Retinal Layer Segmentation in OCT images with Boundary Regression and Feature Polarization

Yubo Tan, Wen-Da Shen, Ming-Yuan Wu, Gui-Na Liu, Shi-Xuan Zhao, Yang Chen, Kai-Fu Yang, and Yong-Jie Li, *Senior Member, IEEE*

***Abstract*— The geometry of retinal layers is an important imaging feature for the diagnosis of some ophthalmic diseases. In recent years, retinal layer segmentation methods for optical coherence tomography (OCT) images have emerged one after another, and huge progress has been achieved. However, challenges due to interference factors such as noise, blurring, fundus effusion, and tissue artifacts remain in existing methods, primarily manifesting as intra-layer false positives and inter-layer boundary deviation. To solve these problems, we propose a method called Tightly combined Cross-Convolution and Transformer with Boundary regression and feature Polarization (TCCT-BP). This method uses a hybrid architecture of CNN and lightweight Transformer to improve the perception of retinal layers. In addition, a feature grouping and sampling method and the corresponding polarization loss function are designed to maximize the differentiation of the feature vectors of different retinal layers, and a boundary regression loss function is devised to constrain the retinal boundary distribution for a better fit to the ground truth. Extensive experiments on four benchmark datasets demonstrate that the proposed method achieves state-of-the-art performance in dealing with problems of false positives and boundary distortion. The proposed method ranked first in the OCT Layer Segmentation task of GOALS challenge held by MICCAI 2022. The source code is available at `https://www.github.com/tyb311/TCCT`.**

***Index Terms*— OCT, retinal layer segmentation, vision transformer, feature polarization, boundary regression.**

I. Introduction

THE clinical evaluation of optic neuropathy requires a thorough analysis of retinal nerve fiber layer (RNFL) thickness via optical coherence tomography (OCT) imaging [3]. Ophthalmologists typically measure the thickness of the retinal layers to identify the loss of axonal fiber bundles for various eye diseases, such as glaucoma [4], compression optic neuropathy, optic neuritis, and others, which can lead to permanent blindness. Accurate segmentation of OCT biomarkers is an indispensable prerequisite for measuring the geometric features of OCT layers [5]. Therefore, developing automated, intelligent, and highly-effective segmentation algorithms for OCT layers is of critical importance for the intelligent imaging diagnosis of ophthalmic diseases.

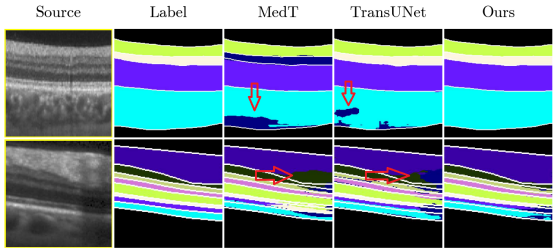

Fig. 1. Challenging cases of OCT layer segmentation, including false positives due to image blurring and noise (top) and boundary deviation due to artifact of fundus tissues (bottom) indicated by arrows. The proposed method can greatly optimize the contour of the retinal layers in the segmentation results and inhibit false positives, compared with the recent state-of-the-art methods, e.g., MedT [1] and TransUNet [2].

OCT segmentation plays a pivotal role in ophthalmic image diagnosis, and the accuracy of segmentation directly influences the measurement of parameters and diagnosis of diseases. Research in this field has spanned a wide range, from the early traditional unsupervised segmentation methods such as contour iteration [6], [7] and graph theory based segmentation [8], [9] methods, to machine learning [10], [11] and neural network approaches [5], [11]. The contour iteration methods try to gradually search for the complete contours of the retinal layers based on the initialized boundaries. The graph theory based methods aim to map the image to a graph structure and then identify the edges and nodes where the boundaries are located. Combining topological order and shape priors of the retinal layers can further optimize the segmentation results [12]. The machine learning methods attempt to automatically learn and classify which layer the pixels belong to by extracting and analyzing local image features [10]. Recent deep learning methods are designed to leverage massive data to achieve more efficientive segmentation of retinal layers [5].

This work was supported by STI2030-Major Projects (#2022ZD0204600), Sichuan Science and Technology Program (#2022ZYD0112), Medico-Engineering Cooperation Funds from UESTC (#ZYGX2022YGRH013) and in part by NSFC (#62076055). (Corresponding authors: Yong-Jie Li and Kai-Fu Yang).

Yubo Tan, Shi-Xuan Zhao, Kai-Fu Yang, and Yong-Jie Li are with the MOE Key Laboratory for Neuroinformation, Radiation Oncology Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu 610054, China. E-mails: tyb@std.uestc.edu.cn, zhaosx@std.uestc.edu.cn, yangkf@uestc.edu.cn, liyj@uestc.edu.cn.

Wen-Da Shen is with Changchun University of Science and Technology, Changchun, 130022, China. Ming-Yuan Wu is with Taizhou Institute of Science and Technology, Nanjing University of Science and Technology, Taizhou, 225306, China. Gui-Na Liu and Yang Chen are with West China Hospital, Sichuan University, Chengdu, 610044, China. E-mails: 13137657719@163.com, 18368131672@163.com, cutelgn@163.com, Sophie-0627@163.com.

However, due to the characteristics of OCT images such as speckle noise, intensity variance, blurring, and artifacts of biological tissues such as effusion and blood vessels, the task of retinal layer segmentation in OCT images is highly challenging. In particular, existing methods have yet to address the two serious issues of intra-layer segmentation false-positives and inter-layer boundary deviation in the segmentation results, as illustrated in Fig. 1. To tackle these problems, we present a novel hybrid deep neural network, TCCT-BP, which involves a novel exploration of network structure design and feature constraint paradigms. The main contributions are as follows.

- To maintain the integrity of retinal layer segmentation as a whole, we propose a hybrid architecture that integrates the vision Transformer (ViT) and convolutional neural network (CNN). Specifically, a multi-scale hierarchical Transformer is utilized to capture the long-distance spatial relationships of the retinal layers, while the cross-convolutional network backbone is used to enable the model to perform both global and local perception.
- To address the false positive challenge of intra-layer segmentation, the concept of feature polarization is introduced. By constraining the feature sampling and feature polarization loss, the feature distances of different categories of layers are maximized to enhance the robustness of prediction results within layers.
- To optimize the retinal layer boundaries, a boundary regression loss is designed to improve the accuracy of boundary continuity and authenticity.
- The results on an international OCT layer segmentation challenge and multiple datasets show that the proposed method achieves state-of-the-art performance.

II. Related Works

A. CNN based Segmentation Methods

It has been observed that classical deep learning models for medical image processing are primarily based on CNNs. For instance, UNet is a popular encoder-decoder deep network with the encoder capturing the context of medical images and the symmetric decoder accurately localizing objects [13]. Many networks have been modified based on UNet. For example, UNet++ [14] incorporated dense skip connection paths in the encoder and decoder, aiming to reduce the semantic gap between the encoder and decoder sub-networks. Furthermore, U^2Net [15] used a two-level embedding pattern to capture more context information, adding only a small computational overhead. nnU-Net is an out-of-the-box tool that configures data preprocessing, network selection, model training, and post-processing to provide state-of-the-art segmentation [16]. To summarize, U-Net like models have achieved tremendous success in medical image segmentation tasks. However, U-Net lacks the ability to explicitly model long-range dependencies, which is a critical limitation that many subsequent algorithms strive to overcome [17]. Different from others, we propose a dual backbone network based on CNN and ViT to compensate for the limited global receptive field of CNN and better model the narrow retinal layer structure.

B. ViT based Segmentation Methods

Convolution operators have inherent local properties and inductive biases, which make it challenging to explicitly model or understand long-distance dependencies in images [1]. Transformer, as a sequence-to-sequence prediction framework, is increasingly used as an alternative due to its built-in global self-attention mechanism [2]. Recently, many medical image segmentation networks based on Transformer architecture have been proposed, combining the idea of modeling long-distance relationships with self-attention. For example, Chen *et al.* proposed TransUNet, which uses Transformer instead of CNN encoder in UNet to encode the tokenized image patches from the input sequence to extract global contexts for medical image segmentation [2]. In addition, Cao *et al.* used hierarchical Swin Transformer with shifted windows in feature upsampling of decoders to better recover the spatial distribution of feature maps [18]. However, self-attention can overlook potential correlations in the entire dataset, and to deal with this problem, Wang *et al.* proposed a Mixed Transformer Module to simulate inter- and intra-affinities learning [17]. Additionally, in order to relieve the difficulty that self-attention structure requires a large amount of training data, Valanarasu *et al.* proposed a new control mechanism, namely gated axial-attention, to improve the efficiency of self-attention on small medical image datasets [1]. To enable cross-scale global information interaction of images with low computational complexity, Lin *et al.* proposed a cross-scale global transformer that uses multiple small-scale feature maps to extract richer global features [19]. Although ViT is a powerful medical image segmentation backbone that can capture global information well, it has limited localization ability due to insufficient operations capturing low-level details [2]. To address this limitation, we propose a dual backbone feature fusion strategy that combines CNN and ViT to complement each other and provide better receptive field characteristics.

C. Retinal Layer Segmentation Methods

1) Traditional Methods: Previous studies have made great efforts to explore traditional methods for retinal layer segmentation in OCT images. Chiu *et al.* utilized graph theory and dynamic programming for automatic and generalized segmentation of retinal layers [20]. Karri *et al.* introduced structured learning to empower traditional graph-based retinal layer segmentation, which can be integrated into any graph-based segmentation technique [21]. Lang *et al.* built a random forest classifier to accurately segment eight retinal layers in macular cube images acquired by OCT [22]. Generally, These traditional methods rely on the careful design of algorithms and parameter fine-tuning, making them inadequate for dealing with the increasing demand for massive data growth.

2) Deep Learning Methods: Most of the recent OCT layer segmentation algorithms are deep networks. Roy *et al.* proposed ReLayNet, which employed a contracting path of convolutional blocks to learn a hierarchy of contextual features, followed by an expansive path of convolutional blocks for semantic segmentation [12]. He *et al.* introduced SR-Net that extracts continuous, smooth, and topology-guaranteed surfaces

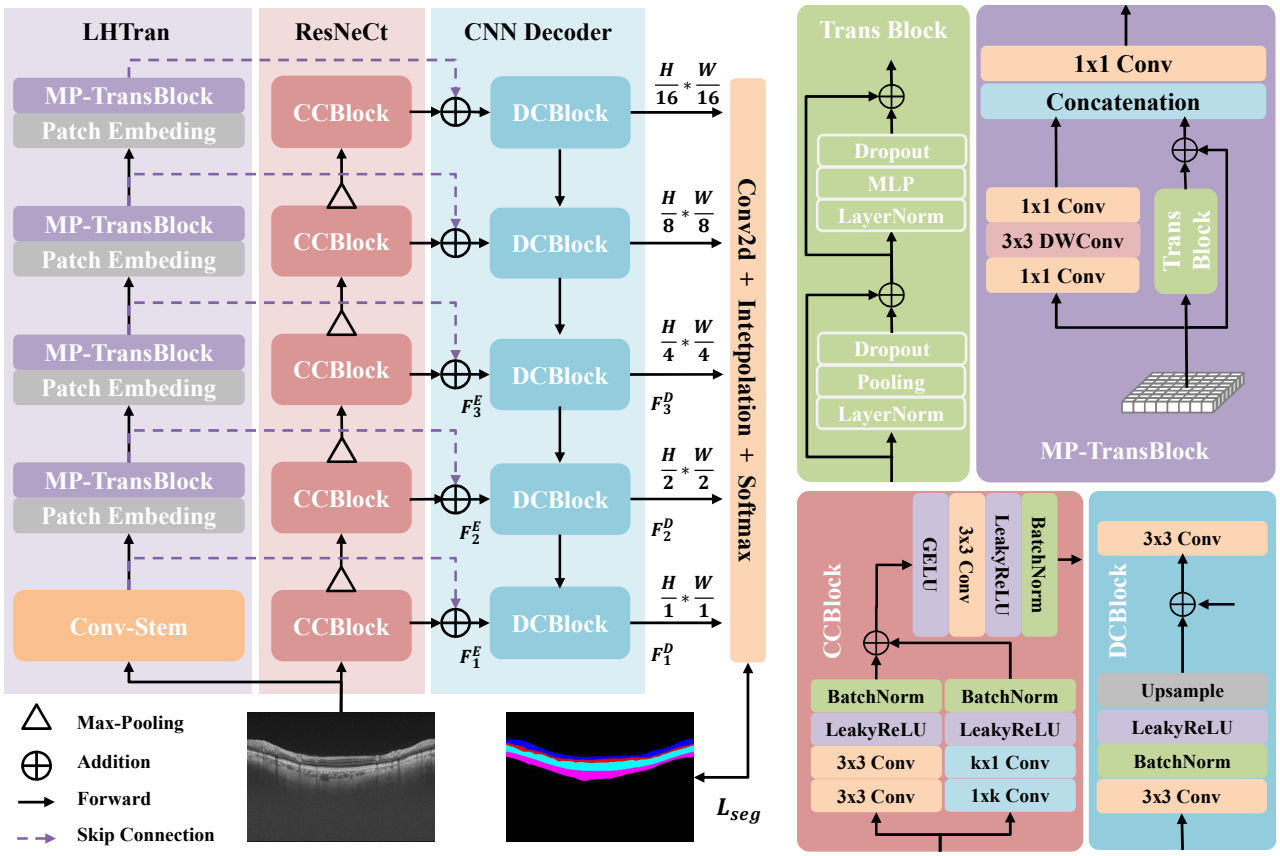

Fig. 2. The network architechture of TCCT. There are two backbones with different structures, LHTran and ResNeCt. LHTran is a lightweight hierarchical transformer structure, while ResNeCt is a residual network. The features extracted in parallel from the two backbones are fused at each level to produce retinal layer segmentation results.

