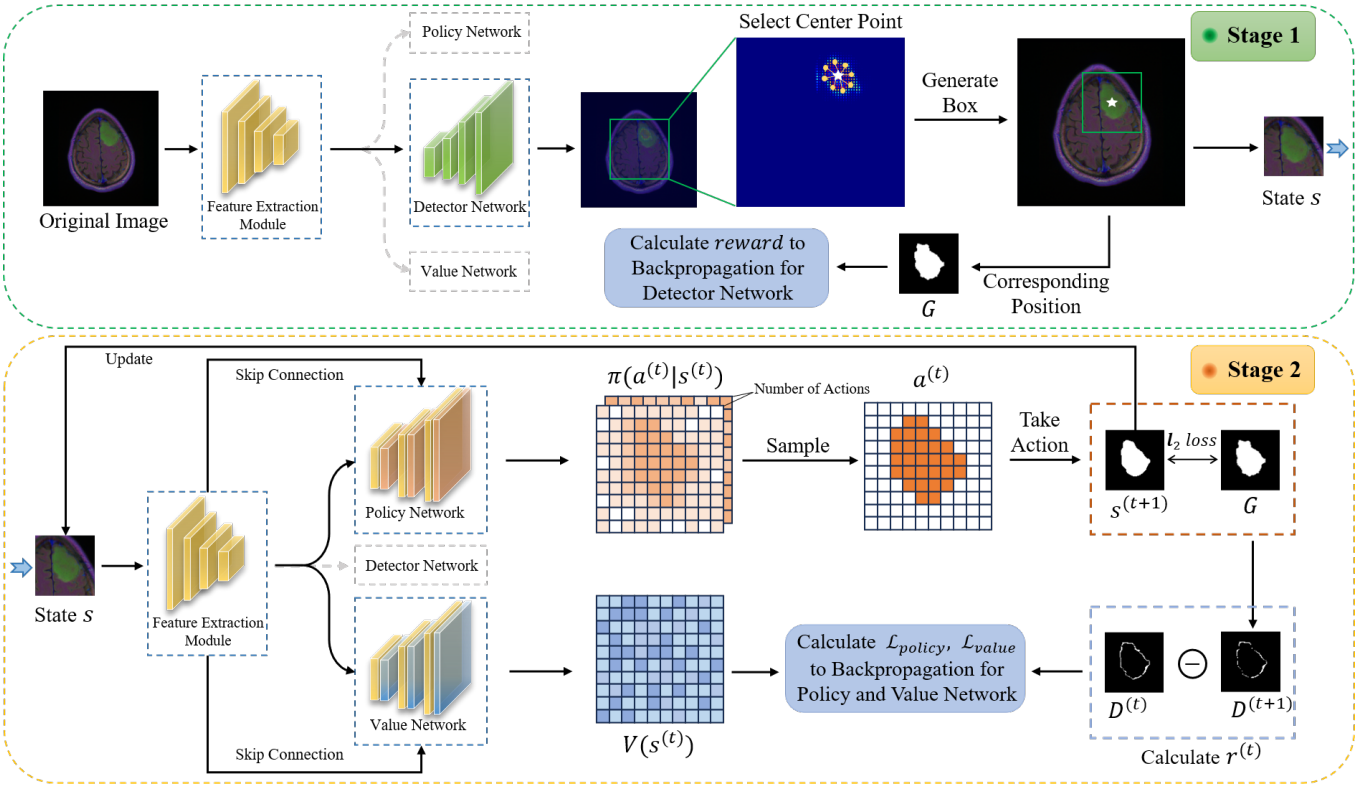


Fig. 2. The framework of the proposed SDAC, $s(t=0)$ is selected through the detector network in the original image, choosing a small region where segmentation objects exist. $X^{(t+1)}$ is the temporary input in step $t+1$, D is the difference between s^t and the ground truth G, $a^{(t)}$ is sampled from the policy $\pi : a^{(t)} \sim \pi(a^{(t)}|s^{(t)})$. The entire process is trained together, not separately. For each image, only in the first step is the state $s(t=0)$ determined through the detector network. The subsequent segmentation refinement process is carried out using only the policy network and value network.

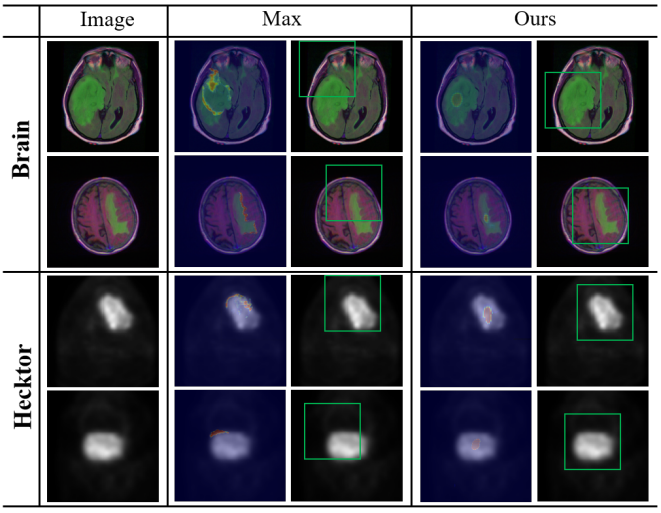


Fig. 3. Comparison of different methods for selecting the center point of the bounding box in the detector network. The bounding box for Max operation is towards the upper left. The green in Brain and the white in Hecktor are the best views.

requires the definition of the gradient of the expected reward R_{det} for the model parameters. The gradient can be defined as:

$$\nabla_{\theta_d} \mathcal{J}(\theta_d) = E_d(a;\theta_d)[\nabla_{\theta_d} \log \pi(a|s;\theta_d) \cdot R_{det}]. \quad (2)$$

The expected reward is not directly estimable and necessitates an approximation approach. As is common practice in the traction of policy gradient, this approximation can be attained through the utilization of the negative log-likelihood loss. This loss function is differentiable with respect to the model parameters and can be effectively adjusted by the reward signal, leading to the formulation of the policy loss of the Detector network as shown below:

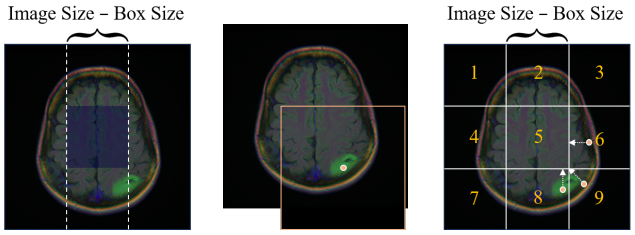
$$\mathcal{L}_{det}(\theta_d) = -\alpha \log \pi(a|s;\theta_d) \cdot R_{det}, \quad (3)$$

where α is the learning rate. At the beginning of the training, D_{pre} is random, and we hope that D_{pre} can cover the segmentation object as much as possible. Therefore, we design an appropriate reward function to encourage this action. The reward function is as follows:

$$R_{det} = \begin{cases} -1 & P_B = 0, \\ 1 & P_B = P_G, \\ \frac{P_B}{P_G} - 1 & 0 < P_B < P_G, \end{cases} \quad (4)$$

where P_G is the total number of pixels contained in the ground truth and P_B refers to the total number of pixels containing the segmentation object in the bounding box. Then D_{pre} is continuously updated through the policy gradient algorithm so that the Detector network can select the optimal bounding box D_{opt} that maximizes the reward.

However, the center points obtained directly through sampling with policy gradient algorithms are not actually optimal.



(a) Crop (b) Padding (c) Shift

Fig. 4. Different methods to adjust the center point of the bounding box for segmentation objects at the edge of the image. (a) Crop operation. Choosing the center point of the box by taking only the central part of the probability map can prevent the box from exceeding the image boundaries. However, when the box size is large, it may miss segmented objects in the edge regions. (b) Padding operation. Padding the parts of the box that exceed the image with zeros. (c) Shift operation. By moving the center point of the box that exceeds the image range to the nearest central region. For example, a point in regions $\mathbf{1}$, $\mathbf{3}$, $\mathbf{7}$, and $\mathbf{9}$ would be moved to the corner of central region $\mathbf{5}$, and a point in regions $\mathbf{2}$, $\mathbf{4}$, $\mathbf{6}$, and $\mathbf{8}$ would be shifted to the boundary of central region $\mathbf{5}$.

