A Permutable Hybrid Network for Volumetric Medical Image Segmentation

Yi Lin[†], Xiao Fang[†], Dong Zhang, Kwang-Ting Cheng, Hao Chen[(✉)]

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology, Kowloon, Hong Kong
jhc@cse.ust.hk

Abstract. The advent of Vision Transformer (ViT) has brought substantial advancements in 3D volumetric benchmarks, particularly in 3D medical image segmentation. Concurrently, Multi-Layer Perceptron (MLP) networks have regained popularity among researchers due to their comparable results to ViT, albeit with the exclusion of the heavy self-attention module. This paper introduces a permutable hybrid network for volumetric medical image segmentation, named PHNet, which exploits the advantages of convolution neural network (CNN) and MLP. PHNet addresses the intrinsic isotropy problem of 3D volumetric data by utilizing both 2D and 3D CNN to extract local information. Besides, we propose an efficient Multi-Layer Permute Perceptron module, named MLPP, which enhances the original MLP by obtaining long-range dependence while retaining positional information. Extensive experimental results validate that PHNet outperforms the state-of-the-art methods on two public datasets, namely, COVID-19-20 and Synapse. Moreover, the ablation study demonstrates the effectiveness of PHNet in harnessing the strengths of both CNN and MLP. The code will be accessible to the public upon acceptance.

1 Introduction

Computer-aided diagnosis (CAD) systems have gained popularity in the healthcare sector, assisting radiologists in diagnosing and treating patients. Convolution neural networks (CNN) have shown remarkable advancements, improving the performance of CAD, particularly in medical image segmentation [2,3]. Over the past decades, significant research efforts have focused on developing efficient and robust segmentation methods for medical images. One of the most popular architectures for this task is U-Net [18], which employs an encoder-decoder structure and skip connections to capture the contextual information of the input image. Following the success of U-Net, many variants of U-Net have been proposed with various convolution-based blocks and different strategies for the skip connections, including ResUNet [8], Y-Net [16], and V-Net [17], etc.

Recently, attention mechanism-based Transformers have shown promising superiority in the field of natural language processing [26]. Subsequent studies

[†] Equal contribution; ✉ corresponding author.

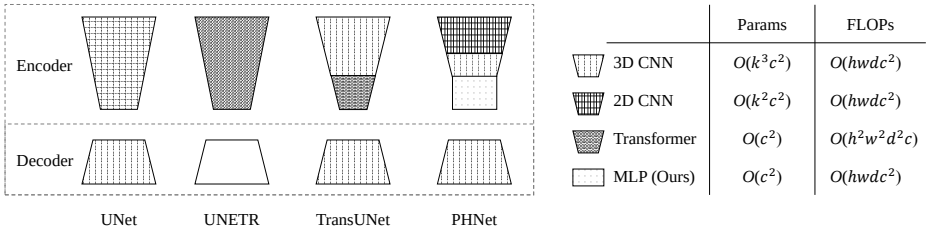

Fig. 1. Illustration of different architectures.

such as ViT [5] and DeiT [23] have demonstrated that the Transformer architecture can achieve state-of-the-art performance on versatile computer vision tasks. Given the remarkable advance of Transformers in natural image recognition tasks, many researchers have investigated the effectiveness of various neural networks for medical image segmentation. To name a few, TransUNet [3] proposed to employ a Transformer in the bottleneck of a U-Net architecture for global information communication. Similarly, UNETR [7] and CoTr [29] designed a hierarchical fusion of Transformer and CNN architecture.

Despite the recent developments in medical image segmentation through the use of Transformers, their heavy computational burden caused by the self-attention mechanism limits their practical application, particularly for 3D medical images, which require a considerable number of forward and backward passes [25]. In light of this, multi-layer perceptron (MLP)-based methods have regained interest in the research community, as they have demonstrated comparable performance with CNN and Transformers, without requiring the heavy self-attention mechanism [22]. For instance, MLP-Mixer [22] enabled information communication by a series of MLPs, capturing long-range dependencies in the input data. However, the effectiveness of MLP in volumetric medical image segmentation remains understudied.