[23]. SR-Net learned shape priors automatically during training rather than being hard-coded as in graph methods. Zhang *et al.* presented a biomarker-infused global-to-local network for choroid segmentation [24]. Ngo *et al.* developed a deep neural network that takes the intensity, gradient, and adaptive normalized intensity score of an image segment as features for learning [11]. Li *et al.* tackled the intricate anatomy of the peripapillary regions of the retina by introducing a novel graph convolutional network (GCN)-assisted two-stage framework [25]. Wang *et al.* proposed a boundary-aware U-Net by incorporating an edge-aware module and a canny edge fusion module [26]. He *et al.* combined the two steps into a unified deep learning framework by directly modeling the distribution of the surface positions [5]. Jeihouni *et al.* designed a multi-stage and multi-discriminatory generative adversarial network for super-resolution and segmentation of the retinal layers [27]. Despite the promising performance of FCN-based methods, the negative impact of the class imbalance problem on the segmentation of small foreground targets such as macular cystoid edemas remains a challenge [28]. CAZAÑAS-GORDÓN *et al.* introduced a novel architecture that leverages spatial and channel-attention gates at multiple scales for fine-grained segmentation and a weighting loss approach to handle class imbalance [28].

These newly-developed methods have achieved remarkable improvements in robustness and accuracy. Nevertheless, due to the limited number of images in OCT datasets, deep networks could not deal with complex retinal tissue, various imaging noises, and indistinct boundaries. To tackle these issues, our model is based on the hybrid architecture of CNN and Vision Transformer (ViT) [29]. CNN can learn accurate local boundary details, while ViT can make up for the shape distortion caused by local noise or artifact interference. In addition to applying the boundary coordinate regression loss, we also introduce spatial boundary regression to constrain the boundary positions of the retinal layers. Furthermore, due to the lack of consideration of tissue continuity, point-by-point prediction leads to an abundance of false positives. To counter this, the proposed approach introduces feature polarization loss to reduce the number of false positives by inferring the overall shape of the retinal layers and maximizing the distances between different categories.

III. Methodology

The proposed TCCT-BP (i.e., Tightly combined Cross-Convolution and Transformer with Boundary regression and feature Polarization) consists of a dual backbone network called TCCT, as illustrated in Fig. 2, and the corresponding loss functions. The first backbone is a lightweight hierarchical Transformer (LHTran) which is used to capture the long-distance dependencies of OCT layers. The other backbone is a residual network with cross-convolution (ResNeCt) to

capture local details. OCT images are first fed to LHTran and ResNeCt, and at each scale the extracted global and local features are then concatenated and sent to the decoder (5 DCBocks) to generate the segmentation probability maps of different layers. During the training stage, in addition to the deeply supervised segmentation loss (L_{seg}) for constraining the segmentation, the feature polarization loss (L_{fpl}) is designed to reduce false positive prediction, and the boundary regression loss (L_{brl}) is used to optimize segmentation boundaries.

The challenge of boundary distortion in retinal layers demands the proposed model with larger receptive fields in order to reduce the effect of local distortion. To this end, TCCT incorporates cross-convolution to increase the receptive field size of the CNN backbone. The fuzzy nature of fluid accumulation introduces difficulties in pinpointing its location, while the long distance dependence of ViTs allows for inference of retinal layer deformations caused by fluid accumulation. Along with the new network structure, TCCT utilizes the loss L_{brl} to optimize the retinal layer boundary regression in the training process, given its high effectiveness in regression tasks [12], [30]. Furthermore, the loss L_{fpl} is introduced to reduce the intra-class feature distance and increase the inter-class feature distance with the assistance of supervised contrastive learning, which also helps reduce false positive problems and effusion segmentation. In addition, in order to reduce the overhead of training and reasoning as much as possible, we attempt to use less parameters to achieve high segmentation performance, especially for ViTs which normally have high computational complexity.

A. Residual Learning with Cross-Convolution

The backbone network (ResNeCt) based on CNN consists of 5 CCBlocks. CNN is known to be effective at processing high-frequency details [31], making them particularly suitable for medical image processing with high resolution requirements. To increase the effective receptive field of common convolutions for a larger perceptual range with minimal computational cost, we incorporate cross convolution into the CCBlock. As shown in Fig. 2, in each CCBlock, the input features are reaggregated after a plain branch (with 3×3 Convolution) and a cross branch (with k×1 and 1×k cross Convolution). The hyperparameter k involved in CCBlock is set to [3,5,7,9,11] at 5 different scales of the backbones from the shallower to the deeper. To keep the network lightweight, the numbers of channels in all convolutional layers except the input and output layers are set to 32. The 5 CCBlocks in ResNeCt are connected by Max-Pooling layers in the middle. Assuming that the dimension of the input OCT image is H×W, the dimension of the multi-scale output features corresponding to each CCBlock is successively halved.

B. Lightweight Hierarchical Transformers

Due to its unique structure, ViT is believed to have the ability to capture global context of images naturally [31], and are playing an increasingly important role in medical image processing [1], [2]. To better learn the shapes of OCT layers, a lightweight hierarchical Transformer (LHTran) is designed as the second backbone of the TCCT-BP. LHTran is modified from a multi-path vision Transformer (MPViT) [29], with a few changes. LHTran preserves the multi-scale processing capability, allowing the backbone to process OCT images of different sizes and extract multi-level features. However, to reduce the high computational complexity of self-attention, we replace the self-attention of MPViT with an Avg-Pooling layer, as the general architecture is the most essential factor to the performance of Transformers, instead of the self-attention [32]. Like the ResNeCt, the parameters of Transformers are also reduced to maximize the lightweight inference of LHTran. The Conv-Stem with two Conv-BN layers is placed as the shallowest block to avoid detail loss due to Transformer patches. The embedded dimension of each layer of Transformer is [64,96,128,160], the multi-path is reduced to single path, and the number of heads of attention is set to 4.

The dual backbone fusion of CNN and Transformers is adopted as it is found that the performance of parallel fusion may be the best compared with the serial and cross-fusion methods of Transformers and CNN [31].

C. Multi-Class Feature Polarization

In order to address the topological segmentation error (i.e., false positives) in OCT layers, a feature polarization loss has been designed based on the principle of contrastive learning [33], [34], which states that the feature vectors of the same categories should be similar while the feature vectors of different categories should be mutually exclusive. Such contrastive learning can be divided into two stages: self-supervision learning (SSL) in the training stage, and supervised learning in the initial stage, such as target contrastive learning (TCL) [35]. TCL determines the prototype vectors of each category before training and maps the features to the prototypes of the corresponding categories in the hyperplane during the training phase, in order to reduce the long tail or false positive problems in classification tasks. Here TCL is extended to our feature polarization learning (FPL) for the task of segmentation. TCL and FPL require the relative contrast between features and prototype vectors, while SSL focuses on the contrast between features only. FPL is composed of three steps, as shown in Fig. 3.

The first step is the construction of prototypes for each category before the training stage. The goal is to generate the targets to which the features of different classes will be mapped. For the two-class classification, it is like generating two poles of a magnet. For the multi-class classification, there are multiple poles, and it is better that all poles are evenly distributed across the hypersphere. For the prototypes $\{t_i\}$ of C OCT layers, the prototype construction loss is

$$\mathcal{L}_{pcl} = \frac{1}{C}\sum_{i=1}^{C} log \sum_{j=1}^{C} e^{t_i^T \cdot t_j/\tau_1} \tag{1}$$

where $\tau_1 = 1$ is the temperature, $\{*\}^T$ is matrix transpose. By minimizing $\mathcal{L}_{pcl}$, the multi-class prototypes can be evenly distributed across the hypersphere as much as possible. The dimension of the prototype vector is set to 32 to ensure the

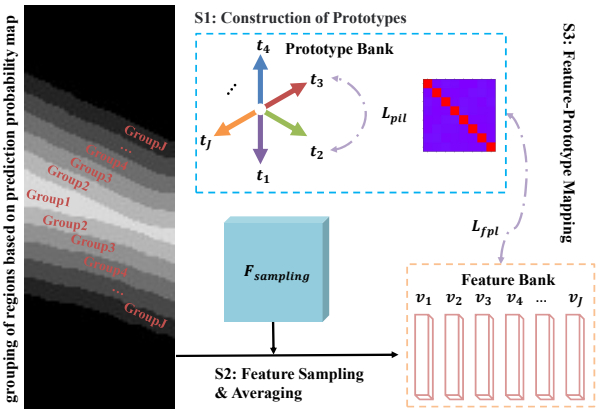

Fig. 3. Illustration of feature grouping, sampling and mapping. The regions with similar confidence in the probability map (i.e., the final output of TCCT-BP) are divided into the same groups. The features of each group are averaged to obtain the sampled features. The sampled vectors in the feature bank are mapped to the prototypes constrained by the feature polarization loss function.

convergence of the optimization process of prototypes. The optimized prototypes $\{t_i\}$ form the prototype bank, as shown in Fig. 3.

The second step is the feature sampling of the grouped regions during the training stage. We collect the three shallowest output features (i.e., F_i^E and F_i^D) of the encoder (i.e., ResNeCt and LHTran) and decoder (i.e., 5 DCBlocks) respectively, map them to the hyperplane and sum them.

$$F_{sampling} = \frac{1}{6}\sum_{i=1,2,3}(||F_i^E||_c + ||F_i^D||_c) \tag{2}$$

where $||\cdot||_c$ is the Euclidean norm in the channel dimmension. The feature for sampling $F_{sampling}$ is first morphed into $F^{(B\times H\times W)L}$, where B is the batch size, the number of channels L=32, H and W are the height and width of $F_{sampling}$. For the i-th OCT layer, the corresponding regions in the prediction probability map P is uniformly divided into J=32 groups according to the prediction confidence and the ground truth G, and the feature $F^{(B\times H\times W)L}$ is divided into J groups in the same order. For example, features with confidence in $[0,\frac{1}{J}]$ belong to the first group. The features in each group are averaged to obtain corresponding feature vectors $\{v_i\}$, and the J feature vectors make up a feature bank.

The third step is feature-prototype mapping. The vectors in the feature bank are mapped to the prototypes of the corresponding category in the prototype bank. The feature polarization loss function to optimize the mapping consistency is

$$\mathcal{L}_{fpl} = \sum_{i=1}^{J}(||v_i - t_i||_2 - \frac{||v_i \cdot t_i||_2}{||v_i||_2 \cdot ||t_i||_2}) \tag{3}$$

where $||*||_2$ is the mean square.

D. Layer Boundary Regression

In addition to the problem of false positives, OCT images also face serious boundary uncertainty caused by boundary blurring. In order to optimize the boundary segmentation, the boundary regression constraint mechanism [12] is introduced into the proposed method, namely the boundary regression loss. The corresponding boundary positions of C OCT layers, namely P_i^{sp} can be estimated by sampling-argmax [30], which can better constrain the shape of probability map compared with soft-argmax [12]. The output of the model can be regarded as the marginal conditional distributions of layers [5]. For the pixel at (h,i), there is a coordinate weighting coefficient

$$m_i(h) = \frac{exp((\eta + logP_{i,h})/\tau_2)}{\sum_{k=1}^{H} exp((\eta + logP_{i,k})/\tau_2)}. \tag{4}$$

where P and G are the prediction probability map and the ground truth respectively, and the teperature $\tau_2 = 0.01$ is set to make the distribution of adjuested P identical to one-hot, and the noise η follows the multivariate Gumbel(0, 1) distribution.

$$P_i^{sp} = \sum_{h=1}^{H} h \cdot m_i(h). \tag{5}$$

The loss of edge constraint includes the coordinate sampling regression term and the longitudinal gradient term. The first term is used to regress the boudary positions layer by layer, which is computed as

$$\mathcal{L}_{brl-pos} = \sum_{c=1}^{C} ||P_c^{sp} - \underset{dim=1}{argmax}(G_c)||_2 \tag{6}$$

while the second term is used to constrain the gradient edge of all layers at once

$$\mathcal{L}_{brl-grad} = ||\sigma(CC(P)) - abs(\nabla_v(G))||_2 \tag{7}$$

where CC is a processing with two Convolution layers, σ is Sigmoid, abs computes the absolution, and ∇_v computes the vertical gradient. And the total boundary regression loss is

$$\mathcal{L}_{brl} = \mathcal{L}_{brl-pos} + \mathcal{L}_{brl-grad} \tag{8}$$

It is true that boundary regression algorithms based on sampling-argmax [30] and soft-argmax [12] have been used for retinal layer segmentation. However, different from [30] and [12], our work realizes the boundary regression from two perspectives, i.e., sampling-argmax is used to conduct the layer-by-layer one-dimensional coordinate position boundary regression ($\mathcal{L}_{brl-pos}$), and the boundary gradient regression is carried out at the two-dimensional spatial scale ($\mathcal{L}_{brl-grad}$). The combination of these two types of regression helps improve the accuracy of the segmented boundaries.

In addition to $\mathcal{L}_{fpl}$ and $\mathcal{L}_{brl}$, the Dice loss is included in order to maximize the degree of coincidence between the prediction probability map and the ground truth.

$$\mathcal{L}_{seg} = \sum_{s}^{S}\sum_{c}^{C}(1 - \frac{2|P_{s,c} \cap G_{s,c}|}{|P_{s,c}| + |G_{s,c}|}) \tag{9}$$

here S=5 and C are the depth of the backbones and the number of layers of the ground truth, respectively. The total loss function is

$$\mathcal{L}_{all} = \lambda_{seg}\mathcal{L}_{seg} + \lambda_{fpl}\mathcal{L}_{fpl} + \lambda_{brl}\mathcal{L}_{brl} \tag{10}$$

where the coefficients of different loss functions are set to λ_{seg}=1, λ_{fpl}=1, and λ_{brl}=0.1.

