

MSVM-UNet: Multi-Scale Vision Mamba UNet for Medical Image Segmentation

Chaowei Chen

*School of Information Science and Engineering
Yunnan University
Kunming, China
chishengchen@stu.ynu.edu.cn*

Shiquan Min

*School of Information Science and Engineering
Yunnan University
Kunming, China
minshiquan@mail.ynu.edu.cn*

Li Yu

*School of Information Science and Engineering
Yunnan University
Kunming, China
yuli0501@163.com*

Shunfang Wang*

*School of Information Science and Engineering
Yunnan University
Kunming, China
sfwang_66@ynu.edu.cn*

Abstract—State Space Models (SSMs), especially Mamba, have shown great promise in medical image segmentation due to their ability to model long-range dependencies with linear computational complexity. However, accurate medical image segmentation requires the effective learning of both multi-scale detailed feature representations and global contextual dependencies. Although existing works have attempted to address this issue by integrating CNNs and SSMs to leverage their respective strengths, they have not designed specialized modules to effectively capture multi-scale feature representations, nor have they adequately addressed the directional sensitivity problem when applying Mamba to 2D image data. To overcome these limitations, we propose a Multi-Scale Vision Mamba UNet model for medical image segmentation, termed MSVM-UNet. Specifically, by introducing multi-scale convolutions in the VSS blocks, we can more effectively capture and aggregate multi-scale feature representations from the hierarchical features of the VMamba encoder and better handle 2D visual data. Additionally, the large kernel patch expanding (LKPE) layers achieve more efficient upsampling of feature maps by simultaneously integrating spatial and channel information. Extensive experiments on the Synapse and ACDC datasets demonstrate that our approach is more effective than some state-of-the-art methods in capturing and aggregating multi-scale feature representations and modeling long-range dependencies between pixels. Our implementation is available at <https://github.com/gndlwch2w/msvm-unet>.

Index Terms—Medical Image Segmentation, Vision State Space Models, Multi-Scale Feature Learning.

I. INTRODUCTION

Precise and efficient medical image segmentation is one of the fundamental yet challenging tasks in medical image analysis. Research in this area leverages techniques such as deep learning to analyze various types of medical images and produce segmentation maps for specific organs or pathological regions, thereby aiding physicians and researchers in disease analysis and diagnosis.

In recent years, medical image segmentation with Convolutional Neural Networks (CNNs) and Vision Transformers

(ViTs) has seen notable success. Specifically, UNet [1], with its elegant U-shaped structure and skip connections, excels in processing high-resolution medical images and seamlessly combines low-level details with high-level semantics for impressive segmentation results. Furthermore, TransUNet [2] proposes a hybrid structure of CNNs and ViTs to simultaneously utilize the detail extraction capabilities of CNNs and the long-range dependency modeling capabilities of ViTs. Although these methods achieve commendable performance and produce high-quality segmentation results, inherent characteristics of CNNs and transformers present performance bottlenecks [2], [3]. Specifically, CNNs rely on local convolutional kernels for feature extraction, which, while effective for capturing local feature patterns, limits their ability to model global and geometric features [4]. Although transformer-based methods perform well in modeling long-range dependencies, self-attention mechanisms have quadratic computational complexity with respect to sequence length [5], which makes it challenging to handle high-resolution segmentation tasks efficiently. Moreover, methods such as Swin Transformer [6], PVT v2 [7], and BiFormer [8] propose effective self-attention computation techniques. However, these methods come with a trade-off between computational complexity and modeling capability, thereby limiting their ability to model long sequences.

Recently, State Space Models (SSMs) [9]–[11] have garnered widespread research interest due to their immense potential in modeling long sequences. Mamba [11] is proficient in modeling long-sequence dependencies with linear computational complexity and has demonstrated significant success in the field of natural language processing. Building on this, VMamba [12] introduces a Cross-Scan Module (CSM) and a well-designed hierarchical architecture design, indicating its significant potential in analyzing 2D image data. In the field of medical image segmentation, efficiently processing high-resolution medical images remains a significant challenge. Inspired by the above work, U-Mamba [13] proposes embed-

* corresponding author.

ding convolutional operations within SSMs to integrate the local feature extraction power of convolutional layers with the long-range dependency capture capabilities of SSMs. Swin-UMamba [14] demonstrates that transferring VMamba pre-trained on ImageNet-1k to the medical image segmentation domain can effectively address issues such as limited data resources. Similar to Swin-UNet [15], VM-UNet [16] proposes using pure Visual State Space (VSS) blocks to construct a medical image segmentation framework.

Unlike 1D sequences, pixels in 2D visual data inherently possess directional dependencies [17]. Directly applying the 1D sequence processing method from Mamba to 2D data fails to effectively capture long-range dependencies between pixels and results in limited receptive fields, due to the lack of consideration for the spatial characteristics of 2D data, which is known as the directional sensitivity problem [12]. Although VMamba employs four scanning strategies to address this issue, it only focuses on four of the eight neighboring directions (i.e., up, down, left, right), leading to certain limitations in modeling dependencies between pixels due to the incomplete consideration of dependencies in all directions. Additionally, although U-Mamba and Swin-UMamba incorporate a hybrid architecture of CNNs and SSMs, they do not specifically address multi-scale feature learning, which results in shortcomings when analyzing objects of varying sizes and shapes [18], [19]. To solve these issues, we propose a Multi-Scale Visual State Space (MSVSS) block, which uses a set of parallel convolution operations with different kernel sizes to capture and aggregate multi-scale feature representations, and not only models dependencies in the original four directions, but also uses convolution operations to aggregate the information in the remaining four diagonal directions.

Moreover, the patch expanding layer is used for feature upsampling in both Swin-UNet and VM-Net. However, since the patch expanding layer only considers channel-wise information and does not account for spatial relationships during upsampling, it leads to insufficient discriminative power. To overcome this issue, we propose a Large Kernel Patch Expanding (LKPE) layer for upsampling, which aggregates features along the channel dimension while also integrating spatial information through a 3×3 depth-wise convolution.

