LambdaUNet: 2.5D Stroke Lesion Segmentation of Diffusion-Weighted MR Images

Yanglan Ou[1(✉)], Ye Yuan[2], Xiaolei Huang[1], Kelvin Wong[3], John Volpi[4], James Z. Wang[1], and Stephen T. C. Wong[3]

[1] The Pennsylvania State University, University Park, PA, USA
yanglanou@psu.edu
[2] Carnegie Mellon University, Pittsburgh, PA, USA
[3] TT and WF Chao Center for BRAIN & Houston Methodist Cancer Center, Houston Methodist Hospital, Houston, TX, USA
[4] Eddy Scurlock Comprehensive Stroke Center, Department of Neurology, Houston Methodist Hospital, Houston, TX, USA

Abstract. Diffusion-weighted (DW) magnetic resonance imaging is essential for the diagnosis and treatment of ischemic stroke. DW images (DWIs) are usually acquired in multi-slice settings where lesion areas in two consecutive 2D slices are highly discontinuous due to large slice thickness and sometimes even slice gaps. Therefore, although DWIs contain rich 3D information, they cannot be treated as regular 3D or 2D images. Instead, DWIs are somewhere in-between (or 2.5D) due to the volumetric nature but inter-slice discontinuities. Thus, it is not ideal to apply most existing segmentation methods as they are designed for either 2D or 3D images. To tackle this problem, we propose a new neural network architecture tailored for segmenting highly discontinuous 2.5D data such as DWIs. Our network, termed `LambdaUNet`, extends UNet by replacing convolutional layers with our proposed `Lambda+` layers. In particular, `Lambda+` layers transform both intra-slice and inter-slice context around a pixel into linear functions, called lambdas, which are then applied to the pixel to produce informative 2.5D features. `LambdaUNet` is simple yet effective in combining sparse inter-slice information from adjacent slices while also capturing dense contextual features within a single slice. Experiments on a unique clinical dataset demonstrate that `LambdaUNet` outperforms existing 3D/2D image segmentation methods including recent variants of UNet. Code for `LambdaUNet` is available. (URL: https://github.com/YanglanOu/LambdaUNet.)

Keywords: Stroke · Lesion segmentation · Inter- and intra-slice context · 2.5-dimensional images

1   Introduction

In the United States, stroke is the second leading cause of death and the third leading cause of disability [9]. About 795,000 people in the US have a stroke each year [12]. A stroke happens when some brain cells suddenly die or are damaged due to lack of oxygen when blood flow to parts of the brain is lost or reduced due to blockage or rupture of an artery [14]. Locating the lesion areas where brain tissue is prevented from getting oxygen and nutrients is essential for accurate evaluation and timely treatment. Diffusion-weighted imaging (DWI) is a commonly performed magnetic resonance imaging (MRI) sequence for evaluating acute ischemic stroke and is sensitive in detecting small and early infarcts [11].
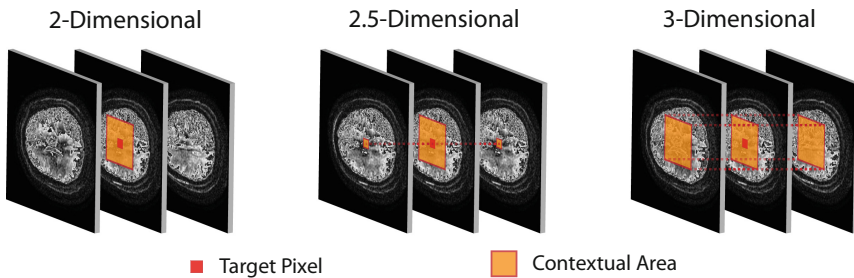

Fig. 1. Comparison of 2D, 2.5D, and 3D feature extraction methods. When extracting features for a target pixel, our 2.5D method restricts the context area in adjacent slices to focus on the most relevant pixels to reduce noise and improve generalization.