IV. Experiments

TABLE I
INFORMATION OF THE EMPLOYED DATASETS (SD: SPECTRAL-DOMAIN, SS: SWEPT-SOURCE, DME: DIABETIC MACULAR EDEMA, MS: MULTIPLE SCLEROSIS, /: NOT AVAILABLE).

Dataset	Duke DME	HEG	HCMS	GOALS
Scanner device	Heidelberg SD-OCT	Heidelberg SD-OCT	Heidelberg SD-OCT	TOPCON DRI SS-OCT
Disease	DME	healthy	MS	glaucoma
Original size	496×768	496×768	496×1024	1100×800
Size for training	224×500	256×672	128×1024	512×496
Number of layers	7 & Fluid	7	8	3
Number of subjects	10	10	35	16
Samples per scan	11	10	49	/
Number of images	110	100	1715	100
Training / Test	88/22	50/50	735/980	50/50
Augmentation ratio	8.4	14.7	1.0	14.7

A. Materials

To fairly evaluate the performance of various segmentation algorithms, four OCT segmentation datasets are selected, namely Duke DME [36], HEG [37], HCMS [38], and GOALS [39]. The details of each dataset are as follows.

Duke DME[1]: The Duke DME dataset contains 10 diabetic macular edema (DME) patients, each of which has 11 B-scans from the Duke Eye Center Medical Retina. All of the 110 B-scans are with the same resolution of 496×768, scanned on a Spectralis HRA+OCT scanner. Each scan was labeled with seven layers. Following BAU-Net [26], the first 8 subjects with 88 B-scans are for training and the left 2 subjects with 22 B-scans for test.

HEG[2]: There are 10 Spectralis SD-OCT volume data from 10 healthy adult subjects (Heidelberg Engineering GmbH, Heidelberg, Germany) in the HEG dataset. Each volume contains 10 B-scans with a size of 496×768 pixels. Seven retinal layers are manually delineated for each B-scan. The first 50 images of the five people are converted into the training set, and the remaining half images are divided into the test set.

HCMS[3]: The HCMS dataset is from 14 healthy controls (HC) and 21 people with multiple sclerosis (MS), with a totally of 49 B-scans scanned on a Heidelberg Spectralis OCT device. There are 9 surfaces manually delineated in each scan, with an image size of 496×1024. Following He *et al.* 's work [5], the models are trained on the last 6 of the HC volumes and the last 9 of the MS volumes (15 volumes for training) and tested on the remaining 20 volumes.

GOALS[4]: Glaucoma OCT Analysis and Layer Segmentation (GOALS) Challenge was held in conjunction with the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) 2022, to provide data for researchers studying OCT layer segmentation (Task 1) and the classification of glaucoma (Task2). The 300 circumpapillary OCT images with a resolution of 1100×800 are from Zhongshan Ophthalmic Center, Sun Yat-sen University, China, acquired by a TOPCON DRI Swept Source OCT device. We divide this dataset in two ways. For one thing, the training set and the test set each have 100 images, referring to the competition setup, the annotations contain 3 layers. For another thing, since the annotations of the validation set and the test set are not available, we divided the 100 annotated images of the training set in half (i.e., the first 50 images for training and the rest 50 images for test, which are publicly available) to compare the performance of different methods, and we set the area between GCIPL and Choroid as a new layer to test the recognition ability of successive retinal layers.

In this paper, frequently-used medical image segmentation metrics such as Dice, IoU, Hough surface distance (HD), and Euclidean surface distance (ED) are adopted. Dice and IoU are indicators of overlap degree. HD and ED measures the maximum and average boundary errors, which are expressed in pixels in the original image size in this work.

B. Details

Our model is built based on the Pytorch framework, and all experiments are carried out on the TITAN RTX (12GB) GPU. The optimizer is AdamW with a learning rate of 1e-3 and a weight decay of 5e-4. If the loss does not decrease in three consecutive epochs, the learning rate decays to 0.8 times of the original rate until the learning rate decreases to 1e-5, and the cooling period is three epochs.

To reduce the computational overhead, the images are resized for training mainly using their regions of interest by cutting the surrounding blank regions of original images, as indicated by "Size for training" in Table I. While in the test phase, the images are predicted in their original sizes. The used image preprocessing methods include zooming, cropping, and zero filling. During the training stage, the OCT images are padded and cropped into patches with a fixed size of 256×256, then normalized and input into the network. For example, an image of size 224×512 is firstly extended to 256×512 with zero padding, then cropped to size of 256×256 for training. The batch size is set to 32. All the test sets were solely for test, not serving as part of validation set. During the training phase, we kept tracking the Dice of the model and chose the weight corresponding to the highest Dice on the training set, and then evaluated its performance on the test set, as commonly did in many works [16], [34]. All the training phases were terminated after 100 epochs.

Considering the large difference of sample numbers among the multiple datasets employed in this study, the samples in the training sets are augmented with different degrees (i.e., the augmentation ratio). As shown in Table I, the aim of augmentation is to enlarge the numbers of training samples for each dataset to 735, the number of training samples of HCMS. Thus, the augmentation ratio for each dataset is calculated as 735 divided by the number of images in its corresponding training set (e.g., 8.4≈735÷88 for Duke DME).

[1] https://people.duke.edu/~sf59/Chiu_BOE_2014_dataset.htm
[2] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0133908
[3] https://iacl.ece.jhu.edu/index.php?title=Resources
[4] https://aistudio.baidu.com/aistudio/competition/detail/230

TABLE II

ABLATION STUDY ON DUKE DME. RED IS THE BEST. (↓: SMALLER IS BETTER, RESNEPT: RESNET WITH PLAIN-CONVLUTION.) THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF THE CORRESPONDING SCORES.

Method	RNFL	GCIPL	INL	OPL	ONL	IS	OS-RPE	Fluid	Dice	IoU	HD↓	ED↓
ResNePt-UNet	0.8816 (0.0365)	0.9162 (0.0228)	0.7794 (0.0573)	0.7835 (0.0555)	0.9051 (0.0404)	0.9088 (0.0145)	0.8793 (0.0208)	0.5047 (0.3853)	0.8198 (0.1167)	0.7256 (0.1076)	28.3227 (50.6730)	27.0381 (102.7122)
ResNeCt-UNet	0.8770 (0.0400)	0.9076 (0.0251)	0.7625 (0.0630)	0.7869 (0.0496)	0.9044 (0.0365)	0.9104 (0.0184)	0.8863 (0.0218)	0.5596 (0.3792)	0.8243 (0.1143)	0.7291 (0.1086)	20.9569 (26.8471)	9.9297 (45.3758)
LHTran-UNet	0.8725 (0.0422)	0.8991 (0.0323)	0.7631 (0.0608)	0.7954 (0.0426)	0.8984 (0.0468)	0.9079 (0.0178)	0.8817 (0.0188)	0.5617 (0.3565)	0.8225 (0.1064)	0.7245 (0.1044)	17.7317 (26.3846)	7.8596 (39.9260)
TCCT	0.8612 (0.0582)	0.8935 (0.0405)	0.7747 (0.0624)	0.7909 (0.0465)	0.9131 (0.0307)	0.9073 (0.0134)	0.8865 (0.0205)	0.6129 (0.3617)	0.8300 (0.1079)	0.7335 (0.1044)	19.7154 (35.5530)	11.8575 (63.5493)
TCCT+L_{fpl}	0.8714 (0.0417)	0.9030 (0.0319)	0.7884 (0.0615)	0.7961 (0.0539)	0.9096 (0.0358)	0.9097 (0.0144)	0.8865 (0.0208)	0.6136 (0.3458)	0.8348 (0.1031)	0.7395 (0.1012)	22.0962 (34.1132)	12.0481 (63.4643)
TCCT+L_{brl}	0.8861 (0.0354)	0.9213 (0.0252)	0.7938 (0.0584)	0.7986 (0.0485)	0.9100 (0.0322)	0.9114 (0.0171)	0.8860 (0.0187)	0.5722 (0.3264)	0.8349 (0.0977)	0.7408 (0.0940)	21.7206 (42.9043)	13.1554 (80.7795)
TCCT-BP (Proposed)	0.8874 (0.0345)	0.9199 (0.0262)	0.7903 (0.0573)	0.7905 (0.0557)	0.9068 (0.0388)	0.9062 (0.0158)	0.8777 (0.0198)	0.6482 (0.2939)	0.8409 (0.0867)	0.7446 (0.0910)	13.7832 (24.7464)	7.2711 (40.0554)

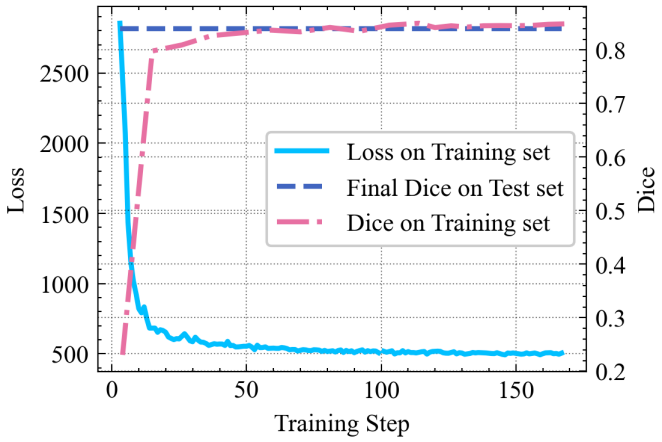

Fig. 4. Visualization of the fitting curves on Duke DME.

To ensure that there is no data overfitting in the proposed TCCT-BP, we draw the curves of loss and Dice during the training process and represent the final Dice of the test set with a dotted line in Fig. 4. We can observe that the loss of TCCT-BP converges rapidly, and there is almost no difference between the final Dice on the training and test sets. These results suggest that no overfitting phenomenon was present in our model.

C. Ablation study

To observe the performance gains caused by changes in network structure and different loss functions, we conducted ablation experiments, shown in Table II.

First, from the experimental results of ResNePt-UNet (i.e., ResNet-UNet without cross-convolution), Dice of Fluid increased from 0.5047 to 0.5596. This shows that cross-convolution can indeed increase CNN's ability of local perception, making it better optimize local details such as Fluid. In addition, the Dice scores of OPL, IS, OS-RPE were also improved. Then, we evaluated the performance of the model containing only a single backbone, i.e., ResNeCt (with cross-convolution) or LHTran (with ViT structure). As shown in Table II, the individual ResNeCt-UNet and LHTran-UNet both achieved Dice and IoU of around 0.82+ and 0.72+. Taking the layer of Fluid as example, ResNeCt-UNet with cross-convolution and LHTran-UNet with ViT structure significantly improve the Fluid segmentation accuracy compared with ResNePt-UNet with CNN. In addition, the best Dice of 0.6129 for Fluid is achieved by the dual backbone network TCCT, which combines LHTran and ResNeCt. This score is much higher than what ResNeCt-UNet and LHTran-UNet can achieve alone (i.e., 0.5596 and 0.5616, respectively). It's worth noting that cross-convolution and transformer algorithms share a common duty, i.e., to increase the receptive field size of a network. However, their structural differences introduce different complementary advantages. While cross-convolution enhances the local perception of a CNN, the transformer excels at capturing long-distance context. Therefore, our TCCT, the combination of ResNeCt and LHTran, can leverage the benefits of both strategies, leading to better segmentation performance by incorporating both local receptive fields and long-range dependencies. This is why the overall performance of TCCT is better than that of ResNeCt-UNet or LHTran-UNet alone.

Furthermore, both L_{fpl} and L_{brl} increased the overlap degree, but L_{fpl} hurt HD and ED to some extent. This is because the uncertainty of boundary labeling is relatively higher than that inside the layers. Combined with L_{fpl} and L_{brl}, both HD and ED errors were suppressed while the coincidence Dice and IoU were improved. Despite the success of L_{brl} in segmenting layers such as GCIPL and INL due to its inspiration from the narrow and long shape of the retina layers, L_{brl} was not suitable for Fluid segmentation. As a result, the Dice score for Fluid of TCCT+L_{brl} was 4% lower compared with that of TCCT+L_{fpl}. Combining L_{brl} and L_{fpl} can provide the advantages of both layer segmentation and Fluid complementarity.

We added our boundary regression loss L_{brl} to ReLayNet and MedT to test the gain for networks of different structures, as is shown in "Performance with L_{brl}" part of Table III. For ReLayNet, L_{brl} increased its scores of GCIPL, OPL, ONL, IS and Fluid layers, increased the average Dice from 0.8136 to 0.8205, and decreased ED from 10.1995 to 9.4551. For MedT, introducing of L_{brl} improved its Dice of ONL and Fluid, with the average Dice increased from 0.8153 to 0.8250, and ED decreased from 14.1964 to 10.5863. This shows that the proposed L_{brl} does bring clear positive effect on optimizing hard-to-divide layers such as OPL (OPL has a lower Dice score

TABLE III

COMPARISON WITH STATE-OF-THE-ART METHODS ON DUKE DME. RED IS THE BEST AMONG METHODS WITH DATA SPLIT OF [26]. ↓ MEANS SMALLER IS BETTER. / MEANS NOT AVAILABLE. THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF THE CORRESPONDING SCORES. ∗ ⋆ ○ MEAN THE P-VALUE OF THE PAIRED T TEST BETWEEN EACH METHOD WITH THE PROPOSED TCCT-BP (∗: P<0.001, ⋆: P<0.05, ○: P≥0.05). A▶B : THE RESULT OF B IS REPORTED IN A.