This paper proposes PHNet, a novel **P**ermutable **H**ybrid **Net**work that combines both CNN and MLP for accurate volumetric medical image segmentation. PHNet employs an encoder-decoder architecture, with the encoder utilizing a 2.5D CNN structure that can capitalize on the inherent isotropy of medical images, while avoiding information loss in shallow layers by capturing the varying information density in different directions of volumetric medical images. The paper further proposes MLPP, a Multi-Layer Permute Perceptron module that can maintain the positional information while integrating global interdependence in a computationally-efficient manner. To enhance computational efficiency, token-group operations are introduced, which efficiently aggregate feature maps at a token level, reducing the number of computations required. This is the first attempt to investigate the effectiveness of combining CNN and MLPs in volumetric medical image segmentation. The proposed method is evaluated on two publicly available datasets, COVID-19 Lung CT Lesion Segmentation Challenge-2020 (COVID-19-20) [19] and Synapse Multi-Organ Segmentation [11]. Extensive ex-

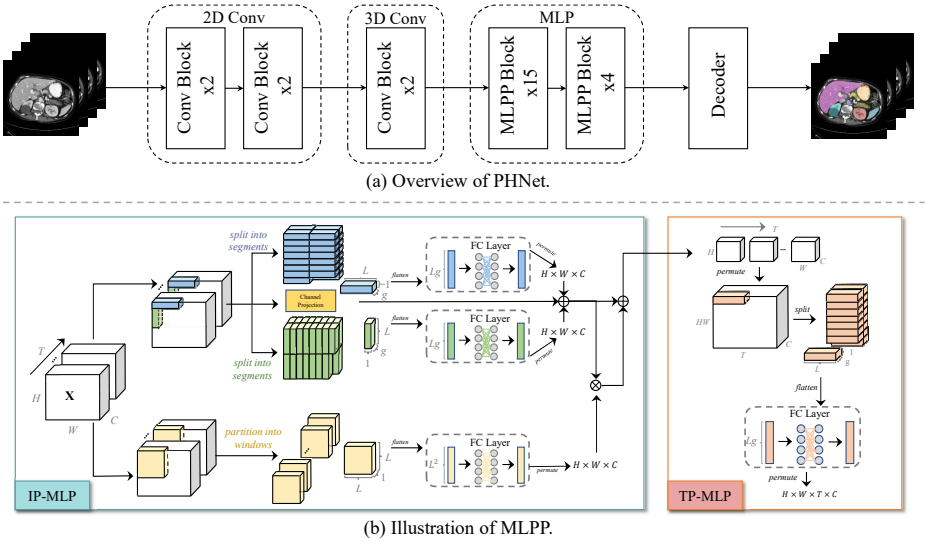

Fig. 2. (a) Overview of PHNet. (b) Detailed architecture of MLPP.

perimental results validate that PHNet achieves state-of-the-art accuracy on both datasets, surpassing the winner in the MICCAI Covid-19-20 challenge.

2 Method

This paper presents PHNet, which follows an encoder-decoder architecture, as illustrated in Fig. 2(a). The encoder consists of two primary components: a 2.5D convolution module (*Sec. 2.1*) and a Multi-Layer Permute Perceptron (MLPP) module (*Sec. 2.2*). The 2.5D convolutional stage extracts local features, and the output feature maps are passed to the MLPP module to learn global features. The decoder processes the hierarchical features for the final prediction. In the following sections, each component is explained in detail.

2.1 2.5D Convolution

Based on the previous work on bias in medical image analysis [1] and the anisotropic nature of volumetric medical images, we incorporate convolutional layers in the shallow layers of the encoder to extract local features. Volumetric images, such as CT and MRI scans, are often affected by anisotropic problems due to their thick-slice scanning [28], resulting in high in-plane (IP) resolution and low through-plane (TP) resolution [4,14]. This discrepancy is particularly pronounced in COVID-19-20 [19], where the IP resolution is 0.74mm on average, whereas the TP resolution is only 5mm. To address this issue, we use 2D conv-blocks to capture the IP information until the feature is reformulated

in approximately uniform resolution across all three axes: axial, coronal, and sagittal. Then, we apply 3D conv-blocks to handle the volumetric information. Each encoder layer consists of two residual convolution blocks, with each block comprising two sequential Convolution-Instance Normalization-Rectified Linear Unit (Conv-IN-ReLU) operations [24]. The residual addition takes place before the final ReLU activation.