Segmenting stroke lesions on DWIs manually is time-consuming and subjective [10]. With the advancement of deep learning, numerous automatic segmentation methods based on deep neural networks (DNNs) have emerged to detect stroke lesions. Some of them perform segmentation on each 2D slice individually [2,4], while others treat DWIs as 3D data and apply 3D segmentation networks [19]. Beyond methods for lesion segmentation in DWIs, there have been many successful methods for general medical image segmentation. For instance, UNet [16] has shown the advantage of skip-connections on biomedical image segmentation. Based on UNet, Oktay *et al.* proposed Attention UNet by adding attention gates that filter the features propagated through the skip connections in U-Net [13]; Chen *et al.* proposed TransUNet, as they find that transformers make strong encoders for medical image segmentation [3]. Çiçek [5] extend UNet to 3D field for volumetric segmentation. Wang *et al.* proposed volumetric attention combined with Mask-RCNN to address the GPU memory limitation of 3D U-net. Zhang *et al.* [19] proposed a 3D fully convolutional and densely connected convolutional network which is derived from the powerful DenseNet [8].

Although previous medical image segmentation methods work well for 2D or 3D data by design, they are not well suited for DWIs, which have contextual characteristics between 2D and 3D. We term such data type as 2.5D [18].[1] Different from 2D data, DWIs contain 3D volumetric information by having multiple DWI slices. However, unlike typical 3D medical images that are isotropic or near isotropic in all three dimensions, DWIs are highly anisotropic with slice dimension at least five times more than in-plane dimensions. Therefore, neighboring slices can have abrupt changes around the same area which is especially problematic for early infarcts that are small and do not extend beyond a few slices. Due to the 2.5D characteristics of DWIs, if we apply 2D segmentation methods to DWIs, we lose valuable 3D contextual information from neighboring slices (Fig. 1 (left)). On the other hand, if we apply a traditional 3D CNN-based segmentation method, due to the high discontinuity between slices, many irrelevant features from neighboring slices are processed by the network (Fig. 1 (right)), which adds substantial noise to the learning process and also makes the network prone to over-fitting.

In this work, our goal is to design a segmentation network tailored for images with 2.5D characteristics like DWIs. To this end, we propose `LambdaUNet` which adopts the UNet [16] structure but replaces convolutional layers with our proposed `Lambda+` layers which can capture both dense intra-slice features and sparse inter-slice features effectively. `Lambda+` layers are inspired by the Lambda layers [1] which transform both global and local context around a pixel into linear functions, called lambdas, and produce features by applying these lambdas to the pixel. Although Lambda layers have shown strong performance for 2D image classification, they are not suitable for 2.5D DWIs because they are designed for 2D data and cannot capture sparse inter-slice features. Our proposed `Lambda+` layers are designed specifically for 2.5D DWI data, where they consider both the intra-slice and inter-slice contexts of each pixel. Here the inter-slice context of a pixel consists of pixels at the same 2D location but in neighboring slices (Fig. 1 (middle)). Note that, unlike many 3D feature extraction methods, `Lambda+` layers do not consider pixels in neighboring slices that are at different 2D locations, because these pixels are less likely to contain relevant features and we suppress them to reduce noise and prevent over-fitting. `Lambda+` layers transform the inter-slice context into a different linear function–inter-slice lambda–which complements other intra-slice Lambdas to derive sparse inter-slice features. As illustrated in Fig. 1, the key design of `Lambda+` layers is that they treat intra-slice and inter-slice features differently by using a dense intra-slice context and a sparse inter-slice context, which suits well the 2.5D DWI data.

Existing works in 2.5D segmentation [7,17,20] also recognize the anisotropy challenge of CT scans. However, they simply combine 3D and 2D convolutions without explicitly considering the anisotropy. To our knowledge, the proposed `LambdaUNet` is the first 2.5D segmentation model that is designed specifically for 2.5D data like DWIs and treats intra-slice and inter-slice pixels differently. Exten-

---

[1] Note that our definition of 2.5D is different from that in computer vision, where 2.5D means the 2D retinal projections of 3D environments.

sive experiments on a large annotated clinical DWI dataset of stroke patients show that `LambdaUNet` significantly outperforms previous art in terms of segmentation accuracy.