The main contributions of this study are summarized as follows:

- We propose a novel Multi-Scale Visual State Space (MSVSS) block, which combines CSM with multi-scale convolution operations to not only effectively model long-range dependencies between pixels but also capture multi-scale feature representations.
- We introduce a new Large Kernel Patch Expanding (LKPE) layer for feature map upsampling. By incorporating a large kernel depth-wise convolution before expanding the channel dimensions, we achieve more discriminative feature representations with acceptable additional overhead.
- We validate our proposed MSVM-Net on the Synapse multi-organ dataset and the ACDC dataset. Specifically,

on the Synapse multi-organ dataset, our model achieved a DSC of 85.00% and an HD95 of 14.75mm. On the ACDC dataset, our model achieved a DSC of 92.58%.

II. METHODS

A. Overall Architecture of MSVM-UNet

The overall architecture of the proposed MSVM-UNet is illustrated in Fig. 1, which adopts a U-shaped hierarchical encoder-decoder structure with skip connections. The encoder is VMamba V2 [12] pre-trained on the ImageNet-1k dataset, which contains 4 stages. Except for the first stage consisting of a patch embedding layer and a VSS block, the other three stages are composed of a patch merging layer and a VSS block. Specifically, the patch embedding layer divides the input $X \in \mathbb{R}^{H \times W \times 3}$ into non-overlapping patches of size 4×4 and maps the channel dimension to dimension C . The VSS block is responsible for learning hierarchical feature representations of the input image. The patch merging layer is used for down-sampling the feature maps. For the input X , we first use the four stages of the encoder to sequentially extract four sets of hierarchical feature representations, denoted as f_1^e , f_2^e , f_3^e , and f_4^e , which are then fed into the decoder. Specifically, the feature f_4^e is passed through the expanding path to the last stage of the decoder, while the features f_1^e , f_2^e , and f_3^e are sent to the corresponding stages of the decoder via skip connections. The decoder consists of three stages, each comprising a Large Kernel Patch Expanding (LKPE) layer and a Multi-Scale Vision State Space (MSVSS) block. Unlike the patch merging layer, the LKPE layer is responsible for up-sampling the feature maps. The MSVSS block captures and aggregates fine-grained multi-scale information from the contracting path and high-level semantic information from the expanding path. Finally, the segmentation prediction is obtained through the final large kernel patch expanding (FLKPE) layer.

B. Multi-Scale Vision State Space (MSVSS) Block

To simultaneously capture multi-scale detail information in hierarchical features and effectively address the direction sensitivity issue in 2D visual data, we propose the Multi-Scale Visual State Space (MSVSS) block. Specifically, MSVSS addresses these challenges by introducing a Multi-Scale Feed-Forward Network (MS-FFN) within the VSS block. Firstly, the 2D-Selective-Scan Block (SS2DBlock) models the long-range dependencies of each feature in four directions, then the convolution operations in the MS-FFN aggregate information from the four remaining diagonal directions to enhance feature representation. Additionally, to effectively capture and aggregate multi-scale feature representations, the MSVSS employs a set of parallel convolution operations with different kernel sizes to achieve this goal. As shown in Fig. 1(c), the MSVSS block includes two layer normalization layers, an SS2DBlock, and an MS-FFN. The definition of the MSVSS block is given by Equations (1) and (2):

