More than Encoder: Introducing Transformer Decoder to Upsample

Yijiang Li\textsuperscript{1,2}, Wentian Cai\textsuperscript{1,2}, Ying Gao\textsuperscript{1,2,3}\textsuperscript{3}, Chengming Li\textsuperscript{3} and Xiping Hu\textsuperscript{4}

\textsuperscript{1}School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
\textsuperscript{2}Guangdong Provincial Key Laboratory of Artificial Intelligence in Medical Image Analysis and Application, Guangdong Provincial People’s Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China
\textsuperscript{3}School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, China
\textsuperscript{4}School of Medical Technology, Beijing Institute of Technology, Beijing, China

\{csliyjiang3000, cscawai\}@mail.scut.edu.cn, gaoying@scut.edu.cn, lichengming@mail.sysu.edu.cn, huxp@bit.edu.cn

Abstract—Medical image segmentation methods downsample images for feature extraction and then upsample them to restore resolution for pixel-level predictions. In such schema, upsample technique is vital in restoring information for better performance. However, existing upsample techniques leverage little information from downsampling paths. The local and detailed feature from the shallower layer such as boundary and tissue texture is crucial in segmentation, especially medical image segmentation. To this end, we propose a novel upsample approach for medical image segmentation, Window Attention Upsample (W AU), which upsamples features conditioned on local and detailed features from downsampling path in local windows by introducing attention decoders of Transformer. WAU could serve as a general upsample method and be incorporated into any segmentation model that possesses lateral connections. We first propose the Attention Upsample which consists of Attention Decoder (AD) and bilinear upsample. AD leverages pixel-level attention to model long-range dependency and global information for a better upsample. Bilinear upsample is introduced as the residual connection to complement the upsamples. Moreover, considering the extensive memory and computation cost of pixel-level attention, we further design a window attention scheme to restrict attention computation in local windows instead of the global range. We evaluate our method (W AU) on classic UNet structure with computation in local windows instead of the global range. We view this decoding process where the output of encoders is conditioned on the last output tokens while also paying attention to multi-scale objects and dense pixels. Interestingly, we notice that transformer also possess an encoder and decoder. So, while most researchers focus on the encoder and explore its feature extracting ability, we instead look at the idea of the decoder in transformer and its applicability to segmentation architectures.

Index Terms—Transformer, upsampling, medical image segmentation, medical image analysis.

I. INTRODUCTION

Deep learning revolutionizes many fields of machine intelligence including multimedia processing \cite{15, 16, 25}, scene understanding \cite{22, 24}, and Computer Aided Diagnosis (CAD) \cite{2, 23} area. In CAD area, particularly, medical image segmentation plays a crucial role in clinical diagnosis and treatment processes. Long et al. \cite{12} proposes the famous FCN architecture which downsamples high resolution images to extract semantic information and then upsamples them to provide dense predictions. UNet \cite{17} extends it to a U-shape architecture with lateral connections between the downsampling and upsampling path. This architecture and its variants become dominant in medical image segmentation \cite{4, 8}. The encoder-decoder structure enlarges the receptive field making the Convolution Neural Network (CNN) better at capturing semantic information. Its pyramid structure enables the model to have multi-scale perception and reduces computation complexity. However, the reduction of resolution inevitably loses information, so maintaining semantic information while recovering the spatial resolution becomes challenging. To resolve this issue, multiple upsample techniques \cite{27}–\cite{29} have been proposed. However, existing upsample techniques leverage little information from downsampling path. The prosper of Transformer in the field of Natural Language Processing (NLP) inspires researchers to explore its applicability to Computer Vision (CV). ViT \cite{5} takes only the encoder of the transformer and obtains comparable results as CNN. Swin Transformer \cite{11} adopts and modifies the ViT architecture \cite{5} into the one that constructs a hierarchical representation with reduced computation. These works prove the adaptability of pure Transformer to CV downstream tasks such as object detection and segmentation which requires modeling over multi-scale objects and dense pixels. Interestingly, we notice that transformer also possess an encoder and decoder. So, while most researchers focus on the encoder and explore its feature extracting ability, we instead look at the idea of the decoder in transformer and its applicability to segmentation architectures.