Method	RNFL	GCIPL	INL	OPL	ONL	IS	OS-RPE	Fluid	Dice	IoU	HD↓	ED↓
CNN Methods (training/test=8/2)												
[26]▶U-Net [13]	0.855	0.914	0.798	0.809	0.890	0.865	0.907	0.452	0.811	/	/	/
UNet++ [14]	0.8693 ∗ (0.0364)	0.9052 ⋆ (0.0327)	0.7769 ○ (0.0678)	0.7756 ⋆ (0.0504)	0.8993 ○ (0.0387)	0.9001 ○ (0.0173)	0.8802 ○ (0.0178)	0.5987 ○ (0.3346)	0.8257 ∗ (0.0995)	0.7261 ∗ (0.0993)	22.0520 ∗ (32.8499)	11.1109 ○ (53.7947)
U^2Net [15]	0.8785 ∗ (0.0420)	0.9133 ⋆ (0.0269)	0.7813 ○ (0.0618)	0.7824 ○ (0.0527)	0.9064 ○ (0.0347)	0.8980 ⋆ (0.0195)	0.8738 ○ (0.0277)	0.6280 ○ (0.3374)	0.8327 ⋆ (0.0999)	0.7353 ⋆ (0.0972)	16.5863 ○ (31.6342)	8.0184 ○ (40.1384)
nnU-Net [16]	0.8806 ⋆ (0.0398)	0.9153 ○ (0.0253)	0.7862 ○ (0.0518)	0.7776 ○ (0.0474)	0.8988 ⋆ (0.0403)	0.8952 ⋆ (0.0204)	0.8706 ○ (0.0287)	0.6449 ○ (0.2819)	0.8336 ⋆ (0.0819)	0.7330 ∗ (0.0870)	14.9271 ○ (26.6841)	7.2051 ○ (39.9842)
ViT Methods (training/test=8/2)												
TransUNet [2]	0.8712 ∗ (0.0380)	0.9009 ∗ (0.0352)	0.7652 ⋆ (0.0638)	0.7744 ⋆ (0.0485)	0.8899 ∗ (0.0447)	0.9039 ○ (0.0184)	0.8827 ○ (0.0222)	0.6111 ⋆ (0.2914)	0.8249 ∗ (0.0847)	0.7226 ∗ (0.0910)	31.0515 ∗ (16.3531)	6.7164 ○ (32.6641)
MedT [1]	0.8704 ∗ (0.0407)	0.9074 ⋆ (0.0276)	0.7804 ○ (0.0577)	0.7786 ○ (0.0541)	0.8942 ⋆ (0.0455)	0.9046 ○ (0.0184)	0.8781 ○ (0.0257)	0.5084 ∗ (0.3846)	0.8153 ∗ (0.1152)	0.7187 ∗ (0.1110)	23.0484 ⋆ (39.3111)	14.1964 ○ (63.0141)
MTU [17]	0.7835 ∗ (0.0474)	0.6427 ∗ (0.0469)	0.5741 ∗ (0.0740)	0.5745 ∗ (0.0894)	0.7386 ∗ (0.0596)	0.8977 ⋆ (0.0165)	0.8828 ○ (0.0245)	0.1713 ∗ (0.1597)	0.6582 ∗ (0.0423)	0.5293 ∗ (0.0234)	115.6923 ∗ (40.7532)	33.5375 ⋆ (80.8146)
C2FTrans [19]	0.6690 ∗ (0.0635)	0.6965 ∗ (0.0551)	0.4249 ∗ (0.1138)	0.4977 ∗ (0.1324)	0.8618 ∗ (0.0401)	0.8848 ∗ (0.0200)	0.8707 ⋆ (0.0187)	0.4683 ⋆ (0.4244)	0.6717 ∗ (0.1254)	0.5507 ∗ (0.1177)	31.4898 ∗ (45.1179)	24.5758 ⋆ (80.7598)
SwinUNet [18]	0.8633 ∗ (0.0391)	0.8959 ∗ (0.0380)	0.7630 ⋆ (0.0642)	0.7686 ⋆ (0.0541)	0.9037 ○ (0.0338)	0.9093 ○ (0.0188)	0.8851 ⋆ (0.0224)	0.5550 ⋆ (0.4029)	0.8180 ∗ (0.1213)	0.7214 ∗ (0.1132)	23.0128 ⋆ (37.5666)	13.3782 ○ (63.0993)
Specifically Designed Methods for Retinal Layer Segmentation (training/test=8/2)												
[26]▶GDP [20]	0.778	0.772	0.652	0.670	0.868	0.878	0.823	/	/	/	/	/
[26]▶LSE-GDP [21]	0.870	0.908	0.805	0.772	0.942	0.880	0.862	/	/	/	/	/
[26]▶BR-Net [5]	0.843	0.860	0.727	0.715	0.811	0.823	0.792	/	/	/	/	/
[26]▶BAU-Net [26]	0.873	0.937	0.818	0.826	0.906	0.894	0.908	0.527	0.836	/	/	/
[26]▶ReLayNet [12]	0.863	0.915	0.787	0.762	0.905	0.823	0.904	0.495	0.807	/	/	/
ReLayNet [12]	0.8721 ∗ (0.0383)	0.9058 ∗ (0.0337)	0.7773 ○ (0.0561)	0.7655 ∗ (0.0511)	0.8900 ∗ (0.0426)	0.9007 ⋆ (0.0151)	0.8774 ○ (0.0206)	0.5197 ○ (0.3076)	0.8136 ⋆ (0.0905)	0.7110 ∗ (0.0886)	35.2074 ∗ (24.5076)	10.1995 ○ (48.6084)
MGU [25]	0.8629 ∗ (0.0396)	0.8898 ∗ (0.0440)	0.7440 ∗ (0.0753)	0.7723 ⋆ (0.0566)	0.8991 ⋆ (0.0423)	0.9058 ○ (0.0208)	0.8830 ○ (0.0239)	0.6127 ○ (0.3340)	0.8212 ∗ (0.0975)	0.7197 ∗ (0.0961)	19.6101 ⋆ (26.6303)	6.6977 ○ (32.0956)
TCCT-BP (Ours)	0.8874 (0.0345)	0.9199 (0.0262)	0.7903 (0.0573)	0.7905 (0.0557)	0.9068 (0.0388)	0.9062 (0.0158)	0.8777 (0.0198)	0.6482 (0.2939)	0.8409 (0.0867)	0.7446 (0.0910)	13.7832 (24.7464)	7.2711 (40.0554)
Performance with L_{brl} (training/test=8/2)												
L_{brl}+ ReLayNet [12]	0.8696 (0.0386)	0.9079 (0.0318)	0.7751 (0.0548)	0.7661 (0.0538)	0.8931 (0.0411)	0.9044 (0.0182)	0.8760 (0.0212)	0.5715 (0.3180)	0.8205 (0.0937)	0.7190 (0.0938)	32.5845 (25.7649)	9.4551 (45.5641)
L_{brl}+ MedT [1]	0.8690 (0.0459)	0.8987 (0.0300)	0.7534 (0.0666)	0.7746 (0.0487)	0.8954 (0.0466)	0.9002 (0.0207)	0.8839 (0.0224)	0.6249 (0.3125)	0.8250 (0.0912)	0.7231 (0.0920)	20.2947 (31.8932)	10.5863 (54.0923)
With Unknown Pre-processing Methods (training/test=8/2)												
[28]▶ReLayNet [12]	0.90	0.94	0.87	0.84	0.93	0.92	0.90	0.77	0.88	/	/	/
[28]▶MAGNet [28]	0.92	0.96	0.92	0.90	0.94	0.95	0.88	0.88	0.9188	/	/	/

than other layers), increasing its Dice score and reducing its surface distance such as ED.

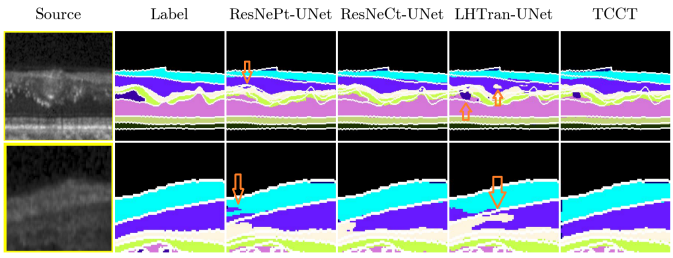

Fig. 5. Visual comparison of OCT layer segmentation with different network structures. The edges of OCT layers in ground truth are outlined with white lines. The predicted layers are shown in pseudo colors.

From Fig. 5, compared with the segmentation results of ResNePt-UNet, ResNeCt-UNet produces better results by improving CNN's ability of local detail processing via the extension of local receptive field using cross-convolution. In case where ResNePt-UNet with CNN is difficult to identify fluid accumulation, LHTran-UNet is capable of inferring the approximate location of the fluid by analyzing the shape of retinal layers due to its ability to capture long-distance dependencies (as indicated by the segmentation result of LHTran-Unet in the first row of Fig. 5). However, LHTran-UNet produces less refined segmentation results than ResNeCt-UNet with cross-convolution due to its slicing of the input. By combining LHTran and ResNeCt to incorporate both cross-convolution and long-distance dependence techniques, the proposed TCCT strikes a balance between fluid accumulation detection and segmentation continuity.

D. Comparison with state-of-the-arts

To comprehensively evaluate the performance of the proposed TCCT-BP, many excellent segmentation methods are included in our experiments, as shown in Tables III, IV, V, VI, and VIII. The Dice scores of each method are listed below each retinal layer name. The compared methods include transformer-based medical image segmentation methods: TransUNet [2], MedT [1], MTU [17] , C2FTrans [19], and SwinUNet [18], specifically designed OCT layer segmentation methods: GDP [20], LSE-GDP [21], AURA [22], BR-Net [5], SR-Net [23], BAU-Net [26], MGU [25] and ReLayNet [12], CNN-based classical medical image segmentation methods: U-Net [13], U^2Net [15], UNet++ [14], and nnU-Net [16]. Since the codes for these methods are available, to be fair, we trained and tested these models under the same data

TABLE IV

COMPARISON WITH STATE-OF-THE-ART METHODS ON DUKE DME. RED IS THE BEST AMONG METHODS WITH DATA SPLIT OF [12]. ↓ MEANS SMALLER IS BETTER. / MEANS NOT AVAILABLE. THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF THE CORRESPONDING SCORES. ∗ ⋆ ∘ MEAN THE P-VALUE OF THE PAIRED T TEST BETWEEN EACH METHOD WITH THE PROPOSED TCCT-BP (∗: P<0.001, ⋆: P<0.05, ∘: P≥0.05). A►B : THE RESULT OF B IS REPORTED IN A.