$$\hat{f}_i^d = SS2DBlock(LN(f_i^d)) + f_i^d \quad (1)$$

$$f_{i+1}^d = MS-FFN(LN(\hat{f}_i^d)) + \hat{f}_i^d \quad (2)$$

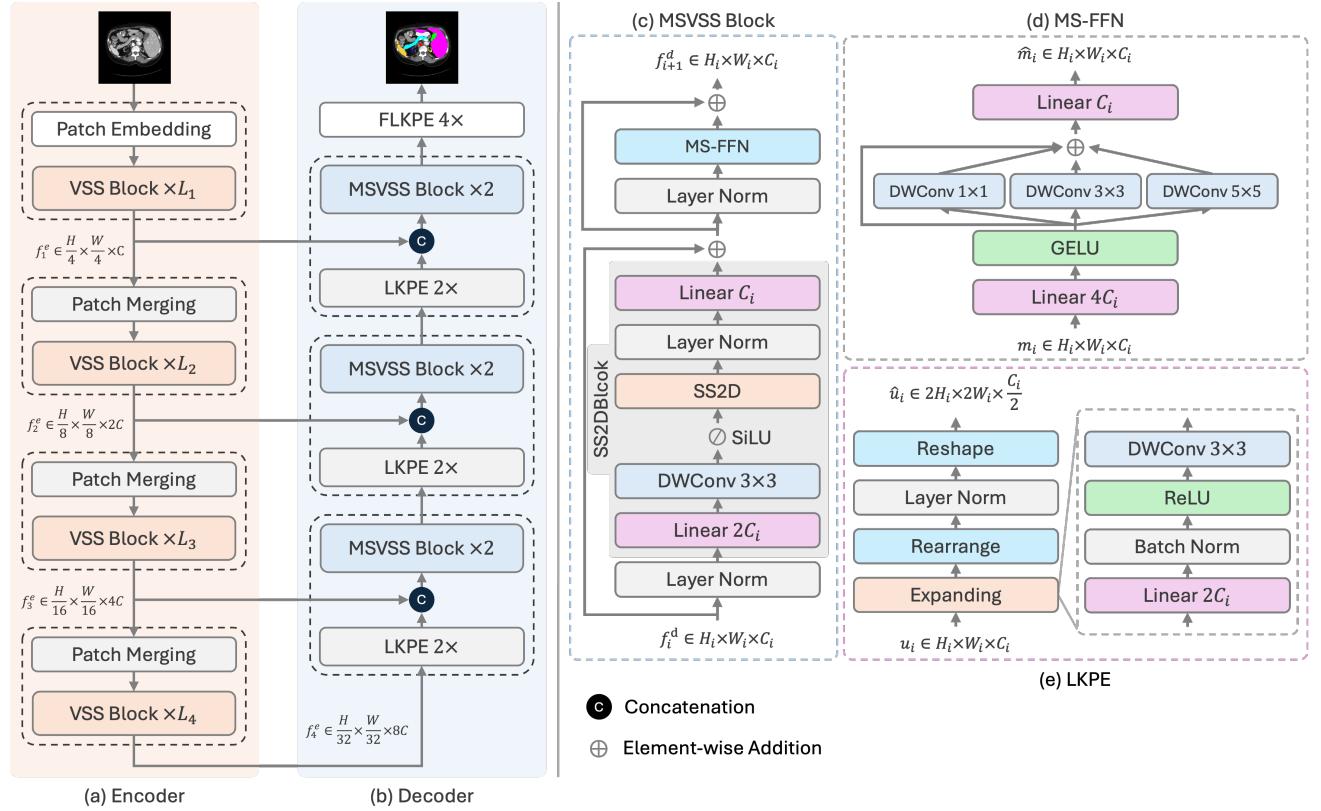


Fig. 1. The overall architecture of our proposed MSVM-UNet. (a) The VMamba V2 encoder backbone network. (b) the decoder network composed of LKPE layers, MSVSS blocks, and FFLKPE layers, (c) the Multi-Scale Visual State Space (MSVSS) block, (d) the Multi-Scale Feed-Forward Network (MS-FFN), and (e) the Large Kernel Patch Expanding (LKPE) layer. f_1^e , f_2^e , f_3^e , and f_4^e are the output features of the four stages of hierarchical encoder backbones. f_i^d and f_{i+1}^d represents the input and output features of the i^{th} stage of the decoder, respectively.

where f_i^d and f_{i+1}^d denote the input and output feature maps of the i^{th} stage, respectively. \hat{f}_i^d represents the output of the SS2DBlock, and $LN(\cdot)$ denotes layer normalization.

1) **2D-Selective-Scan Block (SS2DBlock):** 2D-Selective-Scan block performs selective scanning along four scanning paths on the input feature map to capture global contextual information and long-range dependencies. Specifically, the 2D input feature map first undergoes a linear layer, a depth-wise convolution operation, and an activation function. Then, further feature extraction is carried out through the 2D-Selective-Scan (SS2D) operation. Finally, the output is obtained after another layer normalization and a linear projection. As shown in Fig. 2(a), SS2D first flattens the 2D input along four different scanning paths to obtain four 1D sequences. These sequences are then fed into the S6 blocks [11] for selective scanning to model long-range dependencies. Finally, the four 1D sequences are restored to their original 2D form and summed to produce the output. The definition of the SS2DBlock is given by Equation (3):

$$\hat{z}_i = C_2(LN(SS2D(\sigma_1(DWConv(C_1(z_i)))))) \quad (3)$$

where z_i and \hat{z}_i represent the input and output feature maps of the SS2DBlock in the i^{th} stage, respectively. $C_1(\cdot)$ represents a linear projection used to double the channel dimension. $DWConv(\cdot)$ denotes a depth-wise convolution with a kernel

size of 3×3 . σ_1 represents a $SiLU(x) = x \cdot sigmoid(x)$ activation function. $SS2D(\cdot)$ denotes 2D-Selective-Scan operation. $C_2(\cdot)$ represents another linear projection used to reduce the channel dimension by half.

2) **Multi-Scale Feed-Forward Neural Network (MS-FFN):** As depicted in Fig. 2(b), we introduce convolution operations in the feed-forward network to aggregate information from these four diagonal directions. Additionally, to effectively capture detailed information and multi-scale feature representations in hierarchical features, we employ a set of convolution operations with different kernel sizes. To avoid introducing excessive parameters and computational overhead, we use depth-wise convolutions, which are parameter and computation efficient, to implement the MS-FFN. As shown in Fig. 1(d), the MS-FFN consists of two linear layers and a set of parallel depth-wise convolution layers. The definitions of MS-FFN are given by Equations (4) and (5):

$$m'_i = \sigma_2(C_3(m_i)) \quad (4)$$

$$\hat{m}_i = C_4\left(\sum_{ks \in KS} DWConv_{ks}(m'_i) + m'_i\right) \quad (5)$$

where m_i and \hat{m}_i represent the input and output tensors of the MS-FFN in the i^{th} stage, respectively, and m'_i denote the output of the first linear transformation. $C_3(\cdot)$ represents a linear projection used to expand the channel dimension by four

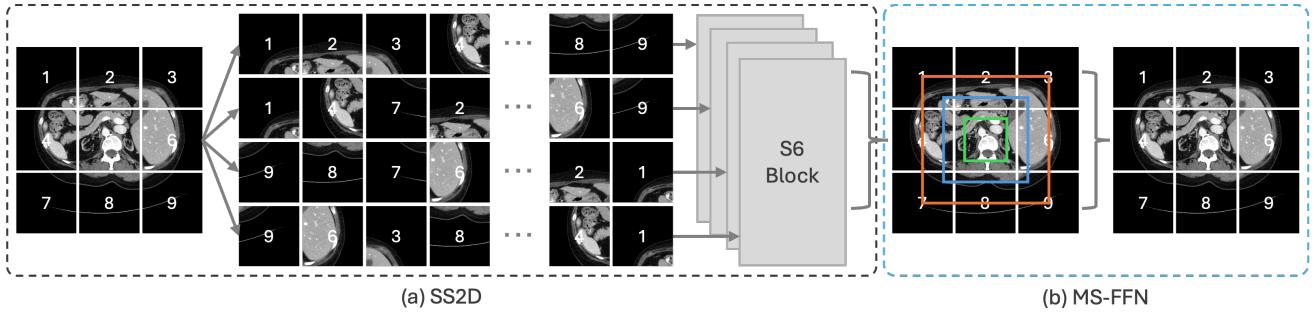


Fig. 2. Illustration of the core operations in the MSVSS block. (a) The 2D-Selective-Scan (SS2D) operation. The input are first divided into patches and then flattened along four scanning paths, and then sent to S6 respectively. Finally, each of them is restored according to the scanning path and added together to obtain the output. (b) The Multi-scale Feed-Forward Neural Network (MS-FFN) layer. The input passes through multi-scale convolutions to further aggregate diagonal information and capture multi-scale information representation.