For example, for a 256×256 image with a bounding box size of 128×128, there may be many center points in the image that can draw a box containing the segmentation object. This is due to the uncertainty inherent in sampling, where even a small probability can lead to the selection of poorly located points. Of course, the most intuitive solution is to choose the point with the highest expected value as the center of the optimal bounding box. However, according to our designed reward function in Equation 4, the point with the highest expectation is not unique, and among these potential points with the highest expected values, some are off-center from the segmentation object. As shown in Figure 3, since the traversal process selects the first maximum value encountered from left to right and from top to bottom, directly choosing the point with the highest expectation tends to result in bounding boxes predominantly located in the upper left corner of the image, leading to the drawn boxes not fully encompassing the segmentation object.

Therefore, we aim to select a center point for the bounding box that is as close to the center of the segmentation object as possible. Specifically, we select multiple points with high expectations, within the numerical range of $[E_{max} - \epsilon, E_{max}]$, where E_{max} is the maximum expected value and ϵ is a small number set to $5e-3$. Then, we calculate the centroid of these points to serve as the center point of the bounding box, ensuring as much as possible that the chosen center point is close to the center of the segmentation object. The specific effect is shown in Figure 3, demonstrating that compared to directly selecting the point with the highest expectation, our method can ensure that the drawn box fully contains the segmentation object, even if the shape of the segmentation object is irregular.

After determining the sampling method, we also need to consider the issue of the bounding box exceeding the size of the image. After all, the segmentation objects of medical images are not always located in the center of the image; they may appear at the edges, potentially causing the bounding box drawn based on the center point to extend beyond the image boundaries. Therefore, we need to appropriately adjust the position of the bounding box's center point for such cases. We have listed several viable methods as shown in Figure 4, and in this work, we chose option Shift (shown in Figure 4 (c)) as our default method for adjusting the center point of the bounding box. This decision is made because (i) Setting the box size too large with option Crop (shown in Figure 4 (a)) can reduce the visible range, making it impossible to observe segmentation objects at the edges. (ii) Compared to option Padding (shown in Figure 4 (b)), option Shift is simpler to implement in coding and does not introduce irrelevant information, although this effect is negligible. In Section IV, we also quantitatively validate these methods from the perspective of box size settings and localization performance.

During the training process, the Detector network encourages the model to consider even less distinctive features by probing different locations within the image. Specifically, the model is expected to learn a rich set of features to address the segmentation problem of segmentation objects located in various positions. One might question if the Detector network could generate many useless features. For instance, during the initial phase of training, patches either outside the segmentation object area or covering only a part of the segmentation object could disrupt the learning of a good representation. However, this can be mitigated through joint optimization with the policy and value networks of deep reinforcement learning. Given that the segmentation loss based on the Actor-Critic model predominates over the Detector network, unnecessary features are discarded throughout the training iterations. To support our point of view, in the ablation studies in Section IV, we examine the class activation maps [30] of models trained solely with Actor-Critic, and SDAC with the Detector network serving as an auxiliary task.

B. PixelSeg

The initial state $s^{(t)}$ obtained from the Detector network first passes through the feature extraction module and is input to the policy network and value network. Then, like AC-based deep reinforcement learning methods, the policy network uses policy gradient to optimize the policy to select an action; the value network evaluates the score through the value function based on the action; the policy network then optimizes the policy based on the score to obtain the highest reward.

However, most existing deep reinforcement learning methods adjust the pixel values of probability maps by manually designing threshold ranges. For example, for each agent, $a \in \{0.1,\ 0.2,\ -0.1,\ -0.2,\ 0\}$ is selected to gradually adjust the pixel value to approximate the ground truth. These methods have two obvious drawbacks: First, these methods can easily lead to unsatisfactory segmentation accuracy due to the lack of objectivity in artificially setting the threshold range, such as over-segmentation or under-segmentation. Besides, the number of output layers of the policy network is equal to the number of actions. To achieve a better segmentation performance, multiple actions need to be set, which will easily increase the computational cost.

Therefore, SDAC designs a new PixelSeg network, using the A3C algorithm to directly select the appropriate action

$a \in \{0,\ donoting\}$ for each agent (i.e., pixel), that is, selecting the action with a large probability value. This can avoid human intervention, select appropriate actions more objectively, and reduce the high computational cost caused by setting multiple actions. Except for the different action selection solutions, PixelSeg's parameter update and loss functions are similar to A3C. The specific process is as follows. The goal of the Policy network is to learn the policy function to maximize the cumulative reward. Its objective function $\mathcal{J}(\pi)$ is as follows:

$$\mathcal{J}(\pi) = E_{p^s}[V(s)], \tag{5}$$

where $V(s)$ is the expected discounted return (i.e., estimated value).It also uses the policy gradient in Equation 2 to update the policy, so the parameter update and loss function are as follows:

$$\nabla_{\theta_p}\mathcal{J}(\theta_p) = \nabla_{\theta_p}\log\pi(a|s;\theta_p)\cdot A(s,a;\theta_p,\theta_v), \tag{6}$$

$$\mathcal{L}_p(\theta_p) = -\nabla_{\theta_p}\mathcal{J}(\theta_p), \tag{7}$$

where θ_p is the parameter of policy network, and θ_v is the parameter of value network. $A(s,a;\theta_p,\theta_v)$ is the Advantage function, indicating the advantage of action a in state s, The goal of Value is to minimize the error of the value function, that is, the difference between the estimated value and the actual cumulative reward. Its parameter update and loss function are as follows:

$$\nabla_{\theta_v}\mathcal{J}(\theta_v) = \frac{1}{2}(R_{drl} - V(s;\theta_v))^2, \tag{8}$$

$$\mathcal{L}_v(\theta_v) = \frac{1}{2}(R_{drl} - V(s;\theta_v))^2, \tag{9}$$

$$r^{(t)} = ||s^{(t)} - G||^2 - ||s^{(t+1)} - G||^2, \tag{10}$$

$$R_{drl} = r^t + \gamma r^{t-1} + \gamma^2 r^{t-2} + \cdots + \gamma^t r^0, \tag{11}$$

where γ is the discount rate; R_{drl} represents the cumulative reward, where the reward r^t at each step is calculated by the l_2 loss between the segmentation map of the previous step t and the ground truth G, subtracted by the l_2 loss between the segmentation map of the current step $t+1$ and the ground truth G. Therefore, the final loss function of PixelSeg is the sum of the policy and value loss functions. In practice, this function usually adds some regularization or other optimization techniques to help the stability and convergence speed of training. Specifically, we first use the advantage function $A(s,a)$ as a baseline during policy network updates to retain the relative quality of actions while filtering out the inherent value of the state itself, thereby reducing the variance of the gradients. Second, our method does not require an experience replay buffer, which not only saves a significant amount of memory but also avoids the bias introduced by data correlation within the buffer. Finally, we strictly constrain the range of rewards to prevent gradient explosion that could lead to training instability. In particular, during the update of the detector network, rewards are clipped within the range of [-1,1]. For the policy and value network updates, since the rewards are designed based on relative changes from the previous timestep rather than absolute values, which inherently keeps the reward values within a controllable range.