A typical decoder in transformer takes the input token embedding of the last position to generate query and obtains the output from encoder to produce key and value \cite{20}. Given the circumstances of translation, the output of the decoder is conditioned on the last output tokens while also paying attention to the input sequence tokens. Intuitively, we can view this decoding process where the output of encoders is decoded conditioned on input token embedding. Notice that, the input token sequence may not be as long as the embedding from the encoder. Consequently, if the former is longer, the decoder outputs longer embedding. In a way, we can view it as being upsampled. Inspired by the above analogy, we propose a novel upsample approach, Window Attention Upsample (W AU), which upsamples features conditioned on local and detailed information from the downsampling path in
local windows. Attention upsampling can enrich the semantic information based on spatial and local information and still outputs features of desired larger shapes. Considering that large feature maps are unaffordable in global attention, we propose Window Attention Decoder (WAD) to trade-off between the global attention and computation expense. To further ease the learning, we use bilinear upsample to form a residual connection. To the best of our knowledge, we are the first to utilize the transformer decoder in segmentation upsampling and explore its ability to upsample feature maps and restore information. We evaluate our method on multiple classic segmentation structures with lateral connection and achieve state-of-the-art performance on Medical Segmentation Decathlon (MSD) Brain and Automatic Cardiac Diagnosis Challenge (ACDC) datasets. We also validate our method on multiple classic architectures and achieve consistent improvement. In a nutshell, contributions of our work can be summarized as follows:

- We propose the idea of upsampling images using the transformer decoder and provide an effective U-shaped architecture for medical image segmentation.
- We adopt window-based self-attention to better model pixel-level information and reduce computational cost and memory usage. To further exploit the potential, convolution projection is raised to model locality and residual connection through bilinear interpolation to complement the upsampled feature maps.
- Extensive experiments on different datasets using various architectures prove the effectiveness and the generalization ability of our Window Attention Upsample method.

II. METHOD

A. Decoder to Upsample

Decoder adopts the idea of dot product attention, much like encoder prevalent in recent work of transformer in vision. Unlike the patch encoder, our decoder attention acts on pixel-level instead of patch level in order to better model the dense information. So here, we refer to one pixel as one token. For the purpose of upsampling, we are majorly concerned about two factors: whether it can maintain or even enrich semantic information necessary for segmentation and whether it outputs feature maps of higher resolution. Transformer decoder inherently uses additional information (i.e. query token) to instruct the process of attention by imposing a larger weighting on tokens whose key are similar with query and a smaller weighting otherwise. In our Attention Decoder (AD), we use the feature maps of larger resolution from downsampling path to generate query and input features from upsampling path to generate key and value. In this way, larger resolution feature can be generated conditioned on rich information from downsampling path. This can be formulated as below:

\[ z^l = AD(LN(z^{l-1}), LN(\hat{a}^{(l)})) + z^{l-1} \quad (1) \]

where \( LN(\cdot) \) represents layer normalization. \( z^{l-1} \in \mathbb{R}^{H^{l-1} \times W^{l-1} \times C^{l-1}} \) denotes features of layer \( l-1 \) in upsampling path and \( \hat{a}^{(l)} \in \mathbb{R}^{H^{l} \times W^{l} \times C^{l}} \) denotes the corresponding feature maps from downsampling path.

\[ H^{l} = n \cdot H^{l-1}, W^{l} = n \cdot W^{l-1} \quad (2) \]

where \( H^{l}, W^{l} \) denotes the height and width of the feature map and \( n \) an integer larger than 1. By taking the context information from the downsampling path, decoder manages to model the global semantic information conditioned on corresponding low level features. Intuitively, context information will increase the weighting of relevant tokens that benefits the upsampling, so the semantic information from upsampling path can be maintained and even enriched.

B. Locality and Computation Considerations

Locality is an excellent property of CNN, which helps model the local features such as edges and corners. The reconstruction of higher resolution should focus more on neighboring regions. However, the transformer attends to all tokens deprived of the this good property. Despite its ability to model long-range dependency, transformer may lose focus on the significant and relevant tokens when there are numerous tokens, which is an essential problem in the pixel upsample process. Moreover, global attention among all tokens possess a quadratic complexity and memory usage with respect to the number of tokens, which is unaffordable for modeling pixel-level attention, especially at upper layers where resolution is high. To restrict the model’s attention in the local area and to reduce computation overhead, we propose to leverage convolution projection and local window attention as detailed in the following sections.