2.2 Multi-Layer Permute Perceptron (MLPP)

Although CNNs are capable of modeling long-range dependencies through deep stacks of convolution layers, studies [25,9,22] have demonstrated that MLP-based networks are more effective in learning global context. Therefore, we design the MLPP module (Fig. 2(b))to obtain global information in deep layers. MLPP decomposes the training of in-plane (IP) feature and through-plane (TP) feature in sequential order. We denote these two blocks as **IP-MLP** and **TP-MLP**, respectively. To enable the communication of cross-axis tokens, we further propose an auxiliary attention branch in IP-MLP, which denotes as **AA-MLP**. We provide an example of the code based on the PyTorch in supplementary material.

IP-MLP. Given one slice of input feature maps $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, common MLP-based methods [12,22] typically directly flatten $\mathbf{X}$ into a 1D vector, which loses the spatial information of the original conv-features [30]. Contrarily, as shown in Fig. 2, we introduce axial decomposition in triplet pathways that separately process $\mathbf{X}$ in horizontal (W), vertical (H), and channel (C) axis, respectively. This enables the preservation of precise positional information along other axes when encoding information along one axis. Next, to balance the long-distance interaction and computation cost, and alleviate image resolution sensitivity problem [13], we present a token segmentation operation[†] that splits the feature vector into multiple tokens, which can be efficiently processed by the following fully-connected (FC) layers. We take the horizontal axis as an example. Instead of encoding the entire dimension, we split $\mathbf{X}$ into non-overlapping segments along the horizontal direction. Then, we set segment length to L and thus obtain $\mathbf{X}_i \in \mathbb{R}^{L\times C}$, where $i \in \{1, ..., HW/L\}$. Similarly, we split each $\mathbf{X}_i$ into multiple non-overlapping groups along channel dimension, where each group has $g = C/L$ channels. Thus we obtain split segments, and each single segment is $\mathbf{X}_i^k \in \mathbb{R}^{Lg}$, where $k \in \{1, ..., C/g\}$. Next, we flatten each segment and map $\mathbb{R}^{Lg} \mapsto \mathbb{R}^{Lg}$ by an FC layer to transform each segment, producing $\mathbf{Y}_i^k$. To recover the original dimension, we permute all segments back to $\mathbf{Y}_W \in \mathbb{R}^{H\times W\times C}$. Similarly, we conduct the same operations in the vertical pathway as above to permute tokens along the vertical direction, yielding $\mathbf{Y}_H$. To enable communication among groups along channel dimension, we add a parallel branch that contains an FC layer that maps $\mathbb{R}^C \mapsto \mathbb{R}^C$ to process each token individually, yielding $\mathbf{Y}_C$. Finally, we feed the element-wise summation of horizontal, vertical, and channel features into a new FC layer to attain the output, which can be formulated as:

$$\mathbf{Y}_{\text{IP}} = (\mathbf{Y}_H + \mathbf{Y}_W + \mathbf{Y}_C)\mathbf{W}, \tag{1}$$

[†] For more discussion, please refer to the supplementary material.

where $\mathbf{W} \in \mathbb{R}^{C\times C}$ denotes an FC weight matrix.

AA-MLP. There are two limitations of the MLPP module that could potentially harm the segmentation performance. First, the axial decomposition cuts off the direct interaction of tokens that are not in the same horizontal or vertical position. Second, the token segmentation operation would suffer from a small local reception field compared with the vanilla MLP. To address these limitations, we design an auxiliary branch to enable intra-axis token communication, which serves as an attention function via a lightweight yet effective MLP-like architecture. Specifically, given an input slice of feature maps $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, we partition $\mathbf{X}$ into non-overlapping windows. We set window size to L and thus obtain $\mathbf{X}_i \in \mathbb{R}^{L\times L}$, where $i \in \{1, ..., HWC/L^2\}$. Then we apply an FC matrix $\mathbf{W} \in \mathbb{R}^{L^2\times L^2}$ to transform each window and get $\mathbf{Y}_i \in \mathbb{R}^{L\times L}$. The final attention map $\mathbf{Y}_A \in \mathbb{R}^{H\times W\times C}$ is obtained by permuting all windows back to the original dimension. Finally, the feature maps $\mathbf{F}_{\text{IP}}$ of IP-MLP are obtained by performing residual attention [27] of $\mathbf{Y}_{\text{IP}}$ and $\mathbf{Y}_A$.

$$\mathbf{F}_{\text{IP}} = (1 + \mathbf{Y}_A) \odot \mathbf{Y}_{\text{IP}}, \tag{2}$$

where $\odot$ denotes element-wise multiplication.