Method	RNFL	GCIPL	INL	OPL	ONL	IS	OS-RPE	Fluid	Dice	IoU	HD↓	ED↓
CNN Methods (training/test=5/5)												
U-Net [13]	0.8858 ∘ (0.0380)	0.9207 ⋆ (0.0251)	0.8202 ⋆ (0.0509)	0.8017 ∘ (0.0559)	0.9031 ∘ (0.0448)	0.8940 ∘ (0.0278)	0.8401 ∗ (0.0392)	0.5854 ∘ (0.3563)	0.8314 ∘ (0.1050)	0.7356 ∘ (0.0980)	23.7871 ⋆ (44.0779)	16.9543 ∘ (87.8906)
UNet++ [14]	0.8878 ∘ (0.0359)	0.9214 ∗ (0.0263)	0.8225 ∗ (0.0520)	0.8042 ∘ (0.0517)	0.9036 ∘ (0.0436)	0.8871 ∗ (0.0317)	0.8348 ∗ (0.0381)	0.6060 ∘ (0.3508)	0.8334 ⋆ (0.1032)	0.7375 ⋆ (0.1006)	24.5392 ∗ (34.4083)	11.8443 ∘ (71.0291)
U^2Net [15]	0.8420 ∗ (0.0431)	0.8966 ∗ (0.0535)	0.8118 ∘ (0.0601)	0.7767 ∗ (0.0655)	0.8972 ⋆ (0.0567)	0.8931 ⋆ (0.0262)	0.8245 ∗ (0.0411)	0.6532 ∘ (0.3301)	0.8244 ∗ (0.0935)	0.7212 ∗ (0.0918)	19.7132 ∘ (38.2958)	12.5917 ∘ (72.0004)
nnU-Net [16]	0.8840 ⋆ (0.0362)	0.9193 ⋆ (0.0262)	0.8241 ∗ (0.0523)	0.8030 ∘ (0.0517)	0.9040 ∘ (0.0431)	0.8889 ∗ (0.0296)	0.8287 ∗ (0.0430)	0.5949 ⋆ (0.3565)	0.8309 ⋆ (0.1049)	0.7346 ∗ (0.1017)	24.4695 ∗ (39.2338)	13.7359 ∘ (74.9315)
ReLayNet [12]	0.8711 ∗ (0.0349)	0.9012 ∗ (0.0378)	0.8004 ∘ (0.0600)	0.7945 ∗ (0.0528)	0.8997 ⋆ (0.0506)	0.8870 ∗ (0.0299)	0.8390 ∗ (0.0364)	0.6080 ∘ (0.3346)	0.8251 ⋆ (0.0969)	0.7236 ∗ (0.0907)	24.9968 ⋆ (39.1808)	15.1559 ∘ (87.5187)
MGU [25]	0.8758 ∗ (0.0349)	0.9096 ⋆ (0.0332)	0.8013 ∘ (0.0605)	0.8005 ⋆ (0.0512)	0.8962 ⋆ (0.0539)	0.8799 ∗ (0.0323)	0.8388 ∗ (0.0294)	0.4839 ⋆ (0.3613)	0.8108 ∗ (0.1061)	0.7104 ∗ (0.0903)	44.5988 ∗ (49.0095)	18.1335 ∘ (67.8550)
ViT Methods (training/test=5/5)												
TransUNet [2]	0.8452 ∗ (0.0538)	0.8960 ∗ (0.0453)	0.8000 ∘ (0.0656)	0.7993 ⋆ (0.0549)	0.8733 ∗ (0.0575)	0.8735 ∗ (0.0330)	0.8079 ∗ (0.0431)	0.4295 ∗ (0.3563)	0.7906 ∗ (0.1016)	0.6840 ∗ (0.0828)	49.9971 ∗ (50.2731)	31.7105 ⋆ (107.4435)
MedT [1]	0.7988 ∗ (0.0639)	0.8604 ∗ (0.0470)	0.7539 ∗ (0.0803)	0.7539 ∗ (0.0718)	0.8897 ∗ (0.0571)	0.8866 ∗ (0.0290)	0.8426 ∗ (0.0375)	0.5679 ⋆ (0.3803)	0.7942 ∗ (0.1087)	0.6849 ∗ (0.1031)	38.4872 ∗ (38.3768)	18.5072 ⋆ (68.5392)
MTU [17]	0.8429 ∗ (0.0548)	0.8660 ∗ (0.0557)	0.7654 ∗ (0.0755)	0.7798 ∗ (0.0617)	0.8974 ⋆ (0.0515)	0.8767 ∗ (0.0323)	0.7914 ∗ (0.0477)	0.6090 ∘ (0.3501)	0.8036 ∗ (0.0985)	0.6940 ∗ (0.0936)	23.0967 ⋆ (34.1420)	11.2183 ∘ (62.0868)
C2FTrans [19]	0.4312 ∗ (0.1002)	0.3397 ∗ (0.1211)	0.1974 ∗ (0.0415)	0.1316 ∗ (0.0281)	0.4349 ∗ (0.0719)	0.3748 ∗ (0.0481)	0.4901 ∗ (0.0549)	0.0012 ∗ (0.0040)	0.3001 ∗ (0.0356)	0.1888 ∗ (0.0277)	150.2761 ∗ (47.3721)	52.2721 ∗ (82.3519)
SwinUNet [18]	0.8348 ∗ (0.0433)	0.8727 ∗ (0.0561)	0.7649 ∗ (0.0706)	0.7555 ∗ (0.0691)	0.8904 ∗ (0.0500)	0.8891 ∗ (0.0246)	0.8375 ∗ (0.0366)	0.6131 ∘ (0.3677)	0.8072 ∗ (0.1060)	0.7002 ∗ (0.1010)	27.2509 ∗ (34.1092)	12.4964 ∘ (57.8469)
TCCT-BP (Ours)	0.8872 (0.0327)	0.9135 (0.0323)	0.8055 (0.0667)	0.8066 (0.0534)	0.9046 (0.0438)	0.8959 (0.0268)	0.8585 (0.0297)	0.6473 (0.3416)	0.8399 (0.1003)	0.7450 (0.0982)	17.3114 (36.4876)	10.8058 (65.9205)
With Unknown Pre-processing Methods (training/test=5/5)												
[12]►ReLayNet [12]	0.90	0.94	0.87	0.84	0.93	0.92	0.90	0.77	0.88	/	/	/

TABLE V

COMPARISON WITH STATE-OF-THE-ART METHODS ON HCMS. RED IS THE BEST AMONG METHODS WITH DATA SPLIT OF [26]. ↓ MEANS SMALLER IS BETTER. / MEANS NOT AVAILABLE. THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF THE CORRESPONDING SCORES. ∗ ⋆ ∘ MEAN THE P-VALUE OF THE PAIRED T TEST BETWEEN EACH METHOD WITH THE PROPOSED TCCT-BP (∗: P<0.001, ⋆: P<0.05, ∘: P≥0.05). A►B : THE RESULT OF B IS REPORTED IN A.

Method	RNFL	GCIPL	INL	OPL	ONL	IS	OS	RPE	Dice	IoU	HD↓	ED↓
CNN Methods (training/test=15/20)												
[26]►U-Net [13]	0.929	0.944	0.878	0.898	0.947	0.877	0.878	0.916	0.9084	/	/	/
UNet++ [14]	0.8752 ∗ (0.0647)	0.9343 ∗ (0.0350)	0.8745 ∗ (0.0359)	0.8967 ∗ (0.0272)	0.9441 ∗ (0.0182)	0.8666 ⋆ (0.0430)	0.8686 ∘ (0.0497)	0.9040 ∗ (0.0468)	0.8955 ∗ (0.0134)	0.8144 ∗ (0.0185)	40.0706 ∗ (10.5337)	3.3813 ∗ (2.0096)
U^2Net [15]	0.9099 ∗ (0.0570)	0.9335 ∗ (0.0380)	0.8726 ∗ (0.0353)	0.8952 ∗ (0.0300)	0.9426 ∗ (0.0235)	0.8685 ⋆ (0.0415)	0.8716 ∗ (0.0479)	0.8845 ∗ (0.0567)	0.8973 ∗ (0.0113)	0.8174 ∗ (0.0160)	24.9869 ∗ (8.8670)	2.4056 ∗ (2.5941)
nnU-Net [16]	0.9342 ∘ (0.0326)	0.9484 ∘ (0.0257)	0.8772 ∗ (0.0370)	0.8999 ∗ (0.0274)	0.9478 ∗ (0.0180)	0.8679 ∗ (0.0435)	0.8770 ∘ (0.0459)	0.9169 ∘ (0.0436)	0.9087 ∗ (0.0094)	0.8358 ∗ (0.0128)	3.3500 ∗ (0.9887)	0.7370 ∗ (0.0848)
ViT Methods (training/test=15/20)												
TransUNet [2]	0.9275 ∗ (0.0394)	0.9422 ∗ (0.0376)	0.8784 ∗ (0.0394)	0.9010 ⋆ (0.0290)	0.9471 ⋆ (0.0229)	0.8721 ∘ (0.0466)	0.8775 ∘ (0.0480)	0.9113 ⋆ (0.0454)	0.9071 ∗ (0.0082)	0.8334 ∗ (0.0114)	19.6482 ∗ (10.7259)	2.1290 ∗ (1.5375)
MedT [1]	0.8917 ∗ (0.0552)	0.9206 ∗ (0.0335)	0.8515 ∗ (0.0383)	0.8804 ∗ (0.0293)	0.9372 ∗ (0.0198)	0.8614 ∗ (0.0385)	0.8687 ∗ (0.0411)	0.9057 ∗ (0.0432)	0.8896 ∗ (0.0097)	0.8044 ∗ (0.0132)	11.8607 ∗ (7.4061)	1.1262 ∗ (0.6626)
MTU [17]	0.6883 ∗ (0.0884)	0.6094 ∗ (0.0902)	0.6990 ∗ (0.0710)	0.6341 ∗ (0.1116)	0.7710 ∗ (0.0614)	0.8219 ∗ (0.0861)	0.7894 ∗ (0.0870)	0.8525 ∗ (0.0669)	0.7332 ∗ (0.0150)	0.5919 ∗ (0.0119)	110.9489 ∗ (12.1654)	18.5432 ∗ (3.3954)
C2FTrans [19]	0.8001 ∗ (0.0860)	0.8167 ∗ (0.0873)	0.6968 ∗ (0.1128)	0.7544 ∗ (0.0857)	0.8962 ∗ (0.0386)	0.8466 ∗ (0.0481)	0.8624 ∗ (0.0400)	0.9096 ∗ (0.0417)	0.8228 ∗ (0.0268)	0.7098 ∗ (0.0232)	15.2249 ∗ (4.2540)	3.1850 ∗ (1.0026)
SwinUNet [18]	0.9149 ∗ (0.0506)	0.9308 ∗ (0.0454)	0.8661 ∗ (0.0488)	0.8935 ∗ (0.0327)	0.9459 ∗ (0.0211)	0.8692 ∗ (0.0459)	0.8775 ∗ (0.0540)	0.9146 ∗ (0.0456)	0.9016 ∗ (0.0101)	0.8246 ∗ (0.0138)	12.2618 ∗ (7.0315)	1.8824 ∗ (1.3639)
Specifically Designed Methods for Retinal Layer Segmentation (training/test=15/20)												
[26]►AURA [22]	0.938	0.945	0.874	0.899	0.947	0.874	0.875	0.916	0.9085	/	/	/
[26]►BR-Net [5]	0.939	0.950	0.879	0.903	0.948	0.868	0.873	0.913	0.9091	/	/	/
[26]►SR-Net [23]	0.937	0.951	0.883	0.902	0.950	0.872	0.872	0.907	0.9093	/	/	/
[26]►BAU-Net [26]	0.941	0.954	0.885	0.894	0.949	0.882	0.890	0.923	0.9148	/	/	/
[26]►ReLayNet [12]	0.925	0.943	0.875	0.897	0.946	0.869	0.876	0.915	0.9058	/	/	/
ReLayNet [12]	0.8218 ∗ (0.0771)	0.9330 ∗ (0.0336)	0.8650 ∗ (0.0373)	0.8857 ∗ (0.0288)	0.9428 ∗ (0.0201)	0.8703 ∘ (0.0380)	0.8694 ∗ (0.0447)	0.9035 ∗ (0.0429)	0.8865 ∗ (0.0157)	0.8004 ∗ (0.0189)	55.5590 ∗ (9.0343)	5.8856 ∗ (1.8274)
MGU [25]	0.8852 ∗ (0.0639)	0.9158 ∗ (0.0686)	0.8614 ∗ (0.0518)	0.8868 ∗ (0.0356)	0.9426 ∗ (0.0264)	0.8577 ∗ (0.0882)	0.8556 ∗ (0.0916)	0.9071 ∗ (0.0603)	0.8890 ∗ (0.0214)	0.8065 ∗ (0.0245)	25.0167 ∗ (9.5655)	2.4172 ∗ (1.6173)
TCCT-BP (Ours)	0.9369 (0.0296)	0.9509 (0.0238)	0.8836 (0.0336)	0.9047 (0.0286)	0.9500 (0.0190)	0.8728 (0.0419)	0.8779 (0.0464)	0.9172 (0.0425)	0.9117 (0.0091)	0.8409 (0.0126)	3.4357 (0.8766)	0.7112 (0.0650)
With Data Split of [28] (training/test=18/17)												
[28]►ReLayNet [12]	0.85	0.82	0.74	0.85	0.93	0.83	0.87	0.93	0.85	/	/	/
[28]►MAGNet [28]	0.89	0.95	0.87	0.90	0.94	0.92	0.89	0.93	0.9113	/	/	/
TCCT-BP (Ours)	0.9248 (0.0381)	0.9490 (0.0227)	0.8838 (0.0319)	0.8969 (0.0242)	0.9479 (0.0162)	0.8785 (0.0389)	0.8898 (0.0442)	0.9269 (0.0346)	0.9122 (0.0089)	0.8412 (0.0125)	4.8882 (3.0644)	0.7637 (0.1421)

TABLE VI

COMPARISON WITH STATE-OF-THE-ART METHODS ON HEG. RED IS THE BEST. ↓ MEANS SMALLER IS BETTER. THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF THE CORRESPONDING SCORES. $*$ $\star$ $\circ$ MEAN THE P-VALUE OF THE PAIRED T TEST BETWEEN EACH METHOD WITH THE PROPOSED TCCT-BP ($*$: P<0.001, $\star$: P<0.05, $\circ$: P$\geq$0.05).