Algorithm 1 Stochastic Detector Actor-Critic

Input: Images and labels (X, G); parameters θ_f, θ_d, θ_p, and θ_v for feature extraction module F, Detector network D, policy network P, and value network V, respectively.
Output: F, D, P, and V.

for each iteration T **do**
  Sample (x, g) from the (X, G)
  Thread-specific parameters $\theta'_p = \theta_p$, $\theta'_v = \theta_v$, $\theta'_d = \theta_d$
  Obtain state $s^{(t)}$ from D
  for each thread step t **do**
    Perform $a^{(t)}$ according to policy $\pi(a^{(t)}|s^{(t)};\theta'_p)$
    Obtain the output $y^{(t)}$
    Receive reward $r^{(t)}_{drl}$ and new state $s^{(t+1)}$
  end for
  Obtain the output y
  Calculate the rewards R_{det} and R_{drl} in Equ. 4 and 11
  for each gradient step k **do**
    θ'_d: $d\theta_d \leftarrow d\theta_d + \mathcal{L}_{det}(\theta_d)$ in Equ. 3
    θ'_p: $d\theta_p \leftarrow d\theta_p + \mathcal{L}_p(\theta_p)$ in Equ. 7
    θ'_v: $d\theta_v \leftarrow d\theta_v + \mathcal{L}_v(\theta_v)$ in Equ. 9
    θ'_f: $d\theta_f \leftarrow d\theta_f + \mathcal{L}_{det}(\theta_f) + \mathcal{L}_p(\theta_f) + \mathcal{L}_v(\theta_f)$ in Equ. 12
  end for
  Update θ_d, θ_p, θ_v, and θ_f with $d\theta_d$, $d\theta_p$, $d\theta_v$, and $d\theta_f$
  Update F, D, P, and V with θ_d, θ_p, θ_v, and θ_f
end for

C. Training

During the training process, SDAC inputs the key area generated by the Detector network as the initial state $s^{(t)}$ to PixelSeg, and then obtains the segmentation output y based on multiple iterations of PixelSeg, and then obtains the rewards of the Detector network and PixelSeg according to Equation 4 and 11. Then, the parameter updates and loss functions of the Detector, policy, and value networks are obtained according to the policy gradient algorithm. Besides, both the Detector network and PixelSeg use the same feature extraction module, the parameter of the feature extraction module is jointly updated by the sum of the parameter updates of the two, as shown in Equation 12. Repeat the above training process until the termination condition is reached, and the integrated update of the Detector network and PixelSeg can be achieved. The complete algorithm of SDAC is described in Algorithm 1.

$$\begin{aligned}\nabla_{\theta_f}\mathcal{J}(\theta_f) =& E_d(a;\theta_f)[\nabla_{\theta_f}\log\pi(a|s;\theta_d)\cdot R_{det}] \\ &- \nabla_{\theta_f}\log\pi(a|s;\theta_p)\cdot A(s,a;\theta_p,\theta_v) \\ &+ \frac{1}{2}(R_{drl} - V(s;\theta_v))^2.\end{aligned} \tag{12}$$

IV. Experiments

We have conducted extensive experiments to evaluate the superiority of the proposed SDAC. This section first introduces information about datasets, experimental settings, evaluation metrics, and baselines. Then, to verify that our framework improves the segmentation performance more effectively than

TABLE I
Datasets information.

Datasets	Quantity	Image size	Modality	Challenge	Source
Brain [31]	3,929	256×256	MRI	Extremely irregular segmentation edges	The Cancer Imaging Archive
Hecktor [32]	28,949	144×144	PET	Ambiguous segmentation edges and objects	HECKTOR Challenge 2020

the latest baselines, we conduct an extensive experimental study to compare the performance of SDAC with baselines. Besides, ablation studies are conducted to further demonstrate the necessity of each module and SDAC's effectiveness in extremely small datasets. Finally, we also conducted supplementary experiments to verify the impact of different solutions by adjusting the center point of the bounding box for the segmentation edges, changes in bounding box size, and changes in the number of thread steps on segmentation performance.

A. Datasets and Metrics

The empirical studies over two publicly available confirm that our framework beat other baselines. (i) The *Barin* dataset [31] is a public MRI dataset designed for brain tumor segmentation released by the Cancer Imaging Archive. Each case has 256×256 images with the number of slices varying from 40 to 176. The segmentation challenge of this dataset is that the segmentation targets change greatly. (ii) The *Hecktor* dataset [32] was released by the Hecktor challenge hosted at MICCAI 2020 for head and neck tumor segmentation. It contains 201 3D head and neck PET-CT scans. In this work we only use PET modality that are convenient for human intuitive perception. The segmentation challenge of this dataset lies in the fact that the object boundaries are relatively blurred and exhibit significant misleading features. Details of the dataset are shown in Table I. There are 70% of the datasets for training, 10% for validation, and 20% for testing. All the results are obtained by running five times to obtain the average.

We use six metrics on two datasets to show the effectiveness of our framework: DICE evaluates the overlap value between the outputs and the ground truths. Positive Predictive Value (PPV) measures the percentage of true positive samples of all predicted positive samples. Sensitivity (SEN) has evaluated the probability that positive samples are correctly classified as positive. IoU measures the accuracy of corresponding objects in a specific dataset. Boundary IoU (BIoU) [33] is a widely used boundary-based metric. Hausdorff Distance (HD) [34] is a widely used distance-based metric, here we use HD95 to eliminate the impact of a small subset of the outliers. It's worth noting that higher values for these metrics, except HD95, mean better performance. Formally,

$$DSC = \frac{2 * TP + \epsilon}{T + P + \epsilon}, \qquad PPV = \frac{TP + \epsilon}{TP + FP + \epsilon},$$
$$SEN = \frac{TP + \epsilon}{TP + FN + \epsilon}, \quad IoU = \frac{TP + \epsilon}{T + P - TP + \epsilon},$$
$$BIoU = \frac{G_d \cap P_d}{G_d \cup P_d},$$
$$HD95 = max_{k95\%}[d(P, G), d(G, P)],$$

where TP, FP, and FN are the number of true positive points, false positive points, and false negative points, respectively. T is the count of ground truth points for the respective class, P corresponds to the number of predicted positive points, G stands for the total number of ground truth positive points, P_d represents the count of predicted positive boundary points, G_d is the number of ground truth positive boundary points, and $d(*)$ denotes a function for calculating surface distance. We set the $\epsilon = 1e-4$ to avoid zero division.

B. Baselines

To evaluate the performances of the proposed SDAC, we choose 15 baselines. First, the eight common advanced U-Net-based image segmentation methods are selected as baselines. U-Net [11] is the backbone of almost all medical image segmentation solutions; Crop [15] is a common operation to reduce image background information; AttnUNet [7] is the first classic model that combines the attention mechanism with U-Net; Cascade [4] is the latest detection-segmentation cascade networks. AAU-Net [8] is a hybrid adaptive attention module that combines channel self-attention blocks and spatial self-attention blocks to replace the traditional convolution operation. MobileViT [35] is a lightweight Transformer model. nnUNet [36] automates preprocessing and post-processing according to different task. nnFormer [37] combines the strengths of Convolutional Neural Networks and Transformers by proposing a hybrid backbone architecture that interleaves convolutional operations with self-attention mechanisms. MedNeXt [38] is a Transformer-inspired large convolution kernel segmentation network. GAE [6] is a multi-region method that performs active visual exploration using the self-supervised attention mechanism and contrastive learning to select multiple bounding boxes with the highest information gain.

Then, the five deep-reinforcement-learning-based methods are chosen as baselines, using U-Net as their network framework. DQN [39] first combines Q-Learning and deep learning to solve the instability problem; Double DQN [40] based on DQN uses different value functions to select and evaluate actions to solve the problem of overestimation; Dueling DQN [41] divides the last layer of DQN into two parts to obtain a more robust learning effect; AC [10] and A3C [27] use Actor and Critic to reduce the variance of gradient estimation.

C. Implementation Details

Our experiments are implemented using PyTorch[1] and run on a single NVIDIA 2080Ti GPU. We evaluate our model on two public datasets using the same experimental setup,

[1]link: https://pytorch.org/

TABLE II
NETWORK ARCHITECTURE OF STOCHASTIC DETECTOR-ACTOR-CRITIC, IN WHICH THERE ARE THREE DOWNSAMPLING LAYERS IN THE FEATURE EXTRACTION NETWORK (I,E., $i = 1, 2, 3$ AND c_0, c_1, c_2, AND c_3 SET TO 16, 24, 46, AND 64, RESPECTIVELY).