1) Introducing convolution to projection: To better model local information, we try to incorporate convolution into projection prior to attention block. We use a kernel of size larger than 1, typically 3 to replace the linear projection that is widely used in transformer attention block. In our paper, all convolutions use kernel of \( 3 \times 3 \) and maintains sizes (i.e. "same" padding). After the projection, three matrices, key (k), value (v) and query (q) are obtained and then flattened into 1D for subsequent multi-head attention process. Notice, since our input feature maps for query are larger than that of key and value, 1D query sequence are longer than key and value sequence. The output of decoder is the same size as query. After reshaping the output back to 2D, the resolution of the output are the same with feature maps from downsampling path. In this way, upsampling is done. The convolution projection can be written as follows:

\[ z_i^q = F(s_c^q \ast LN(\hat{a}_i)) \quad (3) \]

\[ z_i^{k/v} = F(s_c^{k/v} \ast LN(z_i)) \quad (4) \]

Here \( * \) denotes the convolution operator, \( s_c = [s_c^1, s_c^2, \ldots, s_c^{C_c}] \) where \( C_c \) is the number of output channels. \( z_i^q/k/v \) is the corresponding \( k, q, v \) matrices obtained and \( F(\cdot) \) denotes an operation that flattens 2D images into 1D sequence. Then we
apply dot attention on \( k, q, v \) and computes the upsampled feature maps:

\[
\hat{z}^l = s_c \ast \text{reshape}(\text{softmax}(\frac{z^l q z^l k^T}{\sqrt{d_k}}) z^l v)
\]  

(5)

Here, \( \text{reshape}(\cdot) \) denotes an operation that reshapes the 1D sequence back to 2D feature maps. Another convolution with kernel \( s_c \) is applied after the attention function.

2) Attention in Local Window: Inspired by [11], we propose local window attention for the attention decoder. Since self-attention works on one group of tokens, one window is enough. However, in WAD, we have tokens from two different resolution feature maps, so windows with different sizes are required to align the output key, value and query. Inherit from the preceding formulation, feature map from lateral connection \( \hat{a}^{(l)} \in \mathbb{R}^{H \times W \times C} \) is \( n \) times the size of that from upsampling path \( \hat{z}^{l-1} \in \mathbb{R}^{H^l-1 \times W^l-1 \times C^l-1} \). In order to align the number windows in query and key, value, windows sizes ratio between the two should also be \( n \). With window attention, our WAD can be formulated as below:

\[
\hat{z}^l = \text{WAD}(\text{LN}(\hat{z}^{l-1}), \text{LN}(\hat{a}^{(l)}))
\]  

(6)

3) Discussion: Swin transformer leverages the local window attention to save computation resources. Despite its low computation overhead, window attention limits the model’s long-range dependency and leads to a degraded performance [11]. That’s to say, larger window sizes generally leads to better performance [10]. To compensate for the loss of long-range dependency, Swin leverages shifted operation to increase the attention range. However, in this work, we discover that when attending to a large number of tokens (at pixel-level), the attention mechanism loses its focus and pays attention to irrelevant parts of the feature map [26] as we observe a drop in performance when using larger window sizes in Figure 3. To restrict attention in local areas, which is important for upsampling, window attention is used to confine the attention in local windows. We also conduct an ablation study by adding an additional shifted window attention layer before our upsampling module and observe an even lower performance (72.34 DSC compared with 73.65 DSC). This further demonstrates that simply enlarging receptive fields may be sub-optimal for upsampling.