times. $DWConv_{ks}(\cdot)$ denotes a depth-wise convolution with a kernel size of ks . KS defines a group of parallel convolution kernels with values of $\{1 \times 1, 3 \times 3, 5 \times 5\}$. $\sigma_2(\cdot)$ represents a $GELU(x) = x \cdot \Phi(x)$ activation function. $C_4(\cdot)$ represents another linear projection that reduces the channel dimension back to the input dimension.

C. Large Kernel Patch Expanding (LKPE) Layer

We use the LKPE layer to upsample the feature maps of the current stage to match the dimensions of the feature maps from the skip connection. As shown in Fig. 3, it compares our proposed LKPE layer with the patch expanding layer proposed by Swin-UNet. Unlike the latter method, which relies solely on linear projections (equivalent to convolutions with a kernel size of 1×1) for expanding the input feature channel dimension, we consider introducing large convolution kernels. Inspired by other upsampling methods, such as transposed convolutions and UpSample [3], the patch expanding layer only considers the channel information of features and neglects the spatial relationships between adjacent features, making this approach suboptimal in terms of information utilization. To solve this problem, we propose the large kernel patch expanding layer. Specifically, LKPE first applies a 1×1 convolution to double the channel dimension, followed by batch normalization and a ReLU activation function. Next, it uses an efficient depth-wise convolution to aggregate spatial information, and finally performs upsampling by expanding the feature representation that integrates both spatial and channel information. The definition of LKPE is given by Equation (6):

$$\hat{u}_i = RS(LN(RA(DWConv(\sigma_3(BN(C_5(u_i))))))) \quad (6)$$

where u_i and \hat{u}_i represent the feature maps before and after upsampling at the i^{th} stage, respectively. $C_5(\cdot)$ represents a linear transformation responsible for doubling the channel dimension. $BN(\cdot)$ represents batch normalization. $\sigma_3(\cdot)$ denotes a $ReLU(x) = \max(0, x)$ activation function. $RA(\cdot)$ denotes the rearrange operation, and $RS(\cdot)$ denotes the reshape operation.

D. Final Large Kernel Patch Expanding (FLKPE) Layer

We use the FFLKPE layer to generate the segmentation prediction. Taking the feature map from the final stage of the

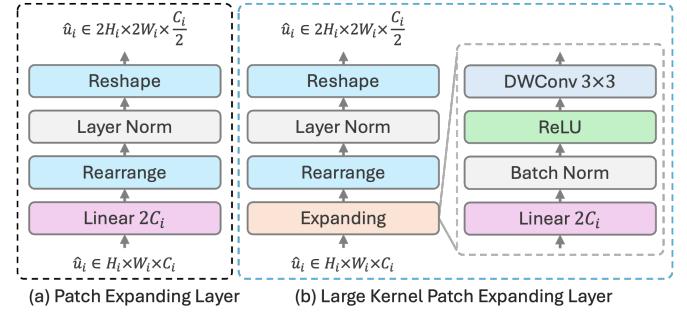


Fig. 3. Comparison of different patch expanding layers. (a) The patch expanding layer proposed by Swin-UNet, (b) the large kernel patch expanding (LKPE) layer proposed by us.

decoder as input, we first apply a linear projection to aggregate the channel dimension information and expand the channel dimension by 16 times. Next, we use a depth-wise convolution to aggregate spatial dimension information. Subsequently, we transform the spatial resolution of the resulting features to match the size of the input image while keeping the channel dimension unchanged. Finally, the transformed feature map is passed through a 1×1 convolution to map it to the segmentation prediction. The definition of FFLKPE is given by Equation (7):

$$\begin{aligned} \hat{f}_3^d &= RS(LN(RA(DWConv(\sigma_3(BN(C_6(f_3^d))))))) \\ y &= Conv_{1 \times 1}(\hat{f}_3^d) \end{aligned} \quad (7)$$

where f_3^d represents the feature map output by the last stage of the decoder. \hat{f}_3^d represents the feature map obtained by upsampling f_3^d to the input image size. y denotes the final segmentation prediction. $C_6(\cdot)$ represents a linear transformation responsible for expanding the input channel dimension by 16 times.

III. EXPERIMENTS AND RESULTS

A. Datasets

We evaluate the performance of the proposed MSVM-UNet on the Synapse abdominal multi-organ segmentation dataset (Synapse) and the Automated Cardiac Diagnosis Challenge dataset (ACDC).

TABLE I
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE SYNAPSE MULTI-ORGAN DATASET. BOLD BLACK DATA INDICATES THE BEST RESULTS, AND UNDERLINED BLACK DATA DENOTES THE SECOND-BEST RESULTS.