Feature Extraction Network		**Input Shape**	**Output Shape**
Input layer	Conv2d, BN2d, ReLU6	$(3, h, w)$	(c_0, h, w)
Downsampling layer i	MobileNetV2 Block	$(c_{i-1}, h/2^{i-1}, w/2^{i-1})$	$(c_i, h/2^i, w/2^i)$
Detector Network		**Input Shape**	**Output Shape**
Upsample, MobileNetV2 Block		$(c_3, h/8, w/8)$	$(c_2, h/4, w/4)$
Upsample, MobileNetV2 Block		$(c_2, h/4, w/4)$	$(c_1, h/2, w/2)$
Upsample, MobileNetV2 Block		$(c_1, h/2, w/2)$	(c_0, h, w)
Conv2d, BN2d, ReLU6		(c_0, h, w)	$(1, h, w)$
Policy Network		**Input Shape**	**Output Shape**
Upsample, MobileNetV2 Block		$(c_3 + c_2, h/8, w/8)$	$(c_2, h/4, w/4)$
Upsample, MobileNetV2 Block		$(c_2 + c_1, h/4, w/4)$	$(c_1, h/2, w/2)$
Upsample, MobileNetV2 Block		$(c_1 + c_0, h/2, w/2)$	(c_0, h, w)
Conv2d,BN2d,ReLU6,SoftMax		(c_0, h, w)	(a, h, w)
Value Network		**Input Shape**	**Output Shape**
Upsample, MobileNetV2 Block		$(c_3 + c_2, h/8, w/8)$	$(c2, h/4, w/4)$
Upsample, MobileNetV2 Block		$(c_2 + c_1, h/4, w/4)$	$(c1, h/2, w/2)$
Upsample, MobileNetV2 Block		$(c_1 + c_0, h/2, w/2)$	(c_0, h, w)
Conv2d,BN2d,ReLU6		(c_0, h, w)	$(1, h, w)$

TABLE III
The segmentation results of the proposed method and baselines on two datasets. **Bold** represents the best result, and underlined is the second best result.

Model	Brain						Hecktor						Params
	DICE ↑	PPV ↑	SEN ↑	IoU ↑	BIoU ↑	HD95 ↓	DICE ↑	PPV ↑	SEN ↑	IoU ↑	BIoU ↑	HD95 ↓	
U-Net	0.7220	0.7783	0.6734	0.6022	0.1932	13.2416	0.5976	0.6541	0.6449	0.4839	0.0966	7.3645	31.03M
Crop	0.7309	0.7888	0.6808	0.6185	0.1973	12.0199	0.6064	0.6416	0.6642	0.4985	0.0986	5.9203	31.03M
AttnUNet	0.7536	0.7956	0.6940	0.6402	0.2079	12.2775	0.6351	0.6661	0.6982	0.5224	0.1053	6.5005	32.43M
Cascade	0.7582	0.8571	0.7109	0.6594	0.2274	10.1332	0.5982	0.6542	0.6459	0.4846	0.0969	7.3184	72.37M
AAU-Net	0.7708	0.8522	0.7058	0.6863	0.2384	10.0421	0.6555	0.6684	0.6506	0.5482	0.1099	5.1248	104.62M
MobileViT	0.8049	0.8649	0.7658	0.7020	0.2425	9.5348	0.6326	0.6604	0.6819	0.5208	0.1023	6.8047	2.28M
nnUNet	0.7943	0.8534	0.7586	0.7045	0.2436	10.9742	0.6452	0.6534	0.6881	0.5334	0.1044	5.9535	31.20M
nnFormer	0.7818	0.8482	0.7434	0.7002	0.2413	11.0674	0.6335	0.6505	0.6795	0.5288	0.0998	6.8431	28.81M
MedNeXt	0.8110	0.8819	0.7700	0.7192	0.2483	9.2904	0.6423	0.6527	0.6899	0.5325	0.1036	6.4175	10.48M
GAE	0.7956	0.8638	0.7364	0.6942	0.2410	9.8755	0.6211	0.6460	0.6596	0.5096	0.0910	6.9676	155.85M
DQN	0.7224	0.7816	0.6479	0.6070	0.1918	13.5471	0.6025	0.6472	0.6635	0.4806	0.0976	7.7744	7.77M
Double DQN	0.7432	0.7987	0.7081	0.6330	0.2077	12.0447	0.6283	0.6578	0.6686	0.5245	0.1013	7.1747	7.77M
Dueling DQN	0.7580	0.8087	0.7133	0.6526	0.2253	10.6835	0.6302	0.6588	0.6734	0.5170	0.1044	6.7688	7.77M
AC	0.7770	0.8375	0.7202	0.6758	0.2430	10.2233	0.6440	0.6652	0.6772	0.5328	0.1108	6.2903	10.82M
A3C	0.7804	0.8401	0.7234	0.6832	0.2484	10.1355	0.6546	0.6681	0.6835	0.5466	0.1118	6.2111	10.82M
SDAC	**0.8267**	**0.8851**	**0.7715**	**0.7272**	**0.2662**	**8.6590**	**0.6751**	**0.6827**	**0.7007**	**0.5630**	**0.1232**	**4.7915**	**1.10M**

including network architecture, and training hyperparameters. We use the popular optimizer *Adam* [42] to train SDAC on the public datasets, where the initial learning rate is set to $3e-4$, the learning rate drops by a factor of 0.9 every 200 epoch. The batch size is set to 1. The discount rate γ is set to 0.95. We set the maximum epoch to 1000 and the length of each episode to 10. We present in Table II the structure and design of each layer of the Stochastic Detector-Actor-Critic in detail. For all experiments, the bounding box size is set to one-fourth the size of the original image unless otherwise noted. Finally, for the settings of all baselines, we follow the original works.

D. Main Results

To demonstrate the effectiveness of our proposed SDAC, we conduct experiments on two public datasets and compare the performance of SDAC with baselines. The quantitative experimental results are shown in Table III, and examples of segmentation results of SDAC and baselines are shown in Figure 5. Through quantitative and qualitative analysis, we have the following conclusions.

TABLE IV
P-VALUES FROM T-TEST BETWEEN OUR METHOD AND OTHER STATE-OF-THE-ART METHODS ON DIFFERENT METRICS ON TWO DATASETS.

$vs.$ Method	Brain						Hecktor					
	DICE	PPV	SEN	IoU	BIoU	HD95	DICE	PPV	SEN	IoU	BIoU	HD95
U-Net	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Crop	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
AttnUNet	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	0.0323	$p < 0.01$	$p < 0.01$	$p < 0.01$
Cascade	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
AAU-Net	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	0.0476	$p < 0.01$	0.0367	0.0251	0.0184
MobileViT	$p < 0.01$	$p < 0.01$	0.0448	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
nnUNet	$p < 0.01$	$p < 0.01$	0.0412	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	0.0390	$p < 0.01$	$p < 0.01$	$p < 0.01$
nnFormer	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
MedNeXt	$p < 0.01$	0.0112	0.0108	0.0480	$p < 0.01$	0.0126	$p < 0.01$	$p < 0.01$	0.0368	$p < 0.01$	$p < 0.01$	$p < 0.01$
GAE	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
DQN	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Double DQN	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Dueling DQN	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
AC	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	0.0443	$p < 0.01$
A3C	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	0.0376	$p < 0.01$	$p < 0.01$	0.0287	$p < 0.01$

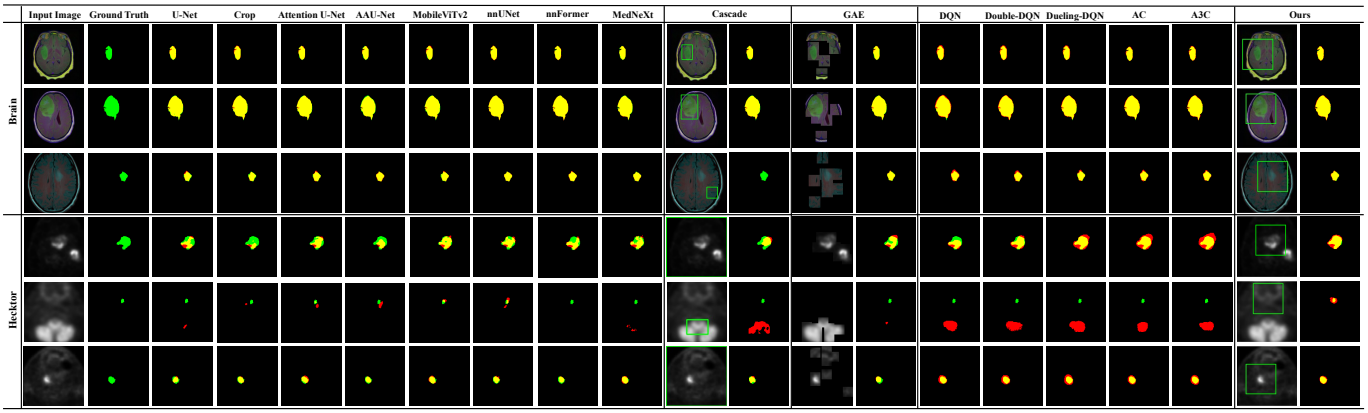


Fig. 5. Examples of visualized segmentation results of the proposed SDAC and the baselines on two public datasets. Red, green, and yellow indicate the prediction, ground truth, and overlapped pixels, respectively.

First, as shown in Table III, SDAC can achieve significant improvements and require fewer parameters than all baselines, which proves that our framework is more effective and computationally cheaper than most existing image segmentation solutions. Specifically, SDAC has significant improvements over deep-learning-based methods (i.e., attention mechanism, cascade models, advanced architectures and multi-region method) and deep-reinforcement-learning-based methods. For instance, on the Brain dataset, SDAC is 1.57%, 0.32%, 0.15%, 0.08%, and 1.79% higher than the suboptimal results (MedNeXt) on DICE, PPV, SEN, IoU, and BIoU, respectively, while utilizing only one-tenth of the parameters (i.e., SDAC vs. MedNeXt is 1.10 vs. 10.48). These findings prove that our framework can achieve better segmentation results by using integrated updates of the Detector network and the deep-reinforcement-learning-based segmentation network. Besides, the number of parameters of SDAC is only half of the model with the second least number of parameters (i.e., SDAC vs. MobileViT is 1.10 vs. 2.28), which proves that our framework can effectively reduce the computational cost.

Furthermore, to evaluate if the improvement of our method is statistically significant, we have conducted the t-test to compare our method with other state-of-the-art methods on different metrics on two public datasets and reported the p-values in which the null hypothesis is assumed to be the two measured samples with the same distribution. As shown in Table IV, the p-values for all metrics are below 0.05, further demonstrating the statistical significance of our method on the two public datasets. Then, six examples in Figure 5 also prove that the segmentation performances of SDAC are closer to ground truths than all baselines. Specifically, SDAC is more accurate in segmentation details than deep-learning-based and deep-reinforcement-learning-based methods. For example, in the upper left area of the first row and the left middle area of the second row in the Brain dataset, SDAC can more accurately segment the edge details of the segmentation objects than all baselines. Besides, comparing existing detection-segmentation methods, we find that SDAC using integrated updates can not only improve the segmentation accuracy, but also improve the superiority of bounding box positioning. For example, Cascade cannot accurately locate the segmentation objects (such as rows 3 and 5), and bounding boxes in the

TABLE V
PERFORMANCE UNDER EXTREME DATA CONSTRAINTS ON PUBLIC DATASETS. **BOLD** REPRESENTS THE BEST RESULTS.

Datasets	Model	50-shot						100-shot					
		DICE ↑	PPV ↑	SEN ↑	IoU ↑	BIoU ↑	HD95 ↓	DICE ↑	PPV ↑	SEN ↑	IoU ↑	BIoU ↑	HD95 ↓
Brain	U-Net	0.6807	0.7518	0.6517	0.5671	0.1604	19.0654	0.7001	0.7651	0.6710	0.5883	0.1826	17.6037
	Crop	0.7074	0.7616	0.6684	0.5913	0.1765	17.1734	0.7220	0.7211	0.7019	0.6072	0.1881	15.7198
	AttnUNet	0.7046	0.7687	0.6730	0.5962	0.1876	16.5067	0.7180	0.7707	0.6805	0.6157	0.1977	14.7960
	Cascade	0.7147	0.7782	0.6813	0.5999	0.1913	15.1433	0.7298	0.7773	0.6886	0.6200	0.2025	13.9598
	AAU-Net	0.6873	0.3308	0.6001	0.5777	0.1321	16.5234	0.7024	0.7528	0.5729	0.5912	0.1823	15.3440
	MobileViT	0.5633	0.4718	0.6654	0.4297	0.0546	55.3913	0.6431	0.6061	0.6610	0.5075	0.0933	29.8677
	nnUNet	0.7229	0.7694	0.7068	0.6084	0.2034	15.5841	0.7369	0.7886	0.7134	0.6394	0.2212	12.9816
	nnFormer	0.7182	0.7539	0.6987	0.5996	0.1966	16.2212	0.7228	0.7702	0.7003	0.6303	0.2107	14.8842
	MedNeXt	0.7301	0.7759	0.7200	0.6152	0.2065	15.5983	0.7479	0.7695	0.7092	0.6415	0.2238	12.4933
	GAE	0.7281	0.8064	0.7019	0.6170	0.2037	14.7981	0.7445	0.8158	0.7056	0.6370	0.2235	12.9057
	DQN	0.6848	0.7558	0.6311	0.5729	0.1814	17.5838	0.7022	0.7613	0.6386	0.5931	0.1952	15.8615
	Double-DQN	0.7049	0.7711	0.6707	0.5982	0.1901	16.2542	0.7176	0.7711	0.6861	0.6080	0.2022	14.7701
	Dueling-DQN	0.7262	0.7847	0.6893	0.6155	0.1984	15.6062	0.7336	0.7882	0.6879	0.6278	0.2086	14.1517
	AC	0.7327	0.7936	0.6951	0.6284	0.2070	14.2097	0.7455	0.8093	0.7045	0.6374	0.2218	12.7538
	A3C	0.7456	0.8173	0.7192	0.6381	0.2114	13.5746	0.7582	0.8109	0.6837	0.6489	0.2321	12.1700
	SDAC	**0.7746**	**0.8306**	**0.7461**	**0.6704**	**0.2372**	**11.4635**	**0.7883**	**0.8402**	**0.7515**	**0.6832**	**0.2461**	**10.7704**
Hecktor	U-Net	0.5723	0.5469	0.6233	0.4621	0.0740	13.1421	0.5854	0.5524	0.6746	0.4763	0.0863	11.4859
	Crop	0.5764	0.5386	0.6397	0.4667	0.0794	12.5231	0.5886	0.5564	0.6791	0.4731	0.0881	10.8830
	AttnUNet	0.5814	0.5654	0.6643	0.4745	0.0928	11.8060	0.5954	0.5845	0.6863	0.4839	0.0951	8.7289
	Cascade	0.5758	0.5473	0.6248	0.4635	0.0743	13.1547	0.5883	0.5635	0.6648	0.4747	0.0868	11.4491
	AAU-Net	0.5846	0.2129	0.5422	0.4756	0.0945	8.5498	0.6208	0.5440	0.6791	0.5117	0.0989	7.5751
	MobileViT	0.4898	0.3313	0.5268	0.3633	0.0409	17.9878	0.5672	0.3828	0.5523	0.4458	0.0618	14.9498
	nnUNet	0.5811	0.5614	0.6571	0.4702	0.0903	11.6420	0.6003	0.5736	0.6698	0.4845	0.0988	9.4682
	nnFormer	0.5776	0.5531	0.6537	0.4563	0.0819	13.0694	0.5894	0.5608	0.6603	0.4736	0.0878	11.6840
	MedNeXt	0.5878	0.5511	0.6543	0.4679	0.0895	12.2098	0.6068	0.5573	0.6718	0.4967	0.0985	8.1733
	GAE	0.5805	0.5553	0.6439	0.4734	0.0835	11.9589	0.5919	0.5677	0.6667	0.4803	0.0866	8.7275
	DQN	0.5761	0.5459	0.6317	0.4663	0.0812	12.6595	0.5890	0.5518	0.6674	0.4778	0.0885	11.3491
	Double-DQN	0.5860	0.5488	0.6379	0.4736	0.0863	12.2191	0.5954	0.5573	0.6712	0.4878	0.0924	10.9024
	Dueling-DQN	0.5949	0.5535	0.6442	0.4843	0.0939	11.5965	0.6080	0.5611	0.6786	0.5009	0.0981	9.6331
	AC	0.6137	0.5680	0.6627	0.5009	0.0995	10.2794	0.6219	0.5753	0.6827	0.5095	0.1045	8.8814
	A3C	0.6239	0.5735	0.6734	0.5118	0.1014	9.9090	0.6310	0.5744	0.7008	0.5200	0.1073	8.2262
	SDAC	**0.6436**	**0.5854**	**0.6993**	**0.5290**	**0.1040**	**7.4621**	**0.6524**	**0.6017**	**0.7159**	**0.5395**	**0.1106**	**6.5747**

Hecktor dataset directly detect the entire image (such as rows 4 and 6). GAE's multi-region bounding boxes cannot locate the segmentation objects well, that is, areas with many boxes do not contain the segmentation objects. In contrast, SDAC can select the optimal bounding boxes centered on the segmentation objects in all examples, effectively avoiding interference from redundant information and maintaining focus on the segmentation objects.

E. Ablation Studies

To demonstrate that the proposed Detector network and PixelSeg are complementary and essential, we conduct ablation experiments and the experimental results are shown in Table VI and Figure 6. In addition, to further prove the universality and optimality of our framework, we also combine SDAC with baselines in extremely small datasets in Table V.

1) Effectiveness of Each Module: By analyzing the experimental results in Table VI, we prove the complementarity and indispensability of each module of SDAC (i.e., PixelSeg and Detector network). Comparing the second row (Detector network and U-Net) with the first row (four-layer U-Net), the third row (PixelSeg only) and the first row, respectively, we find that the last two rows have better segmentation performances than the first row. These findings prove the superiority of PixelSeg compared to the existing deep reinforcement learning and the effectiveness of adding the Detector network. Besides, the segmentation performance of using PixelSeg alone (third row) is better than that of adding the Detector network alone (second row), which also proves the importance of the subjective selection action we proposed. The optimal experimental performance in the fourth row (adding both PixelSeg and Detector network) demonstrates the complementarity of our modules.

As shown in Figure 6, we compare the class activation maps in the last layer of the feature extraction module when the framework has the Detector network (SDAC) and does not have the Detector network (PixelSeg). The results prove that compared with no Detector network, adding the Detector network can provide more accurate activation coverage on the segmentation objects, which means that the framework can better focus on segmenting the object areas and eliminate the

TABLE VI

THE ABLATION EXPERIMENT RESULTS ON TWO PUBLIC DATASETS. **BOLD** REPRESENTS THE BEST RESULT. IN PIXELSEG, ✓ REPRESENTS USING THE PROPOSED PIXEL-LEVEL DEEP REINFORCEMENT LEARNING, AND — MEANS USING TRADITIONAL DEEP LEARNING METHODS FOR TRAINING.

Model		Brain						Hecktor					
PixelSeg	Detector	DICE ↑	PPV ↑	SEN ↑	IoU ↑	BIoU ↑	HD95 ↓	DICE ↑	PPV ↑	SEN ↑	IoU ↑	BIoU ↑	HD95 ↓
–	–	0.7004	0.7461	0.6498	0.5771	0.1823	15.3641	0.5886	0.6486	0.6389	0.4731	0.0872	8.2725
–	✓	0.7232	0.8017	0.6944	0.6141	0.2017	12.1922	0.5991	0.6539	0.6451	0.4855	0.0966	7.5565
✓	–	0.7774	0.7945	0.7527	0.6602	0.2201	10.507	0.6247	0.6224	0.6847	0.5017	0.1019	7.2989
✓	✓	**0.8267**	**0.8851**	**0.7715**	**0.7272**	**0.2662**	**8.6590**	**0.6751**	**0.6827**	**0.7007**	**0.5630**	**0.1232**	**4.7915**

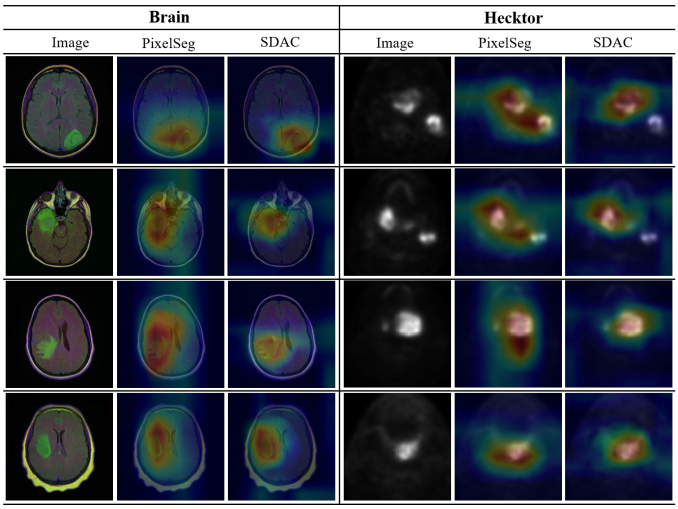
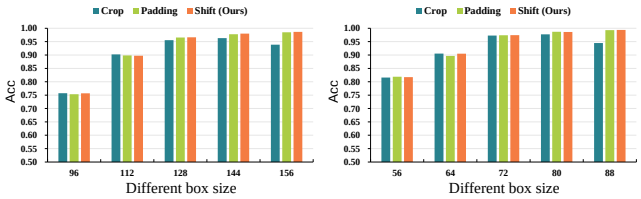


Fig. 6. Compare the class activation maps (CAMs) at the last layer of the feature extraction module when the framework using or not using the Detector network.The CAMs display using the detector network provide the activation of more accurate coverage to the object compared to not using the detector network. In other words, the Detector network can be regarded as an auxiliary task to encourage the model to learn rich features. Best viewed in color.



(a) Brain Dataset (b) Hector Dataset

Fig. 7. Compare the accuracy of different methods to adjust the center point of the bounding box for segmented object edges. Note that it is considered a positive sample only when the segmented object is entirely within the bounding box. If only a part of the segmented object is within the box or if it is outside the box, it is considered a negative sample.

interference of irrelevant information.

2) Effectiveness in Extremely Small Datasets: Observing the experimental results in Table V, we find that SDAC can maintain a relatively ideal segmentation performance even in extremely small datasets. The 50-shot and 100-shot here randomly select 50 and 100 images respectively from the original complete datasets as extremely small datasets. Comparing 50-shot, 100-shot, and the experimental results in Table III, we find that the smaller the amount of data, the more obvious the segmentation performance of each method decreases. However, SDAC can still achieve optimal and ideal segmentation results in extremely small datasets, which proves that our method has good robustness.

F. Impact of Changes in Bounding Box

We compare the impact of Crop, Padding and Shift (our method) operations by adjusting the center point of the bounding box for the edge of the segmentation object, and the impact of setting the size of the bounding box on segmentation performance. The results are shown in Figure 7, where accuracy is taken as an example. Notably, an object is considered a positive sample only when it is entirely within the bounding box. If the object is only partially inside the box or outside it, it is considered a negative sample. First, comparing the accuracy of the three operations, we find that in the Crop operation, setting the box size too large reduces the visible range and leads to a significant decrease in accuracy; while Shift operation can achieve the best results in almost all box sizes. Furthermore, the lack of performance degradation with the Padding operation also suggests the existence of redundancy in medical images, and this redundancy is useless for segmentation targets. Then, we set the box size to one-quarter of the original image in our work, that is, 128×128 in the Brain dataset and 72×72 in the Hecktor dataset. We find that initially as the box size increased, the accuracy also improved significantly; however, when the sizes exceed 128 and 72 in two datasets respectively, the accuracy does not improve significantly or even decreases, which also results in greater computational costs. These findings justify the reasonableness of our box size settings.

G. Effects on Step-Varying

We further study the effect of the thread step size (t in Algorithm 1) on the segmentation performance of SDAC, and the results are shown in Figure 8 with DICE as an example. We observe that the segmentation performance is significantly improved when the thread step size increases from 1 to 10; the increase after 10 is not obvious. It shows that an appropriate increase in the thread step size can improve the model's segmentation performance, but too much thread step size may not only increase the segmentation performance but also cause a waste of computing resources; it also proves the rationality of choosing an iteration step size of 10 in our work.

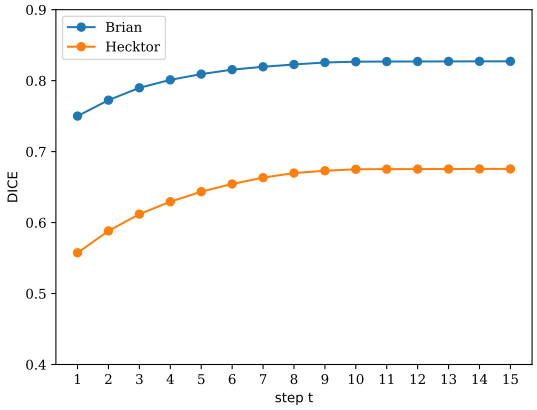


Fig. 8. Compare the Segmentation performance of the proposed SDAC on step-varying.

H. Additional Experiments

Efficiency Analysis. We compared SDAC with other major segmentation methods using four metrics: parameters, inference speed, floating point operations per second (FLOPs), and GPU memory footprint. All measurements were conducted using the same analysis script on a machine equipped with an NVIDIA 3090 GPU. To ensure fairness, we used the same key hyperparameters across all methods (e.g., batch size set to 1, input size to 256×256). As shown in Table VII, we can observe that although SDAC requires longer inference time due to deep reinforcement learning and its multi-step iterative mechanism, it remains within an acceptable range. SDAC demonstrates superior segmentation performance with fewer parameters and lower computational cost, indicating a favorable efficiency-performance trade-off. In future work, we will focus on further reducing training costs and enhancing overall efficiency.

TABLE VII
COMPARISON OF DIFFERENT METHODS BASED ON PARAMETERS, INFERENCE TIME (SECONDS PER PICTURE), FLOPS, AND GPU MEMORY USAGE.

Methods	Brain			
	Param	Inter-Time	FLOPs	Memory
U-Net	31.03M	0.058s	83.70G	3.04G
AAU-Net	104.62M	0.977s	1402.98G	13.51G
nnUNet	31.20M	0.058s	84.21G	3.05G
nnFormer	28.81M	0.072s	102.60G	4.12G
MedNeXt	10.48M	0.017s	22.55G	2.45G
A3C	10.82M	0.163s	17.73G	4.34G
SDAC	1.10M	0.184s	10.45G	7.96G

Effect of the Motion Artifacts. We conducted additional experiments to evaluate the robustness of our method against motion artifacts. Due to constraints such as medical ethics, it is extremely difficult to obtain real motion artifact images paired with artifact-free ground truth. To address this, we referred to deep learning-based artifact removal studies, which typically use simulated MRI motion artifacts to construct paired datasets for network training. Following these approaches, we constructed a dataset using simulated MRI motion artifacts on the Brain dataset for both training and testing. We adopted the method proposed by Pawar et al. [43] to generate highly realistic simulated motion artifacts, which applies rotational motion within a range of $[-5°, +5°]$ along two axes, as well as translational motion within $[-5mm, +5mm]$ on two planes. The results are shown in Table VIII, our method maintains strong segmentation performance even on artifact-corrupted data and outperforms other baselines. This can be attributed to the ability of our detection network to help the subsequent segmentation stage avoid interference from motion artifacts as much as possible. These experimental results demonstrate the robustness of SDAC when facing motion artifact disturbances.

TABLE VIII
RESULTS OF APPLYING THE PROPOSED METHOD AND THE STATE-OF-THE-ART BASELINES ON THE BRAIN DATASET SIMULATED WITH MRI MOTION ARTIFACTS, WHERE THE BEST RESULTS ARE IN BOLD, THE SECOND BEST METHODS ARE UNDERLINED.

Methods	Brain					
	DSC	PPV	Sen	IoU	BIoU	HD95
U-Net	0.7084	0.7376	0.6679	0.5875	0.1837	13.9798
AAU-Net	0.7631	0.8433	0.6901	0.6568	0.2329	10.8663
nnUNet	0.7819	0.8513	0.7315	0.6809	0.2396	11.8849
nnFormer	0.7749	0.8429	0.7282	0.6757	0.2363	12.0485
MedNeXt	<u>0.8035</u>	<u>0.8733</u>	<u>0.7658</u>	<u>0.7090</u>	0.2411	<u>9.9806</u>
A3C	0.7732	0.8315	0.7114	0.6766	<u>0.2421</u>	10.8464
SDAC	**0.8213**	**0.8811**	**0.7692**	**0.7190**	**0.2608**	**9.1560**

V. DISCUSSION

We now briefly summarize the social impact of our approach, as well as the limitations of our work and future works.

A. Social Impact of Proposed Approach

The goal of medical image segmentation is to accurately delineate the contours of organs and lesions, which is crucial for subsequent diagnosis and clinical procedures. However, obtaining precise segmentation masks in practice is a labor-intensive and costly task. As a result, many existing approaches leverage deep learning or deep reinforcement learning to improve the quality of medical image segmentation. Nevertheless, most of these methods originate from the natural image domain, making them prone to overlooking the inherent differences between natural and medical images. While some state-of-the-art techniques can achieve modest improvements when directly applied to medical images, their performance remains limited.

To address this issue, we propose a novel "Position-then-analysis" strategy inspired by the real world diagnostic process used by physicians. We design SDAC to model this process. Our method employs a detector network combined with policy gradient algorithms to scan the entire image and filter out irrelevant background areas. The policy and value networks then focus on smaller, critical regions for refined segmentation. Moreover, our framework is flexible and transferable to other

domains. Its backbone can be replaced with other feature extraction models, and additional improved modules can be easily integrated.

Beyond its technical contributions, this work also brings significant societal benefits to related research and clinical applications. For example, the relatively low training cost reduces deployment difficulty, accelerates the adoption of intelligent computer aided diagnosis systems, and significantly reduces the workload of physicians by helping to save both time and human resources.

B. Limitations and Future Work

In our work, the qualitative analysis only presented results on the axial for the following reasons. First, the axial view is the most commonly used in clinical practice and typically provides the clearest depiction of the anatomical structures of interest in our dataset. Second, the axial plane is also the most frequently used for visual evaluation in other segmentation models. Additionally, the Brain dataset used in our experiments is a publicly available dataset that provides only axial plane images and ground truth annotations. In future work, we plan to address this limitation by extending our method to 3D medical images to enable multi-plane evaluation.

Furthermore, when multiple tumor regions are present, our detection box centers are computed based on the centroid of the segmented area. As a result, the center of the detection box tends to fall near the center of multiple tumor regions, allowing it to effectively cover them. However, this approach has certain limitations. When the tumor regions are far apart, the detection box may only contain part of the tumor regions. This issue can be partially addressed by manually increasing the size of the detection box for such datasets. However, this solution relies on human experience and fails to handle extreme cases (such as when tumor regions are located in the top-left and bottom-right corners of the image). A more intuitive solution to overcome this limitation is to introduce multiple detection boxes within the detector network, but doing so would significantly affect the subsequent segmentation stage. The first is to pass each detection box individually through the subsequent pixel-level segmentation (pixelSeg) stage. However, this not only prevents the policy and value networks from being independently optimized for each detection box, but also significantly increases both training and inference time, which hinders practical deployment. The second solution is to concatenate multiple detection regions into a single image by padding with zero pixels. Yet this approach introduces uncertainty in the final image dimensions and, similar to the padding operation mentioned when adjusting the detection box center, adds redundant information, which is contrary to our goal of eliminating medically irrelevant redundancy. Therefore, a more feasible solution we plan to explore is modifying the task of the detector network into a masking task. For instance, the input image can be divided into 64×64 image patches, and the probing network would determine whether each patch contains a segmentation target. Patches that do not contain targets are masked out. This approach eliminates the need for manually defined fixed-size detection boxes and enables identification of multiple segmentation targets without introducing unnecessary redundancy. Hence, this is an interesting research direction we plan to explore in the future.