2   Methods

Denote a DWI volume as $\boldsymbol{I} \in \mathbb{R}^{T \times H \times W \times C}$, where T is the number of DWI slices, H and W are the spatial dimensions (in pixels) of each 2D slice, respectively, and C is the number of DWI channels. The DWI volumes are preprocessed by skullstripping to remove non-brain tissues in all the DWI channels.

Our goal is to predict the segmentation map $\boldsymbol{O} \in \mathbb{R}^{T \times H \times W}$ of stroke lesions. The spatial resolution within each slice is 1 mm between adjacent pixels while the inter-slice resolution is 6 mm between slices. We can observe that the inter-slice resolution of DWIs is much lower than the intra-slice resolution, which leads to the high discontinuity between adjacent slices—the main characteristic of 2.5D data like DWIs. As discussed in Sect. 1, both 3D and 2D segmentation models are not ideal for DWIs, because common 3D models are likely to overfit irrelevant features in neighboring slices, while 2D models completely disregard 3D contextual information. This motivates us to propose the `LambdaUNet`, a 2.5D segmentation model specifically designed for DWIs. Below, we will first provide an overview of `LambdaUNet` and then elaborate on how its `Lambda+` layers effectively capture 2.5D contextual features.

LambdaUNet. The main structure of our `LambdaUNet` follows the UNet [16] for its strong ability to preserve both high-level semantic features and low-level details. The key difference of `LambdaUNet` from the original UNet is that we replace convolutional layers in the UNet encoder with our proposed `Lambda+` layers (detailed in Sect. 2.1), which can extract both dense intra-slice features and sparse inter-slice features effectively. Since all layers except `Lambda+` layers in `LambdaUNet` are identical with those in UNet, they require 2D features as input; we address this by merging the slice dimension T with the batch dimension to reshape 3D features into 2D features for non-`Lambda+` layers, while `Lambda+` layers undo this reshaping to recover the slice dimension and regenerate a 3D input that is used to extract both intra- and inter-slice features. The final output of `LambdaUNet` is the lesion segmentation mask $\boldsymbol{O} \in \mathbb{R}^{T \times H \times W}$. The Binary Cross-Entropy (BCE) loss is used to train `LambdaUNet` for the pixel-wise binary classification task.

2.1   Lambda+ Layers

`Lambda+` layers are an enhanced version of Lambda layers [1], which transform context around a pixel into linear functions, called *lambdas*, and mimic the attention operation by applying lambdas to the pixel to produce features. Different from attention, the lambdas can encode positional information as we will elaborate later, which affords them a stronger ability to model spatial relations.

`Lambda+` layers extend Lambda layers, which are designed for 2D data, by adding inter-slice lambdas with a restricted context region to effectively extract features from 2.5D data such as DWIs.

The input to a `Lambda+` layer is a 3D feature map $\boldsymbol{X} \in \mathbb{R}^{|n| \times |c|}$, where $|c|$ is the number of channels and n is the linearized pixel index into both spatial (height H and width W) and slice (T) dimensions of the feature map, $i.e.$, n iterates over all pixels $\mathcal{P}$ inside the 3D volume, and $|n|$ equals the total number of pixels $|\mathcal{P}|$. Besides input $\boldsymbol{X}$, we also have context $\boldsymbol{C} \in \mathbb{R}^{|m| \times |c|}$ where $\boldsymbol{C} = \boldsymbol{X}$ (same as self-attention) and m also iterates over all pixels $\mathcal{P}$ in the 3D volume. Importantly, when extracting features for each pixel n, we restrict the region of context pixels m to a 2.5D area $\mathcal{A}(n) \subset \mathcal{P}$. As shown in Fig. 2 (a), the 2.5D context area consists of the entire slice where pixel n is in, as well as pixels with the same 2D location in adjacent $\mathcal{T}$ slices where $\mathcal{T}$ is the inter-slice kernel size.