TP-MLP. Once obtained the in-plane information from the IP-MLP, TP-MLP is applied to capture the long-term through-plane features. Similarly, given input feature maps $\mathbf{F}_{\text{IP}} \in \mathbb{R}^{H\times W\times D\times C}$, we first split $\mathbf{X} = \mathbf{F}_{\text{IP}}$ along depth dimension into non-overlapping segments with segment length L. We thus obtain $\mathbf{X}_i \in \mathbb{R}^{L\times C}$, where $i \in \{1, ..., HWD/L\}$. Next, we split $\mathbf{X}$ into a couple of non-overlapping groups along channel dimension where each group has $g = C/L$ channels and get $\mathbf{X}_i^k \in \mathbb{R}^{Lg}$, where $k \in \{1, ..., C/g\}$. Then we flatten each segment and map $\mathbb{R}^{Lg} \mapsto \mathbb{R}^{Lg}$ by an FC layer, yielding $\mathbf{Y}_i^k$. Finally, we permute all segments $\mathbf{Y}_i^k \in \mathbb{R}^{Lg}$ back to original dimension and output $\mathbf{F}_{\text{TP}} \in \mathbb{R}^{H\times W\times D\times C}$.

2.3 Decoder

The decoder in our proposed method utilizes a pure CNN architecture, employing transpose convolution to progressively upsample the feature maps to match the input image resolution. Following the upsampling process, a residual convolution block is used to refine the feature maps. To further enhance segmentation accuracy, we include skip connections between the encoder and decoder, allowing for the preservation of low-level details.

3 Experiments

3.1 Datasets

We conduct experiments on two publicly available datasets: COVID-19-20 [19] and Synapse [11]. COVID-19-20 is comprised of 249 unenhanced chest CT scans, with 199 samples designated for training and 50 samples for testing. All samples in this dataset are positive for SARS-CoV-2 RT-PCR. Synapse consists of 30

cases of CT scans, with 14, 4, and 12 cases designated for training, validation, and testing, respectively [2]. For COVID-19-20, we utilize the official evaluation metrics from the challenge [19], which include Dice coefficient (Dice), Intersection over Union (IoU), Surface Dice coefficient (SD), Normalized Volume Difference (NVD), and Hausdorff Distance (HD). Meanwhile, following [3], we adopt Dice and HD as evaluation metrics for Synapse.

3.2 Implementation Details

PHNet is implemented using PyTorch and MONAI[†] framework and trained on an NVIDIA RTX 3090 GPU. For COVID-19-20, all images are interpolated into the voxel spacing of $0.74\times0.74\times5.00\text{mm}^3$. Three sub-volumes of $224\times224\times28$ are sampled from each scan. We train PHNet in a total of 250 epochs for Synapse and 450 epochs for COVID-19-20. For all experiments we adopt the AdamW optimizer [15] with an initial learning rate $lr = 10^{-3} \times \frac{\text{batch\_size}}{1024}$ and a weight decay of 5×10^{-2}, as suggested by [9]. Except for the above, we follow baseline from [19] and [10] for the COVID-19-20 and Synapse datasets, respectively.

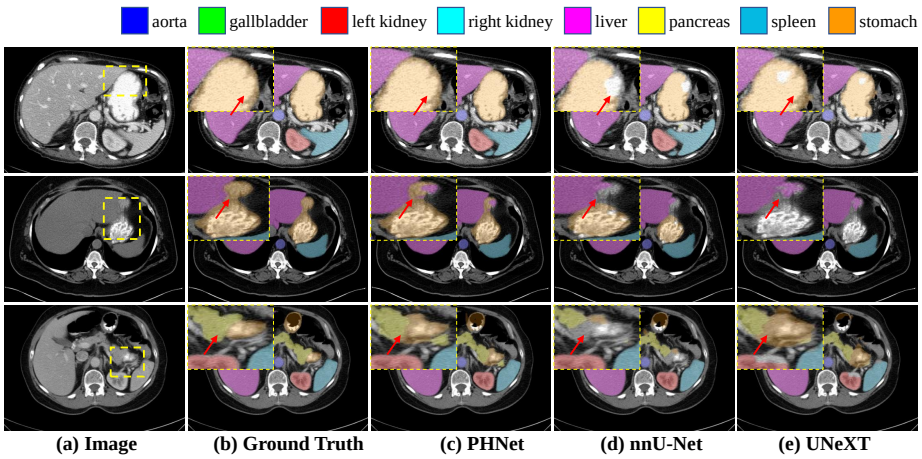

Fig. 3. Qualitative visualizations of different methods on the Synapse [11] dataset. Regions of evident improvements are enlarged to show better details.