Method	OS-RPE	IS	ONL	OPL	INL	GCIPL	RNFL	Dice	IoU	HD↓	ED↓
CNN Methods (training/test=5/5)											
UNet [13]	0.9275 $\star$ (0.0237)	0.9092 $\circ$ (0.0205)	0.9706 $*$ (0.0116)	0.9203 $*$ (0.0228)	0.9235 $*$ (0.0152)	0.9574 $*$ (0.0176)	0.9450 $*$ (0.0124)	0.9362 $*$ (0.0045)	0.8813 $*$ (0.0070)	5.1687 $\circ$ (2.9645)	0.6035 $*$ (0.0337)
UNet++ [14]	0.9330 $\circ$ (0.0183)	0.9114 $\circ$ (0.0218)	0.9739 $*$ (0.0092)	0.9287 $*$ (0.0186)	0.9276 $*$ (0.0168)	0.9604 $*$ (0.0190)	0.9459 $*$ (0.0104)	0.9401 $*$ (0.0043)	0.8882 $*$ (0.0071)	5.1916 $\circ$ (2.1920)	0.5639 $*$ (0.0287)
U^2Net [15]	0.9282 $\star$ (0.0247)	0.9069 $\circ$ (0.0308)	0.9685 $*$ (0.0103)	0.9177 $*$ (0.0169)	0.9278 $*$ (0.0138)	0.9604 $*$ (0.0161)	0.9420 $*$ (0.0160)	0.9359 $*$ (0.0065)	0.8809 $*$ (0.0101)	4.7729 $\circ$ (5.1504)	0.5976 $*$ (0.0405)
nnU-Net [16]	0.9398 $\circ$ (0.0272)	0.9188 $\circ$ (0.0275)	0.9757 $\circ$ (0.0090)	0.9356 $\star$ (0.0176)	0.9343 $\circ$ (0.0141)	0.9666 $\star$ (0.0139)	0.9568 $\star$ (0.0118)	0.9468 $\star$ (0.0068)	0.9001 $*$ (0.0110)	3.7538 $*$ (0.8659)	0.4916 $\star$ (0.0457)
ReLayNet [12]	0.8073 $*$ (0.0708)	0.8789 $*$ (0.0205)	0.9463 $*$ (0.0212)	0.8347 $*$ (0.0398)	0.8325 $*$ (0.0579)	0.9088 $*$ (0.0609)	0.7953 $*$ (0.0898)	0.8577 $*$ (0.0239)	0.7583 $*$ (0.0294)	48.2168 $*$ (6.2265)	4.3568 $*$ (2.6967)
MGU [25]	0.8695 $*$ (0.0415)	0.8612 $*$ (0.0492)	0.9115 $*$ (0.0183)	0.8645 $*$ (0.0364)	0.8527 $*$ (0.0390)	0.8913 $*$ (0.0612)	0.7801 $*$ (0.0573)	0.8615 $*$ (0.0133)	0.7613 $*$ (0.0174)	48.5040 $*$ (4.5802)	6.5856 $*$ (1.4235)
ViT Methods (training/test=5/5)											
TransUNet [2]	0.9327 $\circ$ (0.0220)	0.9139 $\circ$ (0.0199)	0.9737 $*$ (0.0101)	0.9309 $*$ (0.0189)	0.9313 $*$ (0.0150)	0.9628 $*$ (0.0159)	0.9517 $\star$ (0.0133)	0.9424 $*$ (0.0038)	0.8922 $*$ (0.0059)	9.2965 $*$ (6.4545)	0.5817 $*$ (0.0988)
MedT [1]	0.9302 $\star$ (0.0253)	0.9083 $\star$ (0.0282)	0.9723 $*$ (0.0102)	0.9245 $*$ (0.0177)	0.9276 $*$ (0.0169)	0.9604 $*$ (0.0179)	0.9463 $*$ (0.0148)	0.9385 $*$ (0.0057)	0.8855 $*$ (0.0091)	4.3169 $\circ$ (1.6757)	0.5744 $*$ (0.0595)
MTU [17]	0.8223 $*$ (0.0705)	0.8638 $*$ (0.0344)	0.8523 $*$ (0.0336)	0.7461 $*$ (0.0568)	0.8153 $*$ (0.0489)	0.8130 $*$ (0.0765)	0.8288 $*$ (0.0584)	0.8202 $*$ (0.0153)	0.7004 $*$ (0.0181)	80.3007 $*$ (5.2113)	15.1415 $*$ (1.3425)
C2FTrans [19]	0.7870 $*$ (0.0645)	0.7586 $*$ (0.0632)	0.7885 $*$ (0.0853)	0.6413 $*$ (0.0996)	0.6191 $*$ (0.1111)	0.7271 $*$ (0.1038)	0.7170 $*$ (0.1099)	0.7198 $*$ (0.0189)	0.5737 $*$ (0.0176)	69.4416 $*$ (5.4187)	8.1017 $*$ (1.3251)
SwinUNet [18]	0.9191 $*$ (0.0248)	0.8993 $*$ (0.0252)	0.9650 $*$ (0.0123)	0.9039 $*$ (0.0203)	0.9134 $*$ (0.0169)	0.9522 $*$ (0.0204)	0.9328 $*$ (0.0201)	0.9265 $*$ (0.0041)	0.8646 $*$ (0.0065)	7.7218 $*$ (5.0206)	0.7437 $*$ (0.3130)
TCCT+L_{brl}	0.9306 $\star$ (0.0241)	0.9102 $\circ$ (0.0315)	0.9752 $\star$ (0.0087)	0.9341 $\star$ (0.0164)	0.9299 $*$ (0.0153)	0.9622 $*$ (0.0145)	0.9480 $*$ (0.0121)	0.9415 $*$ (0.0072)	0.8907 $*$ (0.0113)	5.7415 $\star$ (5.6354)	0.5430 $*$ (0.0384)
TCCT-BP (Proposed)	0.9414 (0.0243)	0.9213 (0.0246)	0.9757 (0.0082)	0.9355 (0.0164)	0.9351 (0.0137)	0.9662 (0.0138)	0.9566 (0.0112)	0.9474 (0.0058)	0.9011 (0.0096)	3.8300 (1.3752)	0.4878 (0.0390)

TABLE VII

MICCAI 2022 FINAL OF TOP 15 ON GOALS (TASK1: OCT LAYER SEGMENTATION). RED IS THE BEST. VICBIC IS OUR TEAM.

Rank	Team	Score	RNFL-Dice	RNFL-ED↓	GCIPL-Dice	GCIPL-ED↓	Choroid-Dice	Choroid-ED↓
1	**ViCBiC (Ours)**	6.8826	0.9579	1.0528	0.9016	1.2243	0.9557	1.7327
2	Vision Wise	6.8809	0.9569	1.0833	0.8992	1.2566	0.9576	1.6295
3	segmentors	6.8790	0.9576	1.0827	0.8953	1.2322	0.9578	1.6468
4	toot	6.8727	0.9544	1.0689	0.8977	1.2486	0.9572	1.6845
5	SJMED	6.8721	0.9565	1.0899	0.8970	1.2801	0.9578	1.6456
6	WRMT	6.8668	0.9560	1.1020	0.8980	1.2719	0.9567	1.6769
7	AUTOMATE	6.8641	0.9561	1.1014	0.8966	1.2848	0.9569	1.6767
8	MedicalExploer	6.8550	0.9565	1.0888	0.8941	1.2972	0.9563	1.7452
9	OPTIMA-MUW	6.8508	0.9559	1.1084	0.8955	1.2838	0.9550	1.7599
10	CrashKing	6.8464	0.9563	1.0862	0.8943	1.2939	0.9540	1.8153
11	Miracle-boyi	6.8355	0.9562	1.1065	0.8904	1.3346	0.9542	1.7904
12	Latim	6.8338	0.9547	1.1554	0.8950	1.3804	0.9557	1.7120
13	SZUMed	6.8337	0.9555	1.1289	0.8921	1.3565	0.9553	1.7628
14	Parameter Tuner	6.7188	0.9533	1.8497	0.8897	1.3588	0.9539	1.7598
15	gbread	6.6129	0.9525	2.8120	0.8881	1.3845	0.9540	1.7818

enhancement conditions. The hyperparameter settings of MGU and ReLayNet refer to the requirements of the original works.

Note that the results reported in Table III on the Duke DME dataset is under 8:2 data split following BAU-Net [26]. For more comprehensive comparison, following ReLayNet [12], we also tested to divide the Duke DME dataset in 5:5 split, i.e., with the subjects $\#1-\#5$ in the training set and $\#6-\#10$ in the test set (55 B-scans in each set), and the performance comparison is reported in Table IV. Compared with Table III, the performances of various methods (including TCCT-BP, nnU-Net, and ReLayNet) in Table IV are reduced, due to the smaller training set and larger test set. The proposed TCCT-BP still works well, performing best in Table IV in terms of Dice, IoU, HD and ED. Note that the advantage of TCCT-BP in ED metric is not statistically significant compared with some methods like SwinUNet,UNet, and nnU-Net.

From Tables III, IV, V, VI, and VIII, we also noticed that the proposed TCCT-BP achieves promising ED and HD scores, most of which rank best or second best among the methods compared in these tables. Considering ED and HD sensitively measure the average and maximum distance errors of layer boundaries, respectively, the relatively lower ED and HD indicate that our TCCT-BP model performs better than most existing state-of-the-art methods in optimizing average boundary distance and reducing the dispersion degree of boundary distance.

From the perspective of model structure, the compared CNN-based methods are all based on UNet structure. The significant difference among them is that ReLayNet has only 0.793M parameters, which may result that its feature representation ability is not as good as UNet, UNet++ and nnU-Net containing more parameters, as consistently indicated across

TABLE VIII

COMPARISON WITH STATE-OF-THE-ART METHODS ON GOALS. RED IS THE BEST. ↓ MEANS SMALLER IS BETTER. THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF THE CORRESPONDING SCORES. ∗ ⋆ ○ MEAN THE P-VALUE OF THE PAIRED T TEST BETWEEN EACH METHOD WITH THE PROPOSED TCCT-BP (∗: P<0.001, ⋆: P<0.05, ○: P≥0.05).

Method	Choroid	INL-RPE	GCIPL	RNFL	Dice	IoU	HD↓	ED↓
	CNN Methods (training/test=5/5)							
UNet [13]	0.6080 ∗ (0.1115)	0.8376 ∗ (0.0395)	0.7116 ∗ (0.0898)	0.8691 ∗ (0.0778)	0.7566 ∗ (0.0262)	0.6259 ∗ (0.0219)	72.3899 ∗ (17.2543)	13.9580 ∗ (2.7698)
UNet++ [14]	0.9455 ○ (0.0214)	0.9646 ∗ (0.0114)	0.8941 ○ (0.0355)	0.9517 ○ (0.0207)	0.9390 ∗ (0.0086)	0.8870 ⋆ (0.0122)	20.6983 ⋆ (19.6117)	2.1174 ⋆ (2.4842)
U^2Net [15]	0.9495 ○ (0.0210)	0.9639 ∗ (0.0113)	0.8979 ○ (0.0284)	0.9493 ⋆ (0.0192)	0.9401 ○ (0.0061)	0.8888 ○ (0.0089)	6.8318 ○ (0.3975)	1.3860 ○ (0.0787)
nnU-Net [16]	0.9499 ○ (0.0248)	0.9669 ⋆ (0.0126)	0.8966 ○ (0.0307)	0.9515 ○ (0.0160)	0.9412 ○ (0.0072)	0.8909 ○ (0.0105)	6.8893 ○ (0.6726)	1.3505 ○ (0.1501)
ReLayNet [12]	0.7969 ∗ (0.0665)	0.8720 ∗ (0.0352)	0.6852 ∗ (0.0575)	0.8913 ∗ (0.0394)	0.8114 ∗ (0.0129)	0.6930 ∗ (0.0122)	77.1581 ∗ (10.7657)	9.0463 ∗ (1.5436)
MGU [25]	0.9462 ○ (0.0331)	0.9662 ⋆ (0.0130)	0.8946 ⋆ (0.0335)	0.9517 ○ (0.0188)	0.9397 ⋆ (0.0089)	0.8885 ⋆ (0.0130)	10.8991 ⋆ (6.5951)	1.4981 ⋆ (0.3449)
	ViT Methods (training/test=5/5)							
TransUNet [2]	0.9436 ○ (0.0283)	0.9667 ⋆ (0.0155)	0.8923 ⋆ (0.0371)	0.9492 ⋆ (0.0200)	0.9380 ∗ (0.0083)	0.8855 ∗ (0.0115)	19.8753 ∗ (4.1347)	1.7864 ∗ (0.4401)
MedT [1]	0.9332 ⋆ (0.0365)	0.9625 ∗ (0.0111)	0.0000 ∗ (0.0000)	0.9453 ∗ (0.0240)	0.7103 ∗ (0.0137)	0.6755 ∗ (0.0226)	296.6514 ∗ (12.3019)	330.5127 ∗ (56.9299)
MTU [17]	0.7162 ∗ (0.0895)	0.8261 ∗ (0.0613)	0.5245 ∗ (0.1127)	0.7834 ∗ (0.0755)	0.7126 ∗ (0.0190)	0.5715 ∗ (0.0090)	113.4990 ∗ (22.2092)	24.2130 ∗ (2.9917)
C2FTrans [19]	0.5671 ∗ (0.1209)	0.6614 ∗ (0.0608)	0.4514 ∗ (0.0957)	0.6422 ∗ (0.1281)	0.5805 ∗ (0.0263)	0.4208 ∗ (0.0224)	166.1301 ∗ (16.1207)	23.7859 ∗ (2.9324)
SwinUNet [18]	0.9237 ∗ (0.0342)	0.9539 ∗ (0.0127)	0.8596 ∗ (0.0344)	0.9390 ∗ (0.0236)	0.9190 ∗ (0.0089)	0.8533 ∗ (0.0129)	18.5426 ∗ (13.9553)	2.1617 ∗ (0.9589)
TCCT+L_{brl}	0.9444 ○ (0.0287)	0.9654 ∗ (0.0132)	0.8926 ○ (0.0362)	0.9499 ⋆ (0.0178)	0.9381 ∗ (0.0090)	0.8856 ∗ (0.0130)	19.3136 ∗ (14.4575)	1.6913 ○ (2.0701)
TCCT-BP (Proposed)	0.9459 (0.0274)	0.9687 (0.0137)	0.8989 (0.0360)	0.9529 (0.0187)	0.9416 (0.0085)	0.8918 (0.0121)	8.5967 (12.1856)	1.3914 (0.2260)

Tables III, IV, V, VI, and VIII. In addition, each dataset has its own difficulties, which leads to fluctuations in the performance of different algorithms. There is no serious deformation in images of HCMS and HEG, so the CNN-based methods work well on the whole. However, the narrow GCIPL and Choroid layers in GOALS are difficult for the models of UNet and ReLayNet. The performance difference among the compared methods on Duke DME is mainly resulted from the ability of segmenting Fluid. In Tables III and IV, nnU-Net and U2Net obtain higher Dice values, while UNet behaves poorly on Fluid. In short, the above reasons lead to the differences in the performance of the CNN-based methods on different datasets.