Similar to attention, `Lambda+` layer computes queries $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q \in \mathbb{R}^{|n| \times |k|}$, keys $\boldsymbol{K} = \boldsymbol{C}\boldsymbol{W}_K \in \mathbb{R}^{|m| \times |k| \times |u|}$, and values $\boldsymbol{V} = \boldsymbol{C}\boldsymbol{W}_V \in \mathbb{R}^{|m| \times |v| \times |u|}$, where $\boldsymbol{W}_Q \in \mathbb{R}^{|c| \times |k|}$, $\boldsymbol{W}_K \in \mathbb{R}^{|c| \times |k| \times |u|}$ and $\boldsymbol{W}_V \in \mathbb{R}^{|c| \times |v| \times |u|}$ are learnable projection matrices, $|k|$ and $|v|$ are the dimensions of queries (keys) and values, and $|u|$ is an additional dimension to increase model capacity. We normalize the keys across pixels using softmax: $\bar{\boldsymbol{K}} = \mathrm{softmax}(\boldsymbol{K})$. We denote $\boldsymbol{q}_n \in \mathbb{R}^{|k|}$ as the n-th query in $\boldsymbol{Q}$ for a pixel n. We also denote $\bar{\boldsymbol{K}}_m \in \mathbb{R}^{|k| \times |u|}$ and $\boldsymbol{V}_m \in \mathbb{R}^{|v| \times |u|}$ as the m-th key and value in $\boldsymbol{K}$ and $\boldsymbol{V}$ for a context pixel m.

For a target pixel $n \in \mathcal{P}$ inside a slice t, a `lambda+` layer computes three types of lambdas (linear functions) as illustrated in Fig. 2: (1) a **global lambda** that encodes global context within slice t, (2) a **local lambda** that summarizes the local context around pixel n in slice t, and (3) an **inter-slice lambda** that captures inter-slice features from adjacent slices.
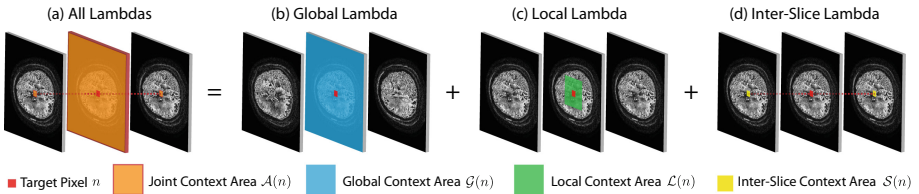

Fig. 2. Context areas of the global lambda, local lambda, and inter-slice lambda.

Global Lambda. As shown in Fig. 2(b), the global lambda aims to encode the global context within slice t where the target pixel n is in, so the context area $\mathcal{G}(n)$ of the global lambda includes all pixels within slice t. For each context pixel $m \in \mathcal{G}(n)$, its contribution to the global lambda is computed as:

$$\boldsymbol{\mu}_m^{\mathtt{G}} = \bar{\boldsymbol{K}}_m \boldsymbol{V}_m^T, \quad m \in \mathcal{G}(n). \tag{1}$$

The global lambda $\boldsymbol{\lambda}_n^{\mathtt{G}}$ is the sum of the contributions from each pixel $m \in \mathcal{G}(n)$:

$$\boldsymbol{\lambda}_n^{\mathtt{G}} = \sum_{m \in \mathcal{G}(n)} \boldsymbol{\mu}_m^{\mathtt{G}} = \sum_{m \in \mathcal{G}(n)} \bar{\boldsymbol{K}}_m \boldsymbol{V}_m^T \in \mathbb{R}^{|k| \times |v|} . \tag{2}$$

Note that $\boldsymbol{\lambda}_n^{\mathtt{G}}$ is invariant for all n within the same slice as $\mathcal{G}(n)$ is the same.