3.3 Comparisons with State-of-the-art Methods

We compare our method with three types of methods, *i.e.*, CNN-based, Transformer-based, and MLP-based methods. Quantitative results on COVID-19-20 and Synapse datasets are reported in Table 1 and Table 2, respectively. In both tables,

[†] https://monai.io/

Table 1. Result comparisons with SOTAs on COVID-19-20 [19].

Method	Dice ↑	IoU ↑	SD ↑	NVD ↓	HD ↓	Method	Dice ↑
UNet [6]	0.6909	0.5527	0.6283	0.4079	134.76	Rank 1	0.7709
nnUNet [10]	0.7251	0.5940	0.6904	0.2787	123.65	Rank 2	0.7687
UNETR [7]	0.5718	0.4310	0.5120	0.4464	174.40	Rank 3	0.7687
SwinUNETR [21]	0.6365	0.5013	0.5819	0.3720	141.42	Rank 4	0.7678
PHNet*	0.7614	0.6305	**0.7266**	**0.2020**	**101.38**	Rank 5	0.7677
PHNet	**0.7634**	**0.6336**	0.7210	0.2060	108.16	Ours	**0.7718**

Table 2. Result comparisons with SOTAs on Synapse [11].

Method	Dice ↑	HD ↓	Aorta	Gallb.	Kid(L)	Kid(R)	Liver	Panc.	Spleen	Stom.
UNeXt [25]	67.07	40.47	76.43	51.64	74.54	67.94	91.11	34.95	79.20	60.70
TransUnet [3]	77.48	31.69	87.23	63.13	81.87	77.02	94.08	55.86	85.08	75.62
SwinUnet [2]	79.13	21.55	85.47	66.53	83.28	79.61	94.29	56.58	90.66	76.60
UNETR [7]	79.57	23.87	89.99	60.56	85.66	84.80	94.46	59.25	87.81	73.99
CoTr [29]	78.08	27.38	85.87	61.38	84.83	79.36	94.28	57.65	87.74	73.55
nnUNet [10]	83.98	**12.56**	**91.73**	66.22	87.30	84.41	96.15	76.04	94.81	75.20
PHNet*	84.94	18.29	91.34	68.04	**87.68**	**85.22**	96.17	76.19	95.45	79.42
PHNet	**85.54**	14.62	89.83	**74.26**	87.05	84.62	**96.29**	**77.19**	**95.50**	**79.55**

PHNet* stands for PHNet w/o AA-MLP. Qualitative results on Synapse [11] are shown in Fig. 6, which demonstrate that the proposed method is able to generate more accurate and detailed segmentation results compared to other methods.

Results on Lung Lesion Segmentation. On COVID-19-20 [19], official evaluation results presented in Table 1 show that our method outperforms the existing baselines and achieves the best score in all metrics. Additionally, following [19], we perform five-fold cross-validation and model ensemble using our proposed method. The result demonstrates that our method achieves the highest dice score of 77.18%, outperforming the top-5 solutions in this challenge[†].

Results on Multi-Organ Segmentation. On Synapse [11], Table 2 shows that our method achieves the highest average dice score of 85.54% and second lowest HD of 14.62, outperforming the SOTA methods. Notably, as for Dice, the distinct improvements can be markedly observed for organs with blurry boundaries, such as the "Gallbladder" and the "Stomach," where our model achieves significant gains over the SOTA methods, *i.e.*, 7.73% and 2.95% in Dice, respectively.

3.4 Ablation Studies

We further conduct ablation studies on Synapse to validate each component in our method. For all models, we use the same decoder as our proposed method and fix the number of channels in each layer. The results are shown in Fig. 4.