C. Residual Connection Through Bilinear

In order to complement the features and form a residual-like operation, we propose to use bilinear interpolation to upsample and adds the two upsampled features together as output. This bilinear upsampled feature serves as a supplement as well as a residual connection that ease the training of WAD.

\[
\hat{z}^l = \text{WAD}(\text{LN}(\hat{z}^{l-1}), \text{LN}(\hat{a}^{(l)})) + \text{Bilinear}(\hat{z}^{l-1})
\]  

(7)

where \( \hat{z}^l \) is the output feature map of decoder upsample module \( l \) and \( \hat{a}^{(l)} \) are corresponding feature maps of twice the resolution from downsampling path.
III. EXPERIMENT

A. Dataset

We evaluate our model on the MSD Task01 BrainTumour dataset (MSD Brain) [19] and Automatic Cardiac Diagnosis Challenge (ACDC) datasets. MSD Brain contains 484 multimodal multisite MRI data (FLAIR, T1w, T1gd, T2w) and four labels including background, edema (Ed), non-enhancing tumor (NET), and enhancing tumor (ET). For MSD Brain, we apply z-scoring normalization to preprocess each case. To alleviate the problem of class imbalance, we remove all blank slices with zero values and crop each slice to the region of nonzero values. Each slice is cropped to 128×128 before feeding into the model. ACDC contains 100 cases of MRI scans from different patients whose goal is to segment the myocardium (Myo) of the left ventricle and the cavity of the right (RV) and left ventricle (LV). Following the settings of [1], 80 and 20 subjects are divided into training and validation set respectively with the resolution re-sampled to 160×160. Multiple data preprocessing techniques are done with the same settings of [14].

B. Implementation Details

For all experiments, we perform some slight data augmentation, e.g., random rotation, and horizontal and vertical flipping. For model invariant, to coincide with the typical U-Net structure, we set n = 2 meaning to upsample by 2 at each WAU module. We use a base window size of 4 for MSD Brain and 5 for ACDC dataset. All models are trained using Adam [9] with betas of 0.9 and 0.999 (default setting) and Cosine Annealing learning rate [13] with a warm up of 2 epochs. The initial learning rate is 0.0001 with a batch size of 12 for MSD Brain and 10 for ACDC. No pre-training is used and all experiments are conducted using two NVIDIA RTX2080Ti GPU.

C. Results

1) Results on MSD Brain Dataset: Results of our model and other state-of-the-art methods are shown in Table I. On the MSD BrainTumour Dataset, our model achieves the best performance of 74.75% Dice similarity coefficient (DSC) with 80.73%, 63.23%, and 80.29% on edema, non-enhancing tumor, and enhancing tumor respectively. When comparing with our baseline model ResUNet, we achieve a significant increase of 2.83%. Compared with nnUNet [8] 2D version, which also builds upon the U-Net architecture, our method obtains an improvement of 3.19%. Moreover, when compared with ensemble 3D nnUNet, we also outperform by 0.86%. We also make a comparison between state-of-art Transformer-based models including recent 3D network UNETR [6] and 2D network SwinUNet [3] which we outperform by a margin of 2.94% and 1.55% on average DSC respectively.

2) Results on ACDC Dataset: Table III demonstrates our model’s performance on ACDC dataset comparing with state-of-the-arts. On ACDC, we achieve more than 2 points improvement on a benchmark dataset with an average DSC of 90%+, which we consider quite tremendous on such dataset. It’s worth mentioning that after being carefully tuned, ResUNet is capable of achieving a performance of 90.06%, even higher than other state-of-the-arts such as SwinUNet and TransUNet. However, our model can outperform it by nearly 2 points in DSC.

D. Analytical study

1) Comparison with Baseline: We compare our WAD with different upsample methods and Table IV shows the performance of various upsample methods on different backbones, we can make the following observations: (i) Despite the difference in upsample method, the overall performance of ResUNet is better than that of classic U-Net, which is why we use ResUNet as the backbone. (ii) Bilinear interpolation

![Table I]

| Methods            | DSC ↑ | Ed  | NET | ET  |
|--------------------|-------|-----|-----|-----|
| FCN32s             | 60.40 | 70.03 | 46.96 | 64.99 |
| FCN16s             | 66.25 | 74.84 | 52.93 | 70.97 |
| FCN8s              | 69.21 | 76.61 | 56.17 | 74.83 |
| ResNet             | 67.65 | 75.03 | 57.87 | 70.06 |
| DeepLabV3          | 68.86 | 77.42 | 57.11 | 72.04 |
| TransUNet          | 71.11 | 77.38 | 59.04 | 76.91 |
| nnUNet (2D)        | 71.56 | 78.60 | 58.65 | 77.42 |
| TransBTS           | 71.79 | 78.62 | 60.14 | 76.61 |
| ResUNet            | 71.92 | 77.73 | 59.47 | 78.57 |
| SegTrans R50       | 73.48 | 80.20 | 61.81 | 78.42 |
| SwinUNet           | 73.20 | 79.41 | 61.38 | 73.20 |