Local Lambda. The local lambda encodes the context of a local $R \times R$ area $\mathcal{L}(n)$ centered around the target pixel n in slice t (see Fig. 2(c)). Compared with the global lambda, besides the difference in context areas, the local lambda uses *learnable* relative-position-dependent weights $\boldsymbol{E}_{nm} \in \mathbb{R}^{|k| \times |u|}$ to encode the position-aware contribution of a context pixel m to the local lambda:

$$\boldsymbol{\mu}_{nm}^{\mathtt{L}} = \boldsymbol{E}_{nm} \boldsymbol{V}_m^T , \quad m \in \mathcal{L}(n) . \tag{3}$$

Note that the weights $\boldsymbol{E}_{nm}$ are shared for any pairs of pixels (n, m) with the same relative position between n and m. The local lambda $\boldsymbol{\lambda}^{\mathtt{L}}$ is obtained by:

$$\boldsymbol{\lambda}_n^{\mathtt{L}} = \sum_{m \in \mathcal{L}(n)} \boldsymbol{\mu}_{nm}^{\mathtt{L}} = \sum_{m \in \mathcal{L}(n)} \boldsymbol{E}_{nm} \boldsymbol{V}_m^T \in \mathbb{R}^{|k| \times |v|} . \tag{4}$$

Inter-Slice Lambda. The inter-slice lambda defines a context area $\mathcal{S}(n)$ including pixels in adjacent slices sharing the same 2D location with the target pixel n, as shown in Fig. 2(d). As discussed before, we use this restricted context area for extracting inter-slice features due to the high discontinuity between slices for 2.5D data like DWIs. Although one context pixel per adjacent slice seems very restrictive, one pixel of a feature map at coarse (downsampled) 2D scales in `LambdaUNet` corresponds to a large area in the original scale. Furthermore, `LambdaUNet` employs multiple `Lambda+` layers, so information from other pixels in adjacent slices can first propagate to pixels in $\mathcal{S}(n)$ and then to the target pixel n. Thus, our design of the restricted context area makes the network focus on the most-informative pixels inside $\mathcal{S}(n)$ and suppress less-relevant pixels, while still allowing long-range interactions as pixels outside the area can indirectly contribute to the feature through multiple `Lambda+` layers.

Similar to the local lambda, the inter-slice lambda $\boldsymbol{\lambda}_n^{\mathtt{S}}$ uses learnable weights $\boldsymbol{F}_{nm} \in \mathbb{R}^{|k| \times |u|}$ to encode position-aware contribution of context pixels:

$$\boldsymbol{\mu}_{nm}^{\mathtt{S}} = \boldsymbol{F}_{nm} \boldsymbol{V}_m^T , \quad m \in \mathcal{S}(n) , \tag{5}$$

$$\boldsymbol{\lambda}_n^{\mathtt{S}} = \sum_{m \in \mathcal{S}(n)} \boldsymbol{\mu}_{nm}^{\mathtt{S}} = \sum_{m \in \mathcal{S}(n)} \boldsymbol{F}_{nm} \boldsymbol{V}_m^T \in \mathbb{R}^{|k| \times |v|} . \tag{6}$$

Applying Lambdas. After computing the global lambda $\boldsymbol{\lambda}_n^{\mathtt{G}}$, local lambda $\boldsymbol{\lambda}_n^{\mathtt{L}}$, and inter-slice lambda $\boldsymbol{\lambda}_n^{\mathtt{S}}$, we are ready to apply them to the query $\boldsymbol{q}_n$ of the target pixel n. The output feature $\boldsymbol{y}_n$ for the target pixel n is:

$$\boldsymbol{y}_n = \boldsymbol{q}_n^T \left(\boldsymbol{\lambda}_n^{\mathtt{G}} + \boldsymbol{\lambda}_n^{\mathtt{L}} + \boldsymbol{\lambda}_n^{\mathtt{S}} \right) \in \mathbb{R}^{|v|} . \tag{7}$$

The final output of `Lambda+` layer is a 3D feature map $\boldsymbol{Y} \in \mathbb{R}^{|n| \times |v|}$ formed by the output features $\boldsymbol{y}_n$ of all pixels $n \in \mathcal{P}$. Although the above procedure for computing lambdas is for a single pixel n, we can easily parallelize the computation for all pixels using standard convolution operations, which makes `Lambda+` layers computationally efficient. We refer readers to the pseudocode in the supplementary materials for detailed implementation.