It is easy to see that the proposed method TCCT-BP ranks first on the Dice scores of multiple OCT layers. Further, TCCT-BP achieves the highest performance on two overlapping metrics (i.e., Dice, IoU) on Duke DME, HEG and GOALS (as shown respectively in Tables III, VI, and VIII), and obtains the second highest Dice on the HCMS dataset, next only to BAU-Net, a method that employs specific data preprocessing. Note that since the images in the datasets of HCMS, HEG and GOALS do not contain Fluid, many methods perform exceptionally well and show minimal variation in their scores. However, on the Duke DME dataset, our TCCT-BP achieves the Dice of 0.6482 in segmentation of Fluid, which surpasses the performance of BAU-Net, ReLayNet, and MGU, as shown in Table III. The advantage of our TCCT-BP exhibited on the four datasets can be attributed to its tight integration of the CNN and ViT backbones, as well as the loss of L_{brl} to specifically address the issue of boundary distortion, and the loss of L_{fpl} to deal with the false positive problem, particularly in Fluid segmentation.

The dual backbone structure provides our TCCT-BP with the ability to take into account both of the large local receptive fields and long-distance dependencies. This allows TCCT-BP to handle retinal layer boundary information more effectively with quite small number of parameters required. In situations where Fluid is challenging to detect due to blurring, TCCT-BP can use retinal topology to approximate the locations of Fluid and combine them with CNN for more precise segmentation. As a result, TCCT-BP outperforms the models such as TransUNet and MGU for the segmentation of Fluid, as demonstrated in Table III.

As for the MICCAI 2022 GOALS Challenge, we list the final ranking of the OCT layer segmentation task, as shown in Table VII (with 100 images with annotations for training and 100 images without annotations for test, which are publicly available). Our team (ViCBiC) took the lead in both the RNFL and GCIPL layers and achieved the first segmentation overall ranking in the final. This also proves the superiority of the proposed method.

E. Cross dataset evaluation

In order to assess the generalizability of the proposed TCCT-BP to unseen OCT images, we conducted new cross evaluation experiments to compare TCCT-BP with other models designed for retinal layer segmentation, namely ReLayNet [12] and MGU [25], as well as two Transformers, TransUNet [2] and SwinUNet [18]. For these cross evaluation experiments, we utilized the Duke DME and HEG datasets, both of which contain seven retinal layers. However, Duke DME additionally includes DME while HEG does not. We first trained the models on Duke DME and tested on HEG. Table IX indicates that the proposed TCCT-BP achieves the highest Dice scores

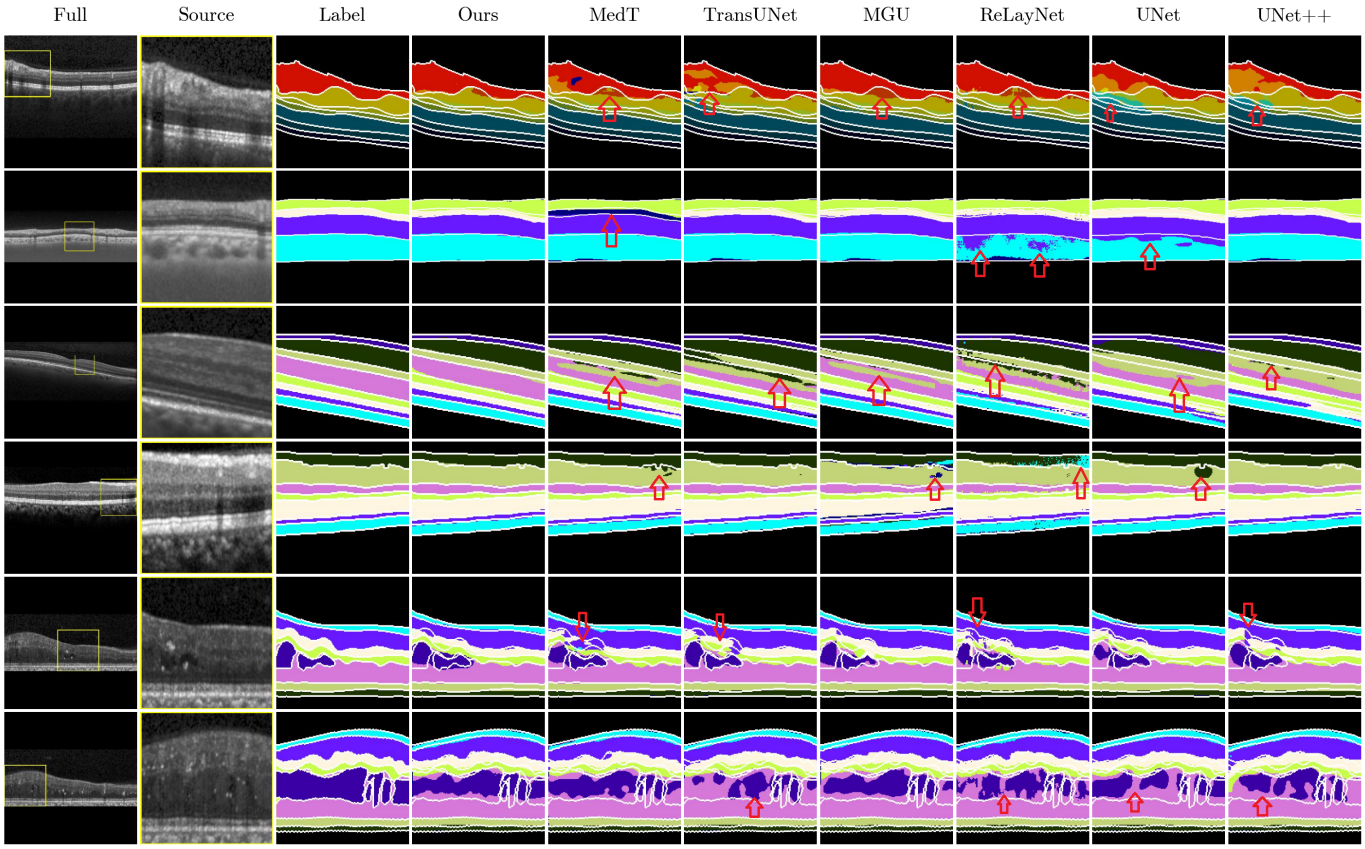

Fig. 6. Visual comparison of OCT layer segmentation by different models. The images are from Duke DME, GOALS, HCMS, and HEG respectively. The edges of OCT layers in ground truth are outlined with white lines. The predicted layers are shown in pseudo colors. The main differences between the results of different methods are marked with arrows.

TABLE IX

CROSS EVALUATION ON DUKE DME AND HEG. RED IS THE BEST. ↓ MEANS SMALLER IS BETTER. THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF THE CORRESPONDING SCORES. $*$ $\star$ $\circ$ MEAN THE P-VALUE OF THE PAIRED T TEST BETWEEN EACH METHOD WITH THE PROPOSED TCCT-BP ($*$: P<0.001, $\star$: P<0.05, $\circ$: P$\geq$0.05).

Method	RNFL	GCIPL	INL	OPL	ONL	IS	OS-RPE	Dice	IoU	HD	ED
Training on Duke DME and test on HEG											
ReLayNet [12]	0.5942 $\star$ (0.0832)	0.6648 $*$ (0.0887)	0.6368 $*$ (0.0819)	0.6219 $*$ (0.1121)	0.6332 $*$ (0.1265)	0.4479 $*$ (0.0643)	0.2750 $*$ (0.1050)	0.5534 $*$ (0.0196)	0.3991 $*$ (0.0213)	101.6266 $*$ (12.0047)	28.2069 $*$ (1.6709)
MGU [25]	0.6269 $\circ$ (0.1031)	0.7444 $*$ (0.0686)	0.5630 $*$ (0.1242)	0.6030 $*$ (0.1148)	0.7505 $*$ (0.1242)	0.3870 $*$ (0.1549)	0.0220 $*$ (0.0245)	0.5281 $*$ (0.0398)	0.3977 $*$ (0.0404)	114.8445 $*$ (50.3789)	42.6227 $*$ (61.3905)
TransUNet [2]	0.6187 $\circ$ (0.0790)	0.6728 $*$ (0.0881)	0.5739 $*$ (0.0805)	0.5658 $*$ (0.0974)	0.5103 $*$ (0.0555)	0.4655 $*$ (0.1467)	0.1001 $*$ (0.0734)	0.5010 $*$ (0.0266)	0.3554 $*$ (0.0259)	96.5398 $*$ (21.8462)	28.8973 $*$ (6.6562)
SwinUNet [18]	0.6399 $\circ$ (0.0565)	0.5234 $*$ (0.0886)	0.3313 $*$ (0.1021)	0.4237 $*$ (0.1002)	0.5742 $*$ (0.0615)	0.1217 $*$ (0.0895)	0.6142 $*$ (0.1067)	0.4612 $*$ (0.0184)	0.3190 $*$ (0.0153)	104.9840 $*$ (9.0398)	36.7807 $*$ (1.5930)
TCCT-BP (Proposed)	0.6295 (0.1354)	0.8459 (0.0719)	0.6740 (0.0821)	0.7134 (0.0793)	0.8482 (0.0755)	0.7655 (0.1008)	0.4241 (0.1829)	0.7001 (0.0381)	0.5644 (0.0269)	70.6721 (15.6358)	12.7427 (5.9994)
Training on HEG and test on Duke DME											
ReLayNet [12]	0.5261 $*$ (0.0641)	0.7122 $*$ (0.0706)	0.5590 $*$ (0.1061)	0.5424 $*$ (0.0939)	0.8019 $*$ (0.0440)	0.5489 $*$ (0.0624)	0.5337 $\star$ (0.1144)	0.6035 $*$ (0.0239)	0.4450 $*$ (0.0194)	63.3062 $*$ (6.3110)	17.4523 $*$ (2.1845)
MGU [25]	0.4662 $*$ (0.0539)	0.6398 $*$ (0.0817)	0.5911 $*$ (0.1431)	0.5552 $*$ (0.1248)	0.8166 $*$ (0.0468)	0.4747 $*$ (0.1172)	0.3659 $*$ (0.2028)	0.5585 $*$ (0.0506)	0.4093 $*$ (0.0375)	56.7700 $*$ (3.9459)	16.6184 $*$ (2.5006)
TransUNet [2]	0.7612 $*$ (0.0700)	0.7503 $*$ (0.0570)	0.6631 $*$ (0.1072)	0.6090 $\star$ (0.1426)	0.3983 $*$ (0.2307)	0.1791 $*$ (0.1701)	0.5617 $*$ (0.1591)	0.5604 $*$ (0.0562)	0.4249 $*$ (0.0400)	72.0907 $*$ (7.3255)	24.5309 $*$ (3.8637)
SwinUNet [18]	0.7744 $*$ (0.0559)	0.7256 $*$ (0.0915)	0.5006 $*$ (0.1944)	0.5593 $\star$ (0.1091)	0.7732 $*$ (0.0709)	0.5362 $*$ (0.1501)	0.4947 $*$ (0.1465)	0.6234 $*$ (0.0456)	0.4744 $*$ (0.0269)	53.4894 $*$ (5.9439)	9.6109 $*$ (3.1674)
TCCT-BP (Proposed)	0.8626 (0.0374)	0.8816 (0.0624)	0.7339 (0.1329)	0.6944 (0.1378)	0.8615 (0.0394)	0.7311 (0.0745)	0.4767 (0.1263)	0.7488 (0.0409)	0.6229 (0.0366)	34.2115 (12.6523)	5.4137 (2.1447)

from GCIPL to IS layers. We then trained the models on HEG and tested on Duke DME. In this scenario, the proposed TCCT-BP achieves the best scores on all the layers except OS-RPE, as shown in Table IX. As for the overall performance, TCCT-BP achieves the clearly best scores in both test scenarios on average Dice, IoU, HD and ED, demonstrating its higher generalization ability compared with ReLayNet, MGU, TransUNet, and SwinUNet.

F. Intra-layer false positives

Under the influence of speckle noise and tissue artifacts, the intra-layer segmentation results of various networks often

TABLE X

RESULTS OF NORMAL AND PATHOLOGICAL IMAGES ON DUKE DME AND HCMS. THE NUMBERS IN PARENTHESES ARE THE VARIANCES OF ALL IMAGES. THE NUMBERS IN BRACKETS ARE THE VARIANCES OF ALL VOLUMES (BY AVERAGING SCORES OF IMAGES FROM PER 3D SCAN).