[†] https://covid-segmentation.grand-challenge.org/evaluation/challenge/leaderboard

Comparisons with different architecture combinations. We compare the performance of different combinations of Conv, Attention, and MLP in shallow and deep layers. For Conv and MLP, We adopt the same module as PHNet. For Attention, we use Swin Transformer block [21] and set window size equal to segment length for a fair comparison. As shown in Fig. 4(a), a combination of Conv in shallow layers and MLP in deep layers achieves the best performance, which conforms to our argument that Conv excels at extracting local features while MLP is more effective in modeling long-range dependencies.
Comparisons with different MLP designs. We compare the performance with related MLP-based variations in Fig. 4(b), including MLP-Mixer (M) [22], ShiftMLP (S) [25], and WaveMLP (W) [20]. We only replace the MLPP module in our PHNet (P) with others for a fair comparison. Results show that our method gains 2.15%, 1.85%, and 0.45% improvements in Dice, respectively, demonstrating the decent performance of our design.
Impact of segment length. In Fig. 4(c), we investigate the impact of different segment lengths L in our PHNet. Expressly, the segment length is set to different ratios of the width (W), *i.e.*, 1, $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$, respectively. It could particularly benefit different sizes of regions of interest (ROI). Results show that the best performance is achieved when $L = \frac{1}{2}W$.
Impact of MLP layers. In Fig. 4(d), we study the influence of a different number of MLP layers in our PHNet. Results show that the best performance is achieved when the number of MLP layers is 2.

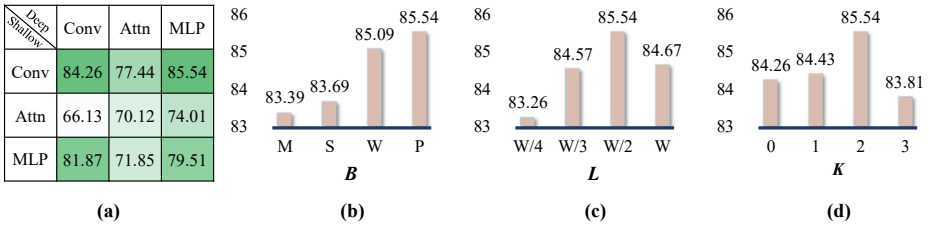

Fig. 4. Average Dice over all organs on Synapse versus (a) different architectures; (b) different MLP designs; (c) segment length L; and (d) the number of MLP layers K.

4 Conclusion

This paper presented a novel permutable hybrid network, referred to as PHNet, designed for the task of volumetric medical image segmentation. PHNet integrated 2D CNN, 3D CNN, and MLP to enable the extraction and integration of both local and global features. Besides, we introduced a permutable MLP block to address the problem of spatial information loss and to minimize the computational burden. Through extensive experiments conducted on two public datasets, our proposed PHNet demonstrated its effectiveness and outperformed

state-of-the-art approaches by a remarkable margin. Future work includes extending the proposed framework on versatile medical image analysis tasks, such as classification and disease localization, and systematically investigating the efficiency, effectiveness, and interactions of CNN, Transformer, and MLP.

References

1. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al.: Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261 (2018)
2. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-Unet: UNet-like pure transformer for medical image segmentation. In: European Conference on Computer Vision (2023)
3. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: TransUNet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
4. Dong, Z., He, Y., Qi, X., Chen, Y., Shu, H., Coatrieux, J.L., Yang, G., Li, S.: MNet: Rethinking 2D/3D networks for anisotropic medical image segmentation. In: International Joint Conferences on Artificial Intelligence (2022)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
6. Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., et al.: U-Net: deep learning for cell counting, detection, and morphometry. Nature Methods **16**(1), 67–70 (2019)
7. Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H.R., Xu, D.: UNETR: Transformers for 3D medical image segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 574–584 (2022)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
9. Hou, Q., Jiang, Z., Yuan, L., Cheng, M.M., Yan, S., Feng, J.: Vision permutator: A permutable MLP-like architecture for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(1), 1328–1334 (2022)
10. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**(2), 203–211 (2021)
11. Landman, B., Xu, Z., Igelsias, J., Styner, M., Langerak, T., Klein, A.: Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In: Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge. vol. 5, p. 12 (2015)
12. Liu, H., Dai, Z., So, D., Le, Q.V.: Pay attention to MLPs. Advances in Neural Information Processing Systems **34**, 9204–9215 (2021)
13. Liu, R., Li, Y., Tao, L., Liang, D., Zheng, H.T.: Are we ready for a new paradigm shift? a survey on visual deep MLP. Patterns **3**(7), 100520 (2022)