3   Experiments

The primary focus of our experiments is to answer the following questions: (1) Does `LambdaUNet` predict lesion segmentation maps more accurately than baselines? (2) Is our 2.5D `Lambda+` layer more effective than the 2D or 3D Lambda layer? (3) Based on qualitative results, does `LambdaUNet` has clinical significance?

Dataset. The clinical data we use to evaluate our model is provided by an urban academic hospital. We sampled 99 acute ischemic stroke cases with large ($n = 42$) and small ($n = 57$) infarct size. The data has an equal distribution of samples from stroke with the left or right middle cerebral artery (MCA), posterior cerebral artery (PCA), and anterior cerebral artery (ACA) origins. The cases contain a mix of 1.5T and 3.0T scans. Certain cases even have a mix of MCA and ACA. The ischemic infarcts are manually segmented by three experts based on diffusion-weighted imaging (DWI) (b = $1000\,\mathrm{s/mm}^2$) and the calculated exponential apparent diffusion map (eADC) using MRIcro v1.4. We use the eADC and DWI images from ischemic stroke patients to form the two channels of input DWIs $\boldsymbol{I}$. We use 67 of the 99 fully labeled cases for training and the remaining 32 fully labeled cases for validation and testing. More specifically, we split the 32 cases into three folds of roughly the same size. Two of the three folds are used for validation and one remaining fold is used for testing. Each of the three folds is used for testing once, and the average result is reported as the final testing result. The 32 cases used for testing were carefully chosen to make sure the stroke size, location, and type are nicely balanced in the testing set.

Implementation Details. Our implementation is using the PyTorch [15] and the Lightning [6] frameworks. All experiments are conducted using four NVIDIA Quadro RTX 6000 GPUs with 24 GB memory. For `Lambda+` layers, both the inter-slice kernel size $\mathcal{T}$ and the local kernel size R are set to 3. We train the model for 100 epochs using the RMSprop optimizer; an initial learning rate of `1e-4` is used for 20 epochs and then the learning rate is linearly reduced to 0. We randomly select 12 DWI sequence segments of 8 slices to form a mini-batch during training. The whole training process takes about 4 h to finish. The training converges after 40 epochs. For testing, we select the model that gives the highest dice score for validation data.

Baselines and Metrics. We compare our method against well-known and recent 2D segmentation methods, U-Net [16], AttnUNet [13], and TransUNet [3], as well as one 3D segmentation method: 3D UNet [5]. All the baseline methods are reproduced based on their open-sourced code with careful hyperparameter

tuning. Besides, we also report the results of two variants of `LambdaUNet` to further evaluate the effectiveness of the proposed 2.5D lambda+ layer. We use four common evaluation metrics—dice score coefficient (DSC), recall, precision, and F_1 score—for stroke lesion segmentation to provide quantitative comparisons.

3.1   Results

In the first group of Table 1, we show the slice-level accuracy of all baselines on our stroke lesion dataset. One can observe that the proposed `LambdaUNet` has significant improvements over baselines, *e.g.*, performance gains range from 3.06% to 8.31% for average DSC. The improvement suggests that our `Lambda+` layers are more suitable for feature extraction of 2.5D DWI data. In the second group, we compare `LambdaUNet` with its 2D and 3D variants. `LambdaUNet2D` directly removes the inter-slice lambda from the `LambdaUNet` while `LambdaUNet3D` uses a 3D local context area $\mathcal{L}(n)$ instead of the inter-slice lambda. As indicated in Table 1, both variants perform worse than `LambdaUNet` in terms of DSC and the F_1 score. This demonstrates the effectiveness of the 2.5D design of the proposed `Lambda+` layers. Although `LambdaUNet` does not achieve the highest precision or recall over the baselines and variants, it can maintain a good balance between recall and precision, which sometimes cancel each other out (*e.g.*, AttnUNet and 3D UNet). This is further confirmed by the superior F_1 score of `LambdaUNet`.