Methods	DSC (%) ↑	HD95 (mm) ↓	Aorta	Gallbladder	Kidney (L)	Kidney (R)	Liver	Pancreas	Spleen	Stomach
UNet [1]	74.82	54.59	85.66	53.24	81.13	71.60	92.69	56.81	87.46	69.93
Att-UNet [3]	71.70	34.47	82.61	61.94	76.07	70.42	87.54	46.70	80.67	67.66
TransUNet [2]	76.76	44.31	86.71	58.97	83.33	77.95	94.13	53.60	84.00	75.38
MISSFormer [20]	81.96	18.20	86.99	68.65	85.21	82.00	94.41	65.67	91.92	80.81
Swin-UNet [15]	79.13	21.55	85.47	66.53	83.28	79.61	94.29	56.58	90.66	76.60
PVT-CASCADE [3]	81.06	20.23	83.01	70.59	82.23	80.37	94.08	64.43	90.10	83.69
TransCASCADE [3]	82.68	17.34	86.63	68.48	87.66	84.56	94.43	65.33	90.79	83.52
2D D-LKA Net [21]	84.27	20.04	<u>88.34</u>	73.79	88.38	84.92	94.88	67.71	91.22	<u>84.94</u>
MERIT-GCASCADE [22]	<u>84.54</u>	10.38	88.05	<u>74.81</u>	88.01	<u>84.83</u>	<u>95.38</u>	<u>69.73</u>	91.92	<u>83.63</u>
PVT-EMCAD-B2 [23]	83.63	15.68	88.14	68.87	<u>88.08</u>	84.10	<u>95.26</u>	<u>68.51</u>	<u>92.17</u>	83.92
VM-UNet [16]	82.38	16.22	87.00	69.37	85.52	82.25	94.10	65.77	91.54	83.51
Swin-U-Mamba [14]	82.26	19.51	86.32	70.77	83.66	81.60	95.23	69.36	89.95	81.14
MSVM-UNet (ours)	85.00	14.75	88.73	74.90	85.62	84.47	95.74	71.53	92.52	86.51

1) **Synapse dataset** [24]: The dataset consists of 30 abdominal CT scans with 3779 axial contrast-enhanced abdominal CT images. Each CT volume contains 85 to 198 slices of 512×512 pixels, with a voxel spatial resolution of $([0.54 \sim 0.54] \times [0.98 \sim 0.98] \times [2.5 \sim 5.0]) \text{ mm}^3$. Similar to TransUNet [2], we randomly divide the dataset into 18 cases for training and 12 cases for testing. We segment only 8 types of abdominal organs: aorta, gallbladder, left kidney, right kidney, liver, pancreas, spleen, and stomach.

2) **ACDC dataset** [25]: The dataset contains 100 cardiac MRI scan images, each of which consists of three sub-organs: right ventricle (RV), myocardium (Myo), and left ventricle (LV). Following the TransUNet [2], we split the data into 70 cases for training, 10 for validation, and 20 for testing.

B. Implementation Details and Evaluation Metrics

1) **Implementation Details**: In our experiments, all models were implemented based on the Pytorch 2.0.0 framework, and all training was conducted on an NVIDIA GeForce RTX 3090 GPU. We initialized the backbone network using pretrained weights from the ImageNet-1k dataset. To reduce overfitting and enhance the model's generalization capability, we employed extensive data augmentation techniques, including resizing the input images to 224×224 , horizontal and vertical flips, random rotations, Gaussian noise, Gaussian blur, and contrast enhancement. We set the batch size to 32 and used the AdamW optimizer to train the network for a maximum of 300 epochs. The initial learning rate was set to 5e-4 and was decayed during training using a cosine annealing schedule. Due to the varying difficulty levels of different datasets and to reduce overfitting, we set different weight decay for different datasets: 1e-3 for the Synapse dataset and 1e-4 for the ACDC dataset. Additionally, the reported FLOPs and parameter counts of the models were calculated using calfflops with an input size of $224 \times 224 \times 3$. We used a combination of Dice and cross-entropy loss functions to train the network, defined as follows:

$$L_{total} = \alpha L_{DICE} + (1 - \alpha) L_{CE} \quad (8)$$

where $\alpha = 0.6$ and $1 - \alpha = 0.4$ are the weights for the Dice loss L_{DICE} and the cross-entropy loss L_{CE} , respectively.

2) **Evaluation Metrics**: Following the commonly adopted metrics to measure model performance, we used the Dice Score (DSC) and the 95% Hausdorff Distance (HD95) to assess our model's performance on the Synapse and ACDC datasets. The DSC and HD are calculated according to Equations (9) and (10):

$$DSC(X, Y) = \frac{2 * |X \cap Y|}{|X| + |Y|} \quad (9)$$

$$\begin{aligned} HD(X, Y) &= \max\{h(X, Y), h(Y, X)\} \\ h(X, Y) &= \max_{x \in X} \min_{y \in Y} d(x, y) \\ h(Y, X) &= \max_{y \in Y} \min_{x \in X} d(x, y) \end{aligned} \quad (10)$$

where the X and Y denote the ground truth and segmented maps, respectively, and $d(x, y)$ represents the distance between points x and y . HD95 is the 95th percentile of the distances of the boundaries of X and Y .

C. Comparisons with State-of-the-Arts

To verify the effectiveness of our proposed method, we compared its performance with state-of-the-art methods based on CNN, transformer, and mamba.

1) **Results on Synapse Multi-Organ Segmentation**: As shown in Table I, compared to various types of methods, our proposed MSVM-UNet achieved the best average DSC of 85.00% and HD95 of 14.75mm. Specifically, compared to CNN-based methods (such as 2D D-LKA Net), our method improved the DSC and HD95 by 0.73% and 5.29mm, respectively; by 1.37% and 0.93mm compared to transformer-based methods (such as PVT-EMCAD-B2); and by 2.62% and 1.47mm compared to mamba-based methods (such as VM-UNet). Additionally, for the small organs, there were 0.09% and 1.80% DSC improvements in the gallbladder and pancreas, respectively, compared to the second-best methods, and a 1.57% improvement in the stomach for the large organ. This is because MSVM-UNet can effectively capture both

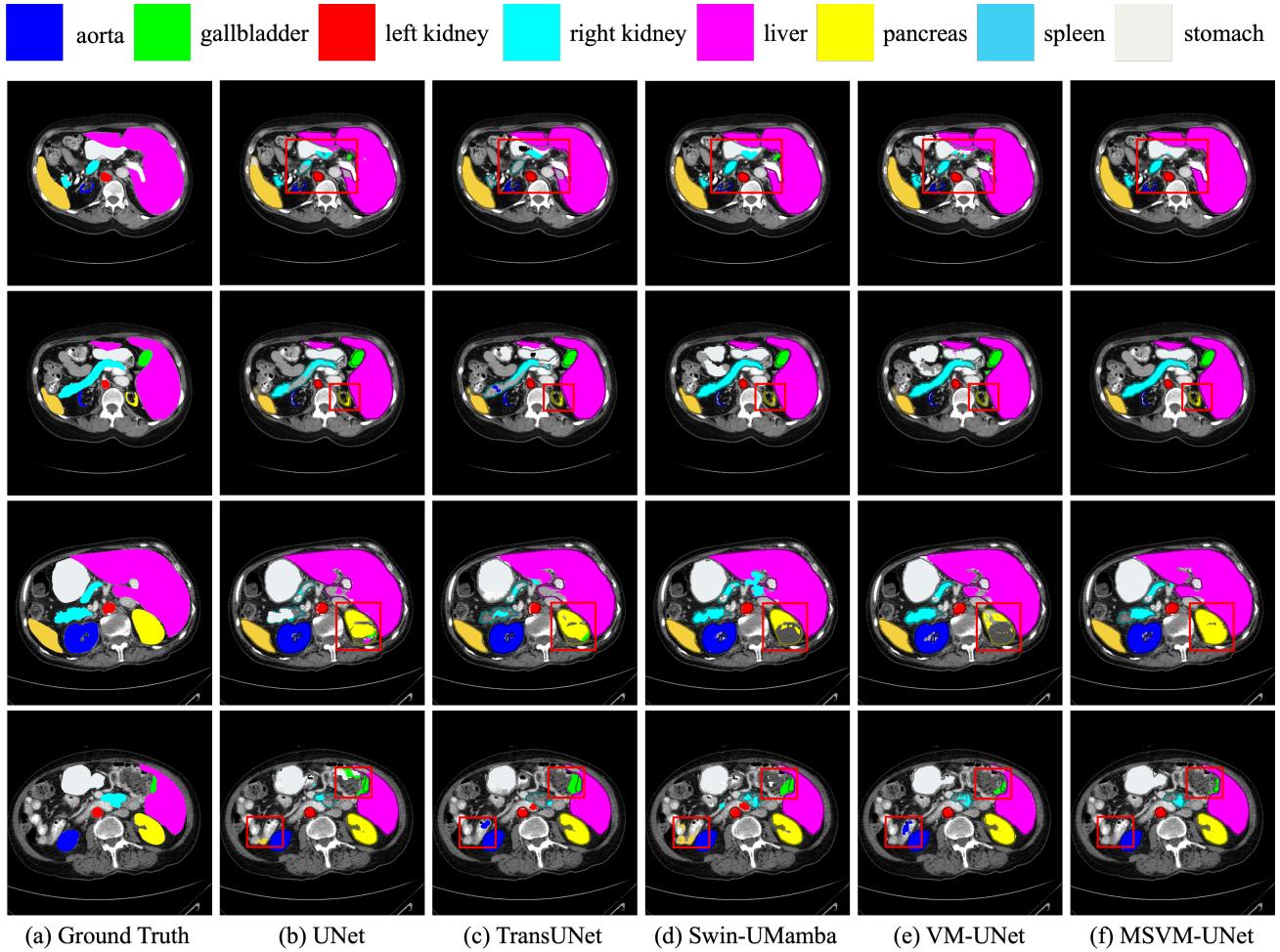


Fig. 4. Visual comparison of different methods on the Synapse multi-organ dataset. The first column represents the ground truth, and the following columns represent the segmentation predictions of UNet, TransUNet, Swin-UMamba, VM-UNet, and MSVM-UNet methods, respectively. The superiority of our proposed method can be clearly seen in the organ regions highlighted by red rectangles. Various colored contour lines indicate the ground truth for the corresponding organs.

long-range dependencies between pixels and local contextual relationships simultaneously. Due to the introduction of multi-scale convolution operations, MSVM-UNet not only handles organs of varying shapes and sizes effectively but also better locates organ boundaries.

2) Results on ACDC for Automated Cardiac Segmentation: As shown in Table II, we present the performance of our method compared to the aforementioned types of methods on MRI medical images, where our proposed MSVM-UNet achieved the best average DSC of 92.58%. Additionally, for the three categories in the ACDC dataset (RV, Myo, and LV), our method achieved the best DSC results of 91.00%, 90.35%, and 96.39%, respectively. This demonstrates the generalizability of our method, as it performs well on different modalities of medical image data (MRI and CT).

D. Qualitative Analysis

As shown in Fig. 4, we present a 2D visual comparison of our method with other methods on the Synapse multi-organ dataset. It can be observed that our method produces

TABLE II
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE ACDC DATASET. **BOLD BLACK DATA INDICATES THE BEST RESULTS, AND UNDERLINED BLACK DATA DENOTES THE SECOND-BEST RESULTS.**

Methods	DSC (%) ↑	RV	Myo	LV
R50 UNet [2]	87.55	87.10	80.63	94.92
R50 Att-UNet [2]	86.75	87.58	79.20	93.47
TransUNet [2]	89.71	88.86	84.53	95.73
Swin-UNet [15]	90.00	88.55	85.62	95.83
MISSFormer [20]	90.86	89.55	88.04	94.99
PVT-CASCADE [3]	91.46	88.90	89.97	95.50
TransCASCADE [3]	91.63	89.14	<u>90.25</u>	95.50
MERIT-GCASCADE [22]	92.23	90.64	89.96	<u>96.08</u>
PVT-EMCAD-B2 [23]	92.12	90.65	89.68	96.02
VM-UNet [16]	<u>92.24</u>	90.74	89.93	96.03
Swin-UMamba [14]	92.14	<u>90.90</u>	89.80	95.72
MSVM-UNet (ours)	92.58	91.00	90.35	96.39

better segmentation results across different organs and, to some extent, avoids issues of over-segmentation (as seen in the segmentation of the gallbladder in the last row) and under-segmentation (as seen in the segmentation of the pancreas in

TABLE III

ABLATION STUDY ON THE EFFECT OF DIFFERENT COMPONENTS OF MSVM-UNET WITH VMAMBA V2 TINY ENCODER ON THE SYNAPSE MULTI-ORGAN DATASET.