![Table II]

| Methods          | Params | GFLOPs |
|------------------|--------|--------|
| ResUNet          | 17.2M  | 14.64  |
| wide-ResUNet     | 30.8M  | 26.08  |
| TransUNet        | 93.19M | 11.71  |
| SegTrans R50     | 129.82M| 53.47  |
| SwinUNet         | 27.12M | 5.49   |
| Ours             | 21.80M | 15.94  |

![Table III]

| Methods          | DSC ↑ | RV  | Myo | LV  |
|------------------|-------|-----|-----|-----|
| R50-UNet         | 87.55 | 87.10 | 80.63 | 94.92 |
| R50-AttUNet      | 86.75 | 87.58 | 79.29 | 93.47 |
| VIT-CUP          | 81.45 | 81.46 | 70.71 | 92.18 |
| R50-VIT-CUP      | 87.57 | 86.07 | 81.88 | 94.75 |
| TransUNet        | 90.05 | 90.14 | 86.00 | 94.00 |
| SwinUNet         | 90.00 | 88.55 | 85.62 | 95.83 |
| ResUNet          | 90.06 | 88.86 | 86.75 | 94.57 |
| Ours             | 92.00 | 91.65 | 88.95 | 95.40 |
is slightly better than the other three upsample methods, but the performance of our proposed W AU far exceeds them all, which suggests that the classic decoder design can be better replaced by our Window Attention Upsample (W AU) strategy. It is worth mentioning that we also compare the number of parameters and FLOPs used in ResUNet, TransUNet, Swin-UNet, SegTran R50, and our model (Table II). We suppose the better performance of TransUNet over ResUNet could be attributed to the large parameters. Our model, however, uses much fewer parameters (only 1/3 of TransUNet) and relatively acceptable operations to achieve much better performance on all datasets. To make a fair comparison with baseline ResUNet in terms of parameters, we increase the base channels from 64 to 72, resulting in the wide-ResUNet with more parameters and flops (30.82M and 26.08 GFLOPs). As per the third row of table IV, we can observe that our method still outperforms this improved baseline by more than 2 DSC. This demonstrates that the improvement is not the result of simply adding more parameters and flops.

### Table IV

| Methods    | Backbone | Upsample | DSC (%) |
|------------|----------|----------|---------|
| UNet       | Bilinear |          | 71.91 (+0.11) |
|            | Transposed |          | 71.80 |
| Wide-ResUNet | Bilinear |          | 72.14 |
| ResUNet    | Bilinear |          | 71.92 (+0.01) |
|            | Transposed |          | 71.85 (+0.54) |
|            | pixelShuffle [18] |          | 71.31 |
|            | CARAFE [21] |          | 71.63 (+0.32) |
|            | WAU      |          | 73.84 (+2.53) |
|            | W AU (W AD w/ Bilinear) |          | 74.75 (+2.83) |
| UNet 3D    | Bilinear |          | 72.15 |
|            | WAU      |          | 72.51 (+0.36) |
| DeepLab    | Bilinear |          | 68.86 |
|            | WAU      |          | 70.33 (+1.47) |
| FCN 32s    | Transposed |          | 60.40 |
|            | WAU      |          | 65.89 (+5.49) |
| FCN 16s    | Transposed |          | 66.25 |
|            | WAU      |          | 68.97 (+2.72) |
| FCN 8s     | Transposed |          | 69.20 |
|            | WAU      |          | 71.32 (+2.12) |

2) **Residual Connection through Bilinear:** We adopt Bilinear Interpolation to form a residual connection. To validate the effectiveness of such design, we perform an ablation study on this operation. From Table IV, we can see that adding Bilinear Interpolation increases DSC by 0.79, which sufficiently proved the effectiveness of residual connection through Bilinear Interpolation.

3) **Convolution Matters:** In Section II-B1, we introduce the convolution projection to obtain the key, query, and value matrices. In this section, we explore the performance of different convolution operations. Results in Figure 3 reveal that Depthwise Separable convolution is slightly better than the other two convolution operations with a window size of 4. This could be attributed to the fewer parameters possesses by the Depthwise Separable Convolution which provides better performance on a relatively small dataset.

4) **Ablation study on Window Size:** We conduct ablation study on different window sizes with W AD and ResUNet architecture on MSD Brain dataset. As per Figure 3, we find that the optimal window size is 4. Increasing the window size leads to a drop in the performance. We hypothesize that the model might lose focus when the window size is too large and this is particularly problematic for upsampling as it depends on the detailed and local features in neighboring regions. Using too small a window, on the other hand, also degrades the performance since it gets rid of the long-range dependency.

5) **Generalizability of W AU with Different Architecture:** We demonstrate that our proposed W AU can be incorporated into any architecture that possesses lateral connections. By incorporating our method into different architectures, we observe consistent improvements in all experiments as shown in table IV. Specifically, we incorporate WAU into UNet 3D and observe an improvement of 0.36 DSC. This demonstrates that our method can be used in 3D volume segmentation which comprises a large category of medical segmentation methods.

---

**E. Visualization**

In this part, we provide visualization of Window Attention weights and the upsampled feature maps of different models in the upsampling path. To obtain our Window Attention weights, we retrieve the attention weights in W AD. Since each attention is computed inside local windows, we select the activated regions (the positive region in ground truth) and show the average attention weights of these windows with positive pixels. Figure 4 is the visualization of our Window Attention weights and shows how the attention is focused on the relevant pixels of the target area in each window. This further demonstrates that by imposing an attention, model is prone to focus more on target. This enriches the information needed for segmentation task and leads to better performance.
usage in upsample. Our work proves that decoder can also be adopted to model visual information and performs even better than traditional upsample techniques. To leverage the ability of such architecture, we propose our Window Attention Upsample that reconstruct semantic pixels to desired shape conditioned on local and detailed information. With this, we provide a better alternative to the basic upsample operation and can be fused in any segmentation model that requires upsample. Moreover, our work partly exploits the possibility of adopting a pure transformer with encoder and decoder into CV.

ACKNOWLEDGMENT

This work is jointly supported by Guangdong Provincial Key Laboratory of Artificial Intelligence in Medical Image Analysis and Application (No. 2022B1212010011) and the International Cooperation Project of Guangdong Province under grants 2021A0505030017.

REFERENCES

[1] C. F. Baumgartner, L. M. Koch, M. Pollefeys, and E. Konukoglu, “An exploration of 2d and 3d deep learning techniques for cardiac mr image segmentation,” in Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges, M. Pop, M. Sermesant, P-M. Jodoin, A. Lalande, X. Zhuang, G. Yang, A. Young, and O. Bernard, Eds. Cham: Springer International Publishing, 2018, pp. 111–119.

[2] W. Cai, L. Xie, W. Yang, Y. Li, Y. Gao, and T. Tung, “Dftnet: Dual-path feature transfer network for weakly supervised medical image segmentation,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, pp. 1–12, 2022.

[3] 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,” arXiv preprint arXiv:2105.05337, 2021.

[4] 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:2102.04306 [cs], Feb. 2021, arXiv:2102.04306. [Online]. Available: http://arxiv.org/abs/2102.04306

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[6] A. Hatamizadeh, D. Yang, H. Roth, and D. Xu, “UNETR: Transformers for 3D Medical Image Segmentation,” arXiv:2103.10504 [cs, eess], Mar. 2021, arXiv:2103.10504. [Online]. Available: http://arxiv.org/abs/2103.10504

[7] Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, “Dints: Differentiable neural network topology search for 3d medical image segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5841–5850.

[8] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Noraqijtra, S. Wirkert, and K. H. Maier-Hein, “mnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation,” arXiv:1809.10486 [cs], Sep. 2018, arXiv:1809.10486. [Online]. Available: http://arxiv.org/abs/1809.10486

[9] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[10] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring plain vision transformer backbones for object detection,” arXiv preprint arXiv:2203.16527, 2022.

[11] 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,” arXiv preprint arXiv:2103.14030, 2021.