Table 1. Segmentation performance comparison between different models.

Method	2D/3D	DSC	Recall/Precision	F_1 Score
UNet [16]	2D	82.15	80.28/86.29	81.61
AttnUNet [13]	2D	81.83	77.45/86.74	80.82
TransUNet [3]	2D	83.45	83.24/87.15	84.48
3D UNet [5]	3D	78.20	**83.54**/78.39	78.21
(Ours) LambdaUNet-2D	2D	84.03	82.27/87.10	84.19
(Ours) LambdaUNet-3D	3D	84.76	79.92/**89.86**	84.09
(Ours) LambdaUNet	2.5D	**86.51**	81.76/89.39	**84.84**

Figure 3 visualizes the predicted segmentation masks on five consecutive slices for one stroke case. We can see that the masks produced by our `LambdaUNet` (last column) are the closest to the ground truth than the baselines. For instance, in slice 3 (S3), the baselines either miss some details (UNet, AttnUNet, TransUNet) indicated by the white areas or generate some false positive predictions (3D UNet) denoted by the red areas, while our `LambdaUNet` captures the irregular shape of lesions well. S4 and S5 also show that `LambdaUNet` performs the best on difficult small lesions. More results are provided in the supplementary materials.
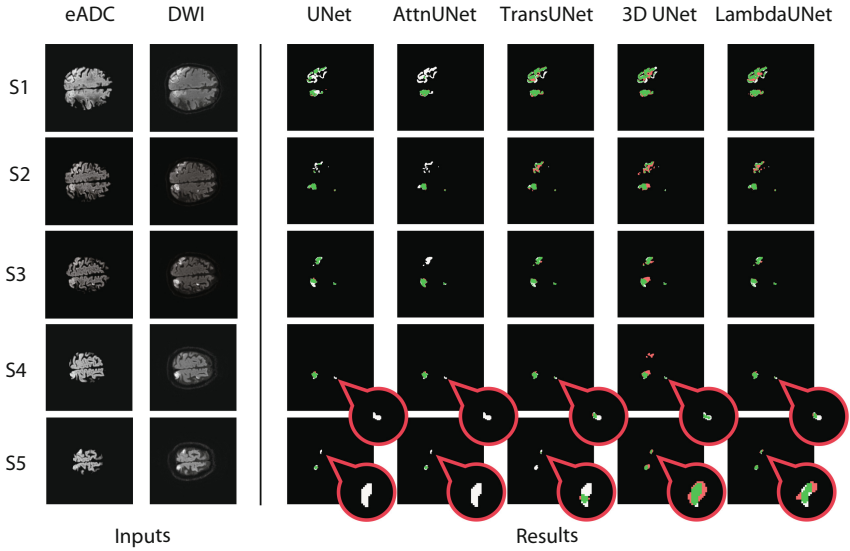
Fig. 3. Qualitative results on five consecutive slices of one ischemic stroke clinical case. Green indicates the correct predictions. White areas are false negative while red areas are false positive. Red circles show a close-up view of the lesion areas. (Color figure online)

3.2 Discussion

Our `LambdaUNet` not only shows advantages on both quantitative and qualitative measurements, the way it extracts features is more like clinicians. As clinicians consider all adjacent slices but only focus on the most informative areas, our `Lambda+` layers capture intra- and inter-slice features and automatically suppress irrelevant 3D interference. Lesion areas of acute stroke are an important end-point for clinical trials, as proper treatment relies on measuring the infarction core volume and estimating salvageable tissue. Therefore, an accurate and reproducible DWI-suited segmentation model like `LambdaUNet` will be of high interest in clinical practice.