Components		DSC (%)	HD95 (mm)	#FLOPs (G)	#Params (M)
MSVSS	LKPE				
No	No	82.80	27.29	15.11	35.68
Yes	No	84.67	15.83	15.40	35.87
No	Yes	84.30	14.21	15.23	35.74
Yes	Yes	85.00	14.75	15.53	35.93

TABLE IV

ABLATION STUDY ON THE EFFECT OF DIFFERENT UPSAMPLING METHODS ON THE SYNAPSE MULTI-ORGAN DATASET.

Upsampling	DSC (%)	HD95 (mm)	#FLOPs (G)	#Params (M)
Transposed Conv	83.14	13.84	4.45	6.06
UpSample [3]	83.94	16.65	7.22	8.00
Patch Expand [15]	82.80	27.29	5.44	6.20
LKPE (ours)	84.30	14.21	5.56	6.26

the third row). This is because our MSVM-UNet can better capture features with varying geometric shapes. Compared to mamba-based methods, our approach shows better segmentation performance for organ boundaries. Additionally, when comparing the liver segmentation in the first row, we found that methods incorporating convolution operations achieve better results in delineating the liver boundary compared to methods without convolution operations. This is due to the inclusion of appropriate convolution operations, which help the model capture local detail features and positional information, leading to more discriminative feature representations and better segmentation results.

E. Ablation Studies

We conducted a comprehensive ablation study on the Synapse dataset to validate and investigate the effectiveness of our proposed method and to demonstrate the rationale behind our architectural design choices. Unless otherwise specified, all ablation experiments use the tiny version of VMamba V2 pre-trained on the ImageNet-1k as the encoder.

1) **Effect of Different Components of MSVM-UNet:** We conducted a series of experiments on the Synapse multi-organ dataset to understand the impact of different components in the MSVM-UNet decoder. We assessed the effects of replacing specific modules in the decoder with our proposed modules. As shown in Table III, significant performance improvements were observed after substituting the original modules with our proposed ones, with only a minimal increase in computational overhead and the number of parameters. Specifically, compared to the decoder with VSS blocks and patch expanding layers, using our proposed modules improved the DSC and HD95 by 2.2% and 12.54mm, respectively, with only an additional 0.42G FLOPs and 0.25M parameters. This demonstrates both the effectiveness and efficiency of our proposed MSVM-UNet.

2) **Effect of Different Upsampling Methods:** To explore the effectiveness of our proposed LKPE, we conducted experi-

TABLE V

ABLATION STUDY ON THE EFFECT OF MULTI-SCALE KERNELS IN DEPTH-WISE CONVOLUTION OF THE MSVSS BLOCK ON THE SYNAPSE MULTI-ORGAN DATASET.

Conv. kernels	DSC (%)	HD95 (mm)	#FLOPs (G)	#Params (M)
[1, 3, 5]	84.67	15.83	5.73	6.39
[3, 5, 7]	84.45	18.65	6.14	6.65
[1, 3, 5, 7]	84.40	18.33	6.14	6.66

TABLE VI

ABLATION STUDY ON THE EFFECT OF THE ENCODER MODEL SCALES ON THE SYNAPSE MULTI-ORGAN DATASET.

Encoder scales	DSC (%)	HD95 (mm)	#FLOPs (G)	#Params (M)
tiny	85.00	14.75	15.53	35.93
small	84.75	14.66	22.86	54.69

ments on the Synapse multi-organ dataset by using the original decoder with transposed convolution, the UpSample block [3], the patch expanding layer, and the LKPE layer as upsampling layers, respectively, to evaluate their individual performance. As shown in Table IV, we reported the performances and overheads corresponding to different upsampling methods. To provide a clearer comparison of the computational overhead and the number of parameters of different upsampling operations, we only report the FLOPs and parameters of the decoder. Compared to the original patch expanding layer, LKPE improved the DSC and HD95 by 1.5% and 13.08mm, respectively, while only introducing an additional 0.12G FLOPs and 0.06M parameters. Moreover, it can be observed that methods aggregating channel and spatial information generally achieve better performance, further demonstrating the effectiveness of our proposed LKPE.

3) Effect of Multi-Scale Kernels in MSVSS Block:

We also conducted additional experiments on the Synapse multi-organ dataset to explore the impact of different multi-scale convolution kernels in the depth-wise convolution of the MSVSS block. As shown in Table V, the performances for various multi-scale convolution kernels are reported. Similarly, the FLOPs and the number of parameters are provided only for the decoder. To avoid excessive computational overhead, we designed three sets of kernels: 1×1 and 7×7 in the first set, 3×3 and 5×5 in the second, and a union of both in the third. As the number and size of convolution kernels increase, we found that the performance decreases. Based on these observations, we selected [1, 3, 5] as the default multi-scale convolution kernels in the MSVSS block.