14. Liu, S., Xu, D., Zhou, S.K., Pauly, O., Grbic, S., Mertelmeier, T., Wicklein, J., Jerebko, A., Cai, W., Comaniciu, D.: 3D anisotropic hybrid network: Transferring convolutional features from 2D images to 3D anisotropic volumes. In: International Conference on Medical Image Computing and Computer Assisted Intervention. pp. 851–858. Springer (2018)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
16. Mehta, S., Mercan, E., Bartlett, J., Weaver, D., Elmore, J.G., Shapiro, L.: Y-Net: joint segmentation and classification for diagnosis of breast biopsy images. In: International Conference on Medical Image Computing and Computer Assisted Intervention. pp. 893–901. Springer (2018)
17. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: International Conference on 3D Vision. pp. 565–571. Ieee (2016)
18. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer Assisted Intervention. pp. 234–241. Springer (2015)
19. Roth, H.R., Xu, Z., Tor-Díez, C., Jacob, R.S., Zember, J., Molto, J., Li, W., Xu, S., Turkbey, B., Turkbey, E., et al.: Rapid artificial intelligence solutions in a pandemic—the covid-19-20 lung ct lesion segmentation challenge. Medical Image Analysis **82**, 102605 (2022)
20. Tang, Y., Han, K., Guo, J., Xu, C., Li, Y., Xu, C., Wang, Y.: An image patch is a wave: Phase-aware vision MLP. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10935–10944 (2022)
21. Tang, Y., Yang, D., Li, W., Roth, H.R., Landman, B., Xu, D., Nath, V., Hatamizadeh, A.: Self-supervised pre-training of swin transformers for 3D medical image analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 20730–20740 (2022)
22. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al.: MLP-mixer: An all-MLP architecture for vision. Advances in Neural Information Processing Systems **34**, 24261–24272 (2021)
23. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)
24. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
25. Valanarasu, J.M.J., Patel, V.M.: UNeXt: MLP-based rapid medical image segmentation network. In: International Conference on Medical Image Computing and Computer Assisted Intervention. pp. 23–33. Springer (2022)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
27. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3156–3164 (2017)
28. Wang, G., Shapey, J., Li, W., Dorent, R., Dimitriadis, A., Bisdas, S., Paddick, I., Bradford, R., Zhang, S., Ourselin, S., et al.: Automatic segmentation of vestibular schwannoma from T2-weighted mri by deep spatial attention with hardness-weighted loss. In: International Conference on Medical Image Computing and Computer Assisted Intervention. pp. 264–272. Springer (2019)

29. Xie, Y., Zhang, J., Shen, C., Xia, Y.: CoTr: Efficiently bridging cnn and transformer for 3D medical image segmentation. In: International Conference on Medical Image Computing and Computer Assisted Intervention. pp. 171–180. Springer (2021)
30. Zhang, D.J., Li, K., Wang, Y., Chen, Y., Chandra, S., Qiao, Y., Liu, L., Shou, M.Z.: MorphMLP: An efficient mlp-like backbone for spatial-temporal representation learning. In: European Conference on Computer Vision. pp. 230–248. Springer (2022)

Supplementary Materials

Resolution Sensitivity

Definition *A network cannot deal with a flexible resolution input once the number of parameters is set.* [13]
Vanilla MLP Given input feature maps $\mathbf{X} \in \mathbb{R}^{H\times W\times D\times C}$, the number of parameters could be set only with prior knowledge of H, W, D and C.
Axial decomposition Given an input slice of feature maps $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, the shape of X requires $C = kW$ which means width has to be a divisor of the number of channel once the number of parameter is fixed.
MLPP Given an input slice of feature maps $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, the shape of X requires $\mathrm{cd}(C, W) = L$, which only requires segment length L to be a common divisor of the number of channel and width, thus accepting a more flexible input shape. Additionally, by setting $g = C/L$, the number of parameters of our method does not depend on the shape of input feature maps.

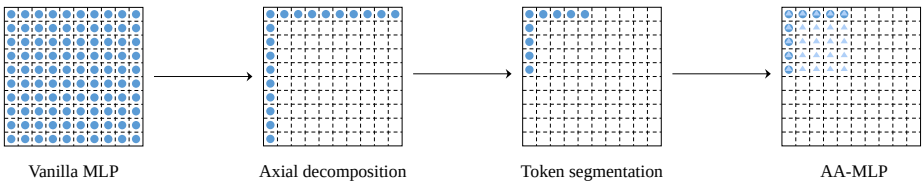

Fig. 5. Illustration of the sampling location of each operation. **Vanilla MLP** has a global receptive field. **Axial decomposition** could model long-range dependencies and preserve precise positional information. **Token segmentation** alleviates the resolution sensitivity problem. **AA-MLP** enlarges the receptive field of MLPP and enables tokens that are not in the same horizontal or vertical position to interact directly.