4 Conclusion

We defined DWIs as 2.5D data for their dense intra-slice resolution and sparse inter-slice resolution. Based on the 2.5D characteristics, we proposed a segmentation network `LambdaUNet`, which includes a new 2.5D feature extractor, termed `Lambda+` layers. `Lambda+` layers effectively capture features in 2.5D data by using dense intra-slice and sparse inter-slice context areas. This design allows the network to focus on informative features while suppressing less relevant features to reduce noise and improve generalization. Experiments on the clinical stroke dataset verify that our `LambdaUNet` outperforms state-of-the-art segmentation methods and shows strong potential in clinical practice.

References

1. Bello, I.: LambdaNetworks: modeling long-range interactions without attention. In: Proceedings of the International Conference on Learning Representations (2021). https://openreview.net/forum?id=xTJEN-ggl1b

2. Charoensuk, W., Covavisaruch, N., Lerdlum, S., Likitjaroen, Y.: Acute stroke brain infarct segmentation in DWI images. Int. J. Pharma Med. Biological Sci. **4**(2), 115–122 (2015)

3. Chen, J., et al.: TransUNet: transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)

4. Chen, L., Bentley, P., Rueckert, D.: Fully automatic acute ischemic lesion segmentation in DWI using convolutional neural networks. NeuroImage: Clinical **15**, 633–643 (2017)

5. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49

6. Falcon, W., et al: PyTorch Lightning. GitHub 3 (2019). https://github.com/PyTorchLightning/pytorch-lightning

7. Han, L., Chen, Y., Li, J., Zhong, B., Lei, Y., Sun, M.: Liver segmentation with 2.5D perpendicular UNets. Comput. Electr. Eng. **91**, 107118 (2021)

8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)

9. Johnson, W., Onuma, O., Owolabi, M., Sachdev, S.: Stroke: a global response is needed. Bull. World Health Organ. **94**(9), 634 (2016)

10. Kanchana, R., Menaka, R.: Ischemic stroke lesion detection, characterization and classification in CT images with optimal features selection. Biomed. Eng. Lett. **10**, 333–344 (2020)

11. Lansberg, M.G.: Diffusion-weighted MRI in Acute Stroke. Utrecht University Dissertation (2002)

12. Mozaffarian, D., Benjamin, E.J., Go, A.S., Arnett, D.K., Blaha, M.J., Cushman, M., Das, S.R., De Ferranti, S., Després, J.P., Fullerton, H.J., et al.: Heart disease and stroke statistics–2016 update: a report from the American Heart Association. Circulation **133**(4), e38–e360 (2016)

13. Oktay, O., et al.: Attention U-Net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 (2018)

14. Owolabi, M.O., et al.: The burden of stroke in Africa: a glance at the present and a glimpse into the future. Cardiovascular J. Africa **26**(2 H3Africa Suppl), S27 (2015)

15. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 (2019)

16. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

17. Zhang, C., Hua, Q., Chu, Y., Wang, P.: Liver tumor segmentation using 2.5D UV-Net with multi-scale convolution. Comput. Biology Med. **133**, 104424 (2021)

18. Zhang, H., Valcarcel, A.M., Bakshi, R., Chu, R., Bagnato, F., Shinohara, R.T., Hett, K., Oguz, I.: Multiple sclerosis lesion segmentation with tiramisu and 2.5D stacked slices. In: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.-T., Khan, A. (eds.) MICCAI 2019. LNCS, vol. 11766, pp. 338–346. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32248-9_38
19. Zhang, R., Zhao, L., Lou, W., Abrigo, J.M., Mok, V.C., Chu, W.C., Wang, D., Shi, L.: Automatic segmentation of acute ischemic stroke from DWI using 3-D fully convolutional DenseNets. IEEE Trans. Med. Imaging **37**(9), 2149–2160 (2018)
20. Zhou, D., et al.: Eso-Net: a novel 2.5D segmentation network with the multi-structure response filter for the cancerous esophagus. IEEE Access **8**, 155548–155562 (2020)