4) **Effect of Encoder Model Scales:** To investigate the impact of different encoder depths on model performance, we conducted two sets of experiments on the Synapse multi-organ dataset to study the effects of varying encoder scales. As shown in Table VI, there is a slight decrease in performance as the encoder becomes larger and deeper. Since both sets of experiments used the same setup, we hypothesize that this minor performance drop may be due to slight overfitting caused by the increased complexity of the model. Based on these observations and considering the computational overhead

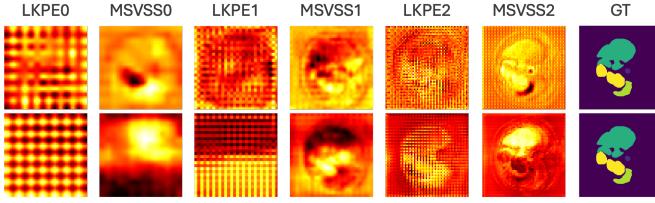


Fig. 5. Visual comparison of decoder features with and without our proposed blocks. The first and second rows present the features of our decoder (with our proposed blocks) and the original decoder (without our proposed blocks), respectively. Layer numbers are numbered from the bottom to the top of the decoder.

and the number of parameters, we chose the tiny version of the encoder as the default scale.

5) Effect of Feature Enhancement: The features of the corresponding layers of the original decoder and our MSVM-UNet decoder are visualized in Fig. 5. We compute the average of all channels in the feature map and then produce the heatmap using Matplotlib. It is evident from Fig. 5 that our method helps in handling organs of varying sizes and shapes and in obtaining more discriminative feature representations.

IV. CONCLUSION

In this paper, we propose a novel multi-scale visual Mamba UNet to address medical image segmentation challenges. Thanks to the design of multi-scale depth-wise convolution, MSVM-UNet not only captures information at various scales and models long-range dependencies in all directions but also maintains computational efficiency and acceptable parameter counts. Additionally, by effectively integrating channel and spatial information for upsampling, MSVM-UNet achieves more discriminative feature representations, leading to more accurate medical image segmentation results. Our experimental results demonstrate the outstanding performance of MSVM-UNet on two medical image datasets, with improvements in DSC and HD95 on the Synapse multi-organ dataset surpassing VM-UNet by 2.62% and 1.47mm, respectively. Qualitative analysis also shows that MSVM-UNet can accurately localize organs and handle organs of varying sizes and shapes.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (62062067).

REFERENCES

- [1] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention-MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III* 18. Springer, 2015, pp. 234–241.
- [2] 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.
- [3] M. M. Rahman and R. Marculescu, “Medical image segmentation via cascaded attention decoding,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 6222–6231.
- [4] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.
- [5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10012–10022.
- [7] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, “Pvt v2: Improved baselines with pyramid vision transformer,” *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.
- [8] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. W. Lau, “Biformer: Vision transformer with bi-level routing attention,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 10323–10333.
- [9] A. Gu, K. Goel, and C. Ré, “Efficiently modeling long sequences with structured state spaces,” *arXiv preprint arXiv:2111.00396*, 2021.
- [10] H. Mehta, A. Gupta, A. Cutkosky, and B. Neyshabur, “Long range language modeling via gated state spaces,” *arXiv preprint arXiv:2206.13947*, 2022.
- [11] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” *arXiv preprint arXiv:2312.00752*, 2023.
- [12] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu, “Vmamba: Visual state space model,” *arXiv preprint arXiv:2401.10166*, 2024.
- [13] J. Ma, F. Li, and B. Wang, “U-mamba: Enhancing long-range dependency for biomedical image segmentation,” *arXiv preprint arXiv:2401.04722*, 2024.
- [14] J. Liu, H. Yang, H.-Y. Zhou, Y. Xi, L. Yu, Y. Yu, Y. Liang, G. Shi, S. Zhang, H. Zheng *et al.*, “Swin-umamba: Mamba-based unet with imagenet-based pretraining,” *arXiv preprint arXiv:2402.03302*, 2024.
- [15] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, “Swin-unet: Unet-like pure transformer for medical image segmentation,” in *European conference on computer vision*. Springer, 2022, pp. 205–218.
- [16] J. Ruan and S. Xiang, “Vm-unet: Vision mamba unet for medical image segmentation,” *arXiv preprint arXiv:2402.02491*, 2024.
- [17] Z. Yang and S. Farsiu, “Directional connectivity-based segmentation of medical images,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 11525–11535.
- [18] M. Heidari, A. Kazerouni, M. Soltany, R. Azad, E. K. Aghdam, J. Cohen-Adad, and D. Merhof, “Hiformer: Hierarchical multi-scale representations using transformers for medical image segmentation,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 6202–6212.
- [19] C.-F. R. Chen, Q. Fan, and R. Panda, “Crossvit: Cross-attention multi-scale vision transformer for image classification,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 357–366.
- [20] X. Huang, Z. Deng, D. Li, and X. Yuan, “Missformer: An effective medical image segmentation transformer,” *arXiv preprint arXiv:2109.07162*, 2021.
- [21] R. Azad, L. Niggemeier, M. Hüttemann, A. Kazerouni, E. K. Aghdam, Y. Velichko, U. Bagci, and D. Merhof, “Beyond self-attention: Deformable large kernel attention for medical image segmentation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 1287–1297.
- [22] M. M. Rahman and R. Marculescu, “G-cascade: Efficient cascaded graph convolutional decoding for 2d medical image segmentation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 7728–7737.
- [23] M. M. Rahman, M. Munir, and R. Marculescu, “Emcad: Efficient multi-scale convolutional attention decoding for medical image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11769–11779.
- [24] B. Landman, Z. Xu, J. Iglesias, M. Styner, T. Langerak, and A. Klein, “Miccai multi-atlas labeling beyond the cranial vault-workshop and challenge,” in *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, vol. 5, 2015, p. 12.
- [25] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. G. Ballester *et al.*, “Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved?” *IEEE transactions on medical imaging*, vol. 37, no. 11, pp. 2514–2525, 2018.