Results of normal and pathological images on Duke DME												
Dataset	RNFL	GCIPL	INL	OPL	ONL	IS	OS-RPE	Fluid	Dice	IoU	HD	ED
Normal	0.9132 (0.0084)	0.9060 (0.0359)	0.7835 (0.0682)	0.8238 (0.0539)	0.9412 (0.0104)	0.8988 (0.0174)	0.8731 (0.0241)	1.0000 (0.0000)	0.8924 (0.0222)	0.8130 (0.0286)	4.1440 (1.3891)	0.9220 (0.1801)
Pathological	0.8770 (0.0354)	0.9250 (0.0191)	0.7928 (0.0524)	0.7779 (0.0511)	0.8937 (0.0376)	0.9089 (0.0143)	0.8792 (0.0177)	0.5163 (0.2344)	0.8213 (0.0682)	0.7186 (0.0514)	17.9356 (28.5939)	10.7096 (52.8465)
Whole	0.8874 (0.0345)	0.9199 (0.0262)	0.7903 (0.0573)	0.7905 (0.0557)	0.9068 (0.0388)	0.9062 (0.0158)	0.8777 (0.0198)	0.6482 (0.2939)	0.8409 (0.0867)	0.7446 (0.0910)	13.7832 (24.7464)	7.2711 (40.0554)
Results of normal and pathological images on HCMS												
Dataset	RNFL	GCIPL	INL	OPL	ONL	IS	OS	RPE	Dice	IoU	HD	ED
Normal	0.9393 (0.0298) [0.0001]	0.9579 (0.0165) [0.0000]	0.8901 (0.0292) [0.0001]	0.9104 (0.0232) [0.0000]	0.9515 (0.0162) [0.0001]	0.8914 (0.0341) [0.0004]	0.8856 (0.0517) [0.0012]	0.9241 (0.0385) [0.0010]	0.9188 (0.0111) [0.0001]	0.8524 (0.0157) [0.0003]	7.8016 (6.3555) [2.2060]	1.3791 (0.9454) [0.0250]
Pathological	0.9277 (0.0380) [0.0014]	0.9432 (0.0292) [0.0003]	0.8756 (0.0392) [0.0003]	0.9003 (0.0308) [0.0002]	0.9479 (0.0201) [0.0002]	0.8596 (0.0509) [0.0008]	0.8667 (0.0506) [0.0008]	0.9052 (0.0442) [0.0013]	0.9033 (0.0101) [0.0003]	0.8273 (0.0134) [0.0006]	9.9918 (8.0200) [78.7082]	1.5922 (0.5376) [0.2883]
Whole	0.9324 (0.0354) [0.0009]	0.9491 (0.0259) [0.0002]	0.8814 (0.0362) [0.0003]	0.9043 (0.0284) [0.0002]	0.9494 (0.0187) [0.0001]	0.8723 (0.0476) [0.0009]	0.8742 (0.0519) [0.0010]	0.9128 (0.0430) [0.0012]	0.9095 (0.0106) [0.0003]	0.8374 (0.0145) [0.0006]	9.1157 (7.3335) [50.8167]	1.5070 (0.7013) [0.2086]

show high false positives, as illustrated in the upper two rows of Fig. 6. On the one hand, effusion in the second row caused significant damage to the intra-layer consistency of ReLayNet and UNet, while the white spots in the first row rendered the segmentation results of various ViT models and CNN methods unacceptable. However, the proposed method is almost unaffected by effusion and blotches. On the other hand, vascular artifacts in the first row also caused large areas of confusion in the segmentation results of MedT, TransUNet, and UNet models, but had no impact on the proposed method. This could be attributed to the feature polarization loss which maximizes the distance between the features of different layers. The feature polarization loss constrains the three-scale features of the encoder and decoder of TCCT, which enables the model to enhance the discrimination of different retinal layers in the two processes of feature extraction and reconstruction of segmentation results.

G. Inter-layer boundary distortion and topology

The problem of false positives is accompanied by boundary distortion, which is more prevalent, as evidenced by Fig. 6. This is due to the abundance of artifacts and blurring in OCT images, manifesting as boundary collapse and uneven bulge. For instance, the artifact in the second row caused damage to retinal layers of MedT, ReLayNet, and UNet, resulting in an absence of a smooth boundary. The high blurring and very low contrast of the third and fourth rows of images caused a wide range of layer boundary offset by the networks of various ViT and CNN structures. In contrast, the proposed method is less affected by these issues due to the introduced boundary regression loss, which encourages the layer boundary to be smoother. Furthermore, the boundary regression loss term of coordinate regression layer by layer in L_{brl} can make the output of our model maintain a certain topological order to some extent, in addition to the gradient edge regression.

For the methods to be translated to clinical practice the output needs to be anatomically correct. In fact, several studies have attempted to enhance the algorithm's topological characteristics. Wang *et al.* [26] introduced a topology guarantee loss to realize better boundary detection. He *et al.* [5] designed a topology guaranteeing module aiming to obtain correct topology. In contrast, ReLayNet [12] and the proposed TCCT-BP adopted boundary regression strategies to guarantee the topology of retinal layers as much as possible. Quantitatively, a segmentation model producing a higher Dice score is indicated to have the better ability of guaranteeing the topology of the retinal layers as well as the reliability of clinical studies (e.g., GCIPL thickness measurement and visual improvement evaluation before and after epiretinal membrane surgery [40]. From the experimental results on the four widely used datasets (Tables III, IV, V, VI, and VIII), our TCCT-BP can robustly produce quite clinically acceptable Dice scores (0.8409, 0.9122, 0.9474, and 0.9416, ranking first or second among various state-of-the-arts).

H. Segmentation with Fluid

It should be noted that severe pathology, such as Fluid in the Duke DME dataset, can result in profound deformation of the retinal layers, and subsequently, impact the segmentation of retinal layers. That is why some image segmentation methods such as MTU [17] and MedT [1] perform well on normal OCT images, but exhibit a significant degradation of performance when applied to images containing effusion.

We segregated the test set into normal and pathological images (i.e., OCT images with Fluid or MS as pathological images and the rest as normal images) and evaluated the performance of TCCT-BP on the both subsets. As for the Duke DME test set including 22 images, its normal and pathological subsets contain 6 and 16 images, respectively, and as for the HCMS test set containing 980 images, its normal and pathological subsets comprise respectively 392 and 588 images. As shown in Table X, the presence of fluid accumulation significantly affected the segmentation accuracy of the RNFL and OPL of Duke DME, resulting in an average drop of approximately 4% to 5% of RNFL and OPL. However, the impact on other layers was relatively less pronounced. In contrast, multiple sclerosis seemed to have a negative impact on the segmentation of each retinal layer (from RNFL to

RPE), for example, the Dice of IS decreased from 0.8914 to 0.8596. The HCMS dataset consists of 3D OCT-scans only containing normal or pathological images (i.e., with Multiple Sclerosis). Therefore, we computed the mean/variance over all the images and over all volumes (by averaging the results on images per OCT scan), as is shown in Table X. While a 3D OCT-scan from Duke DME may contain both normal and pathological (i.e., with Fluid) images, therefore we only computed the mean/variance over all the images.

Qualitatively, the final two rows of Fig. 6 display the images featuring Fluid and their segmentation results of multiple models. The results indicate that TCCT-BP and MGU exhibit superior consistency in effusion segmentation compared with other models including TransUNet, ReLayNet, UNet, and UNet++. This contrast is particularly evident in the last row of Fig. 6. Furthermore, the second last row of the diagram illustrates that MedT, TransUNet, ReLayNet, UNet, and UNet++ produce segmentation fractures and distortions in the retinal layers due to Fluid, whereas these deformations due to Fluid have minimal impact on TCCT-BP and MGU.

I. Analysis of performance

We added the comparisons of inference time, FLOPs and parameter amount of our model and other deep learning models in Table XI. In terms of parameter amount, the proposed ResNeCt and LHTran are lightweight with only 0.355M and 0.709M learnable parameters, respectively, and the proposed TCCT-BP has only 0.988M learnable parameters, ranking second in comparison to all the other models, slightly heavier than ReLayNet having 0.793M parameters. TCCT-BP achieves competitive performance with less than 1/30 of the parameters of nnU-Net. Concerning computational complexity, our TCCT-BP is the fourth simplest one in terms of FLOPs with a value of 7.332G. With relatively less parameter amount and complexity, our TCCT-BP achieves acceptable inference time, i.e., 8.198 seconds and 1.406 seconds, respectively, on CPU and GPU for 32 images with a size of 256x256 pixels, surpassing the medical Transformers like TransUNet and MedT.

To summarize, the superior segmentation performance of the proposed TCCT-BP is mainly gained by the new network structure and the loss functions, which also bring the network lightness and computation efficiency.

J. Failure cases

As demonstrated above, the proposed method makes progress in both anti-false positive and boundary optimization. However, for some special cases, the segmentation results are not satisfactory enough, as shown in Fig. 7. On the one hand, in the first row of Fig. 7, the retinal layers are seriously distorted by blood vessels, and the segmentation results of all the listed models are difficult to fit the sharp bulges of the deformation. The proposed model and boundary regression loss are suitable for smoothing layer boundaries but fail for such case. On the other hand, a thick blood vessel artifact in the second row of Fig. 7 makes the fuzzy boundaries more difficult to identify. Although the proposed method has better inference results for this layer than other models, a small number of false positives still appear in the thinner layers. Both of these issues need better solutions.

TABLE XI
ILLUSTRATION OF THE COMPUTATIONAL COMPLEXITY AND PARAMETERS, AS WELL AS THE INFERENCE TIME FOR 32 256×256 IMAGES ON CPU/GPU. (CPU: INTEL(R) XEON(R) CPU E5-2620 V4 @ 2.10GHZ. GPU: 24 GB TITAN RTX. ⓜ: INPUT SIZE IS MULTI-SCALE. ⓕ: INPUT SIZE IS FIXED SCALE. /: NOT AVAILABLE.)

Method	FLOPs(G)	Param(M)	Inference Time on CPU (s)	Inference Time on GPU (s)
CNN Methods				
UNet [13] ⓜ	16.431	8.638	5.114	0.418
UNet++ [14] ⓜ	34.673	9.164	11.561	1.066
U^2Net [15] ⓜ	13.183	1.159	5.796	2.518
nnU-Net [16] ⓜ	15.745	32.959	4.193	1.983
MGU [25] ⓜ	3.712	2.094	2.059	0.890
ReLayNet [12] ⓜ	16.971	0.793	4.451	0.481
BAU-Net [26] ⓜ	/	1.27	/	/
MAGNet [28] ⓜ	/	1.45	/	/
ViT Methods				
TransUNet [2] ⓕ	47.417	125.903	11.997	1.986
MedT [1] ⓕ	2.904	1.568	47.773	11.305
C2FTrans [19] ⓜ	8.635	1.178	7.226	7.617
MTU [17] ⓕ	41.908	79.075	20.088	45.386
SwinUNet [18] ⓕ	6.164	27.169	6.385	0.696
Our Methods				
ResNeCt-UNet ⓜ	5.839	0.355	7.792	0.483
LHTran-UNet ⓜ	6.040	0.709	7.327	0.518
TCCT-BP ⓜ	7.332	0.988	8.198	1.406

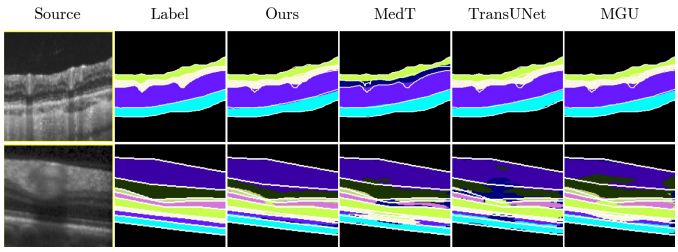

Fig. 7. Visual comparison of OCT layer segmentation under condition of vessel artifacts. The edges of OCT layers in ground truth are outlined with white lines. The predicted layers are shown in pseudo colors.

Although TCCT-BP achieved high performance on the datasets of Duke DME, HCMS, HEG and GOALS, even surpassing many algorithms designed specifically for OCT segmentation in Dice scores, there are some challenges for TCCT-BP in optimizing boundary distance, as evidenced by the less than ideal HD and the high variance in boundary distance in the segmentation results. Table X and the last two rows of Fig. 6 demonstrate that there is considerable room for improvement in effusion segmentation. Additionally, TCCT-BP struggles to achieve better segmentation in case of severe retinal layer deformation caused by thick blood vessels, as shown in Fig. 7. Finally, in the cross evaluation experiments detailed in Table IX, TCCT-BP exhibits clear decline in performance like other methods, which warrants further investigation.

V. CONCLUSION

In this work, we proposed a CNN and ViT hybrid network as well as the feature polarization loss and the boundary

regression loss for OCT layer segmentation. This study has identified the benefits of the complementary receptive fields of the ViT and CNN dual backbone networks for OCT layer segmentation. In addition, the loss of feature polarization by maximizing the distance between the hidden layers of different retinal layers also proved to be effective in suppressing false positives. Finally, the boundary regression loss used to constrain the retinal layer boundaries also successfully maintained small surface distances in the segmentation results. The experiments confirmed that the proposed TCCT-BP has achieved state-of-the-art performance. Future research will focus on designing simpler and more robust OCT layer segmentation networks to adapt various types of retinal layer deformation and applying the segmentation results to the analysis and diagnosis of ophthalmic diseases, for example, glaucoma.

REFERENCES

[1] J. M. J. Valanarasu, P. Oza, I. Hacihaliloglu, and V. M. Patel, "Medical transformer: Gated axial-attention for medical image segmentation," in *Proc. MICCAI*. Springer, 2021, pp. 36–46.

[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] C. K. S. Leung, A. K. N. Lam, R. N. Weinreb, D. F. Garway-Heath, M. Yu, P. Y. Guo, V. S. M. Chiu, K. H. N. Wan, M. Wong, K. Z. Wu *et al.*, "Diagnostic assessment of glaucoma and non-glaucomatous optic neuropathies via optical texture analysis of the retinal nerve fibre layer," *Nature Biomedical Engineering*, vol. 6, no. 5, pp. 593–604, 2022.

[4] S. C. Lin, K. Singh, H. D. Jampel, E. A. Hodapp, S. D. Smith, B. A. Francis, D. K. Dueker, R. D. Fechtner, J. S. Samples, J. S. Schuman *et al.*, "Optic nerve head and retinal nerve fiber layer analysis: a report by the american academy of ophthalmology," *Ophthalmology*, vol. 114, no. 10, pp. 1937–1949, 2007.

[5] Y. He, A. Carass, Y. Liu, B. M. Jedynak, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Structured layer surface segmentation for retina oct using fully convolutional regression networks," *Med. Image Anal.*, vol. 68, p. 101856, 2021.

[6] I. Ghorbel, F. Rossant, I. Bloch, S. Tick, and M. Paques, "Automated segmentation of macular layers in oct images and quantitative evaluation of performances," *Pattern Recognition*, vol. 44, no. 8, pp. 1590–1603, 2011.

[7] A. Yazdanpanah, G. Hamarneh, B. R. Smith, and M. V. Sarunic, "Segmentation of intra-retinal layers from optical coherence tomography images using an active contour approach," *IEEE Trans. Med. Imag.*, vol. 30, no. 2, pