

Figure 1.The framework of our proposed method xxx consist of four main components : (a) The raw ultrasound video is initially departed into several clips. (b) The Multi-exports Mask Generation Network conduct the calculation of the token score (c) The encoder is trained by reconstructing the masked the cubic tokens according to the token score with decoder. (d) The final diagnostic result of a prostate ultrasound imaging clip is calculated using the token score and features extracted by the encoder through the classification head.

**3.Methodology**

In this section,we first present the motivation and overall framework of the proposed method xxx in Section 3.1. Next , we turn into the details of the proposed Multi-exports Mask Generation Network (MeMGN) in the pre-training phase and the additional prompt based on MeMGN in the downstream classification fine-tuning phase in Section 3.2 and 3.3.

**3.1 Overview**

**3.1.1 Motivation** (感觉motivation可以从我们使用的是预训练模型出发再写一个分支？因为后面不可避免的要提到FocusMAE，他难就难在是自己标注数据训练的分割模型)

Given a prostate ultrasound video contains frames , we aim to address the following two issues:

1. ***Insufficient key region representation pre-training leads to details missing****：*The mainstream masking methods lack the concentration on the lesion areas of prostate ultrasound images, which may influence the feature extraction in these key regions crucial for downstream tasks.
2. ***Direct applying general masked features leads to insufficient classification***: Most downstream classification methods using masked features directly on the pre-training encoder, which may cause the model to overlook generalizable features, weakening robustness and cross-domain performance.

To address these issues , we propose xxx that mask the tokens by MeMGN, guided by multiple pre-trained medical expert models and give additional prompt based on MeMGN in the downstream classification fine-tuning. This enables us to make the training of our encoder **representation extraction focus on key regions** , while paying more attention to **general features** during classification fine-tuning.

**3.1.2 Framework**

In this section , we will briefly introduce the overall framework of the proposed xxx in Fig.1. Given a prostate ultrasound video , which contains represents the frame, height, and width of the video. The objective of XXX is to focus on key regions in prostate ultrasound images to achieve highly accurate classification. The framework is divided into two stages: pretraining and fine-tuning. Here's a refined and academically appropriate version of your paragraph, with improved clarity and structure:

The objective of the pre-training phase in our XXX framework is to train the encoder to learn meaningful feature representations from video ultrasound images. The input video is divided into non-overlapping 3D cubic tokens of size , where t, h, and w represent the temporal and spatial dimensions of each token. Then, the token is projected into a feature vector of dimension d (set to 768 in our design). During the fine-tuning phase, the encoder extracts features from the input video are fed into a classification head to predict the final label of the video.

**3.2 Multi-exports guided Pre-training**

(这边按照总分总进行修改，先说明预训练部分参见图几，然后以此说明每个模块的作用，给出公式和符号，公式符号是描述的时候复用的。然后后面专门说loss function, 最好是看图写话，按照流程从输入到输出一步一步写明白。)

3.2.1 Pre-training phase

Specifically,for the redundant information in video data , we use a sampling stride of 4 to alleviate data redundancy. Suppose a video contains frames , we sample a total of frames. Then , for these T/4 frames, we further divide them into clips, with each clip containing 16 frames. Each frame has a size of , where H is the height of the image and W is the width of the image. Before entering the subsequent process , the frame size will be adjusted to .As a result , we will get the clip with the size of .

We first divide the clip into tokens of size .To achieve this , we use 3D convolution of kernel size is , stride , and d out-put channels.So that we can get the embeded tokens as follows:

(1)

Where represents the token number.

After the patch embedding , the embeded tokens and the clip are given to the MeMGN , outputing the mask scores of these tokens as follows :

(2)

Then, we sort the mask scores and mask the tokens with the top-k scores. The k is defined by mask ratio .Then we input the remaining visible tokens that have not been masked into the encoder, which is based on the Vision Transformer architecture and get the encoded visible tokens . The overall process of encode can be represented as follows:

(3)

Finally, the encoded visible tokens is appended with masked tokens to get the combined tokens . Then , is fed into the decoder of the Vision Transformer to obtain the reconstruction result of the token .

(4)

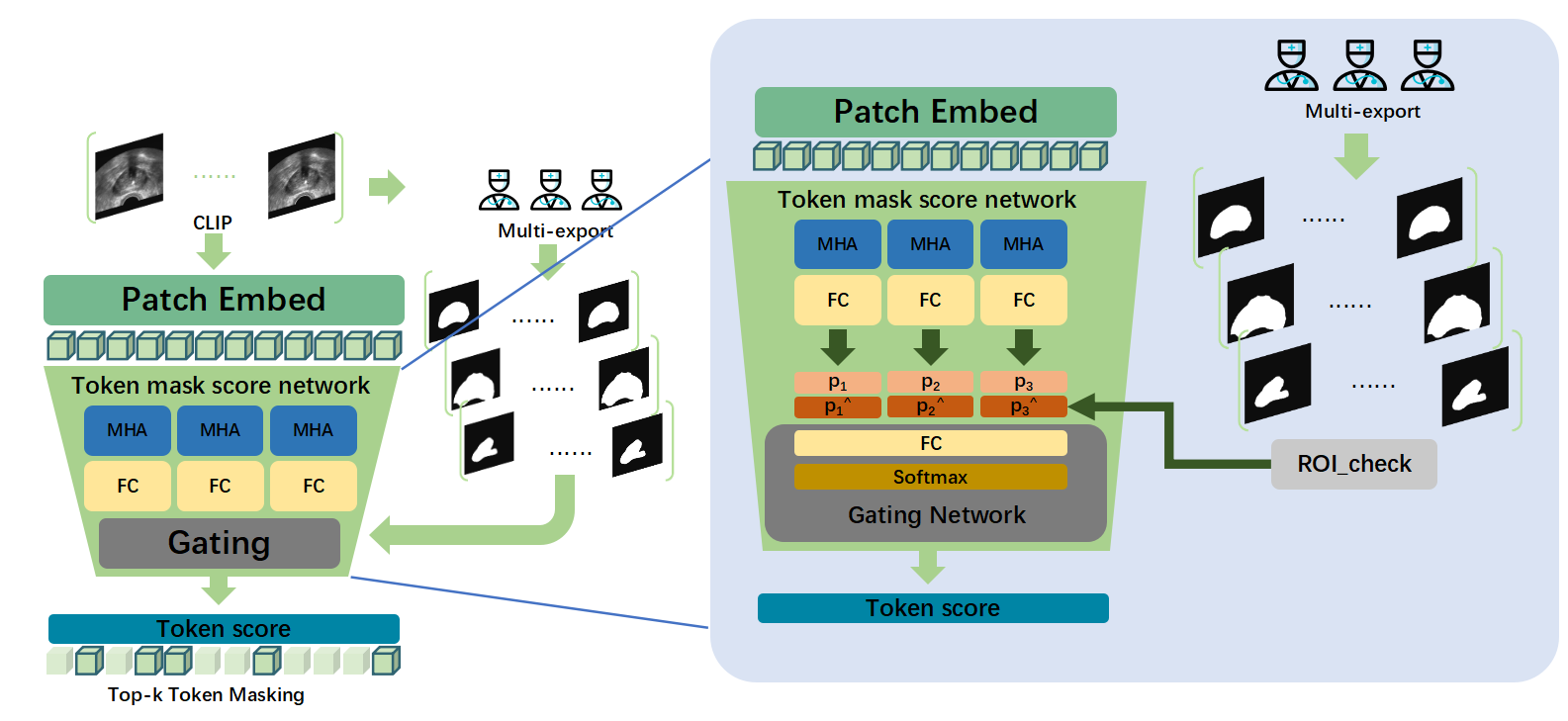


Figure 2.Detailed structure of the proposed MeMGN , including four steps : (a) The input cubic tokens perform Multi-head attention (MHA) operations at different tokens to obtain the basic token scores.(b) Raw images of input clip get segments from pretrained export models.(c) Some of the basic token scores are increased due to the segments provided by export models.(d) Three different exports’ token scores are fused by Gating Network to obtain the final output token scores.

3.2.2 Multi-exports Mask Generation Network

As illustrated in Fig. 2, we introduce a multi-exports mask generation network (MeMGN) including three different token mask score network and their export model, effectively capture the importance of the lesion area and the relationship between lesion area and other regions. Specifically , given the input cubic tokens

, the MeMGN initially Make three identical copies of the cubic tokens for multi-exports network :

(5)

Where represents the copy operation and is the tokens for the i-th export , whereranges form 1 to 3. Inspired by the Masking probability network of FocusMAE , we construct a muti-exports guided mask generation network that combines the priori knowledge from pretrained models instead of training a segment model with large-scale annotated images. For tokens , we use the i-th token mask score network to conduct the basic token score by performing multi-head attention , followed by fully connected layer. ​The network aggregates contextual information from the video and determines the importance between tokens, thereby guiding the subsequent masking process and compelling the model to learn representations of key high-scoring regions.It can be formulated as:

(6)

Where represents a series of multi-head attention blocks.Note that the multi-head attention’s output is the same shape as input .Then the output is fed into the fully connected layer to calculate the basic token score:

(7)

Where denotes the fully connected layer. Outcomes .Afterward , the raw images of the clip are collected to generate the priori knowledge by pretrained export models. Given the raw images , each export model give out The binarized result represents the segmentation result. To meet the token scores’ format of the cubic tokens, we perform patch processing on the binarized segmentation results. The overall steps of the patch processing can be described as :

(8)

(9)

Where denotes The set of elements at corresponding positions in the binarized segmentation result, checks whether all elements in a block contain 1, which corresponds to the segmented area. If so, it outputs 1, effectively performing an erosion operation on the segmentation region. .Then, we flatten all the values to obtain the final a priori knowledge proposal .

Region priors then boost the probability score as follows:

(10)

Then if the i-th token spatially lies within the candidate regions, then we inflate the score of the token by , where is a small tensor less than 2. Inspired by the multi-expert mechanism and the multi-expert consultations in medicine, we choose to fuse the scores of three export network through a learnable gating mechanism using a fully connected layer to calculate the weight for each export network rather than using a fixed mechanism such as averaging :

(10)

Then, perform pixel-wise weighted averaging to obtain the final score.

3.2.3 Loss Function

In order to training the encoder to obtain feature representation of video images better. We design the reconstruction loss and sampling loss to optimize the encoder-decoder and MMGN.

**Masking Reconstruction Loss**.We have used the Mean Squared Error loss (MSE) between the predicted and ground-truth RGB values of the masked tokens as the objective function to pretrain the MAE. The loss function is given as:

(4)

Here and denote the predicted token and the ground-truth RGB values of the token.represent the number of masked tokens.

**Smoothness Loss** essentially solves the blockiness problem (also known as the "blocking effect"), where reconstructed images may show unnatural grid-like patterns due to overfitting or discontinuities. By applying smoothness loss, the algorithm encourages the pixels in the image to vary more gradually, leading to more natural-looking, continuous structures.

The vertical (diff\_h) and horizontal (diff\_w) smoothness losses each measure the differences between adjacent pixels in their respective directions. When both losses are minimized, the result is a smoother image with fewer abrupt changes between neighboring pixels, which significantly reduces blocky artifacts.The loss formulation is as follows:

(5)

(6)

Here I represents each image of the reconstructed cilp .The final is compute as follows ：

(7)

**Token Sampling Loss**. We use a token sampling loss, Lsample, to train the sampling network that generates the sampling probability. We adapt the sampling loss proposed by AdaMAE and use maximization of the average reconstruction error to define the loss. The formulation of such a formulation is motivated by the expected reward maximization of the REINFORCE algorithm in RL. Here, the visible token sampling process is the action, the MAE is the environment, and the masked token reconstruction error is the return. The reconstruction error is high in the high information regions as compared to the low information background regions. Thus, maximizing the expected reconstruction error would result in the network predicting a higher probability score for a high information region. The loss formulation is as follows:

(8)

Here denote the Probability of the token to be masked.

**3.3 Reverse attention Fine-tuning**

3.3.1 这一部分还在画图，哥可以先不用看

After the pre-training phase, we freeze the MeMGN and encoder that were optimized during pre-training and transfer them to the fine-tuning phase.Given a labeled video with ground-truth label , first we will do the exactly same process as the pre-training phase to get the clip with the size of and token scores .And then we input all the embedded tokens into pretrained vision transformer encoder , The token scores are multiplied by a negative sign and then weighted and multiplied with the deep features from the encoder's output , ultimately leading to the classification result through the classification head , which is designed as a fully connected layer. .The overall process can be represented as follows:

(9)

(10)

(11)

In the fine-tuning task, we use cross-entropy to update the parameters of the classification head.

3.3.2 Loss Function

**4.Datasets**