VI. Conclusion

In this work, we first identified the limitations of existing medical image segmentation methods due to the higher redundancy in medical images compared to natural images. To address this issue, we have proposed a novel Stochastic Detector-Actor-Critic framework (SDAC) that effectively segments medical images by mimicking the diagnostic process of doctors. Unlike traditional methods that process the entire image, SDAC incorporates a Detector network optimized with policy gradient algorithms, which identifies smaller segmentation areas by removing irrelevant background regions. This enables the model to focus more on the segmentation objects. Subsequently, it utilizes the Actor-Critic algorithm to dynamically generate mask pixel by pixel for these smaller segmentation areas, selecting appropriate actions for each pixel to determine whether it belongs to the foreground or background. We conducted extensive experiments on the Brain and Hecktor datasets, and the results demonstrate SDAC's superior segmentation performance over existing methods, using fewer parameters. The framework also exhibits robust performance even with extremely small datasets.

[1] S. Zhang, J. Zhang, B. Tian, T. Lukasiewicz, and Z. Xu, "Multi-modal contrastive mutual learning and pseudo-label re-learning for semi-supervised medical image segmentation," *Medical Image Analysis*, vol. 83, p. 102656, 2023.

[2] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proceedings of the International Conference on Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2018, pp. 3–11.

[3] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, "UNet 3+: A full-scale connected U-Net for medical image segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 1055–1059.

[4] M. Tang, Z. Zhang, D. Cobzas, M. Jagersand, and J. L. Jaremko, "Segmentation-by-detection: A cascade network for volumetric medical image segmentation," in *Proceedings of 2018 IEEE 15th international symposium on biomedical imaging*, 2018, pp. 1356–1359.

[5] D. Zhao, Y. Chen, and L. Lv, "Deep reinforcement learning with visual attention for vehicle classification," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 4, pp. 356–367, 2016.

[6] S. Seifi, A. Jha, and T. Tuytelaars, "Glimpse-attend-and-explore: Self-attention for active visual exploration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 137–16 146.

[7] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention U-Net: Learning where to look for the pancreas," *ArXiv Preprint, ArXiv:1804.03999*, 2018.

[8] G. Chen, L. Li, Y. Dai, J. Zhang, and M. H. Yap, "Aau-net: an adaptive attention u-net for breast lesions segmentation in ultrasound images," *IEEE Transactions on Medical Imaging*, 2022.

[9] I. Momennejad, E. M. Russek, J. H. Cheong, M. M. Botvinick, N. D. Daw, and S. J. Gershman, "The successor representation in human reinforcement learning," *Nature human behaviour*, vol. 1, no. 9, pp. 680–692, 2017.

[10] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, 2000, pp. 1008–1014.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[12] D. Yuan, Z. Xu, B. Tian, H. Wang, Y. Zhan, and T. Lukasiewicz, "μ-net: Medical image segmentation using efficient and effective deep supervision," *Computers in Biology and Medicine*, vol. 160, p. 106963, 2023.

[13] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers make strong encoders for medical image segmentation," *ArXiv Preprint, ArXiv:2102.04306*, 2021.

[14] B. Xu, J. Liu, X. Hou, B. Liu, J. Garibaldi, I. O. Ellis, A. Green, L. Shen, and G. Qiu, "Attention by selection: A deep selective attention approach to breast cancer classification," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 1930–1941, 2019.

[15] M. Benčević, Y. Qiu, I. Galić, and A. Pižurica, "Segment-then-segment: Context-preserving crop-based segmentation for large biomedical images," *Sensors*, vol. 23, no. 2, p. 633, 2023.

[16] Y. Song, J. Wang, T. Lukasiewicz, Z. Xu, S. Zhang, A. Wojcicki, and M. Xu, "Mega-reward: Achieving human-level play without extrinsic rewards," in *Proceedings of AAAI*, vol. 34, no. 04, 2020, pp. 5826–5833.

[17] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *Advances in neural information processing systems*, vol. 33, pp. 19 884–19 895, 2020.

[18] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Applied Intelligence*, vol. 50, no. 8, pp. 2488–2502, 2020.

[19] D. Yuan, Y. Liu, Z. Xu, Y. Zhan, J. Chen, and T. Lukasiewicz, "Painless and accurate medical image analysis using deep reinforcement learning with task-oriented homogenized automatic pre-processing," *Computers in Biology and Medicine*, vol. 153, p. 106487, 2023.

[20] K. Cheng, C. Iriondo, F. Calivá, J. Krogue, S. Majumdar, and V. Pedoia, "Adversarial policy gradient for deep learning image augmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part VI 22*. Springer, 2019, pp. 450–458.

[21] G. Xu, S. Wang, T. Lukasiewicz, and Z. Xu, "Adaptive-masking policy with deep reinforcement learning for self-supervised medical image segmentation," in *Proceedings of ICME*, 2023.

[22] Z. Xu, Y. Liu, G. Xu, and T. Lukasiewicz, "Self-supervised medical image segmentation using deep reinforced adaptive masking," *IEEE Transactions on Medical Imaging*, vol. 44, no. 1, pp. 180–193, 2025.

[23] Y. Song, A. Wojcicki, T. Lukasiewicz, J. Wang, A. Aryan, Z. Xu, M. Xu, Z. Ding, and L. Wu, "Arena: A general evaluation platform and building toolkit for multi-agent intelligence," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7253–7260.

[24] X. Liao, W. Li, Q. Xu, X. Wang, B. Jin, X. Zhang, Y. Wang, and Y. Zhang, "Iteratively-refined interactive 3D medical image segmentation with multi-agent reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9394–9402.

[25] Z. Tian, X. Si, Y. Zheng, Z. Chen, and X. Li, "Multi-step medical image segmentation based on reinforcement learning," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2020.

[26] R. Furuta, N. Inoue, and T. Yamasaki, "PixelRL: Fully convolutional network with reinforcement learning for image processing," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1704–1719, 2019.

[27] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the International Conference on Machine Learning*, 2016, pp. 1928–1937.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[29] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, pp. 229–256, 1992.

[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[31] M. Buda, A. Saha, and M. A. Mazurowski, "Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm," *Computers in Biology and Medicine*, vol. 109, pp. 218–225, 2019.

[32] V. Andrearczyk, V. Oreiller, S. Boughdad, C. C. L. Rest, H. Elhalawani, M. Jreige, J. O. Prior, M. Vallières, D. Visvikis, M. Hatt *et al.*, "Overview of the hecktor challenge at miccai 2021: automatic head and neck tumor segmentation and outcome prediction in pet/ct images," in *Proceedings of the 3D head and neck tumor segmentation in PET/CT challenge*, 2021, pp. 1–37.

[33] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary IoU: Improving object-centric image segmentation evaluation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 334–15 342.

[34] D. Karimi and S. E. Salcudean, "Reducing the hausdorff distance in medical image segmentation with convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 499–513, 2019.

[35] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proceedings of ICLR*, 2022.

[36] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021.

[37] H.-Y. Zhou, J. Guo, Y. Zhang, X. Han, L. Yu, L. Wang, and Y. Yu, "nnformer: volumetric medical image segmentation via a 3d transformer," *IEEE transactions on image processing*, vol. 32, pp. 4036–4045, 2023.

[38] S. Roy, G. Koehler, C. Ulrich, M. Baumgartner, J. Petersen, F. Isensee, P. F. Jaeger, and K. H. Maier-Hein, "Mednext: transformer-driven scaling of convnets for medical image segmentation," in *Proceedings of MICCAI*, 2023, pp. 405–415.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[40] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[41] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2016, pp. 1995–2003.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations*, April 2014, pp. 1–15.

[43] K. Pawar, Z. Chen, N. J. Shah, and G. F. Egan, "Suppressing motion artefacts in mri using an inception-resnet network with motion simulation augmentation," *NMR in Biomedicine*, vol. 35, no. 4, p. e4225, 2022.