Algorithm 1 Pseudo code for MLPP (PyTorch-like)

```
# H: height, W: width, C: channel, D: depth, L: segment length
# x: input tensor of shape (H, W, D, C)

def init():
    g = C // L
    proj_h = Linear(C, C) # Encoding information along the vertical axis
    proj_w = Linear(C, C) # Encoding information along the horizontal axis
    proj_c = Linear(C, C) # Encoding channel information
    proj_d = Linear(C, C) # Encoding through-plane information
    attn = Linear(L * L, L * L) # Auxiliary Attention
    proj_1, proj_2, proj_3 = Linear(C, C), Linear(C, C), Linear(C, C) # Information fusion
    norm_1, norm_2, norm_3 = Layernorm(), Layernorm(), Layernorm()

def IP-MLP(x):
    x_att = x.reshape(H // L, L, W // L, L, D, C)
    x_att = x_att.permute(0, 2, 4, 5, 1, 3).reshape(H // L, W // L, D, C, L * L)
    x_att = attn(x_att).reshape(H // L, W // L, D, C, L, L)
    x_att = x_att.permute(0, 4, 1, 5, 2, 3).reshape(H, W, D, C)

    x_h = x.transpose(1,0).reshape(H * W // L, L, D, L, g)
    x_h = x_h.permute(0, 3, 2, 1, 4).reshape(H * W // L, L, D, L * g)
    x_h = proj_h(x_h).reshape(H * W // L, L, D, L, g)
    x_h = x_h.permute(0, 3, 2, 1, 4).reshape(W, H, D, C).transpose(1,0)

    x_w = x.reshape(H * W // L, L, D, L, g)
    x_w = x_w.permute(0, 3, 2, 1, 4).reshape(H * W // L, L, D, L * g)
    x_w = proj_w(x_w).reshape(H * W // L, L, D, L, g)
    x_w = x_w.permute(0, 3, 2, 1, 4).reshape(H, W, D, C)

    x_c = proj_c(x)

    x = x_h + x_w + x_c
    x = (1 + x_att) * x
    x = proj_1(x)
    return x

def TP-MLP(x):
    x_d = x.reshape(H, W, D // L, L, L, g)
    x_d = x.permute(0, 1, 2, 4, 3, 5).reshape(H, W, D // L, L, L * g)
    x_d = proj_d(x_d).reshape(H, W, D // L, L, L, g)
    x_d = x_d.permute(0, 1, 2, 4, 3, 5)
    x_d = proj_2(x_d)
    return x_d

def MLPP(x):
    x = x + IP-MLP(norm_1(x))
    x = x + TP-MLP(norm_2(x))
    x = x + proj_3(norm_3(x))
    return x
```

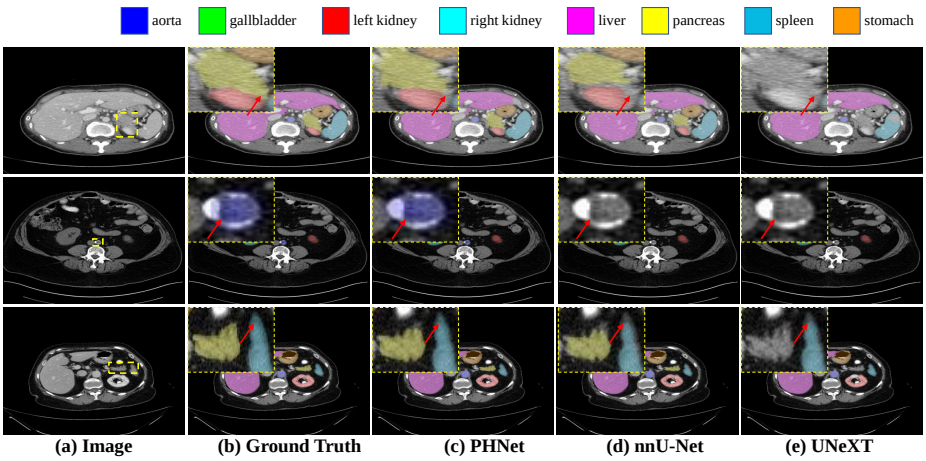

Fig. 6. Visualization results on Synapse. Regions of evident improvements are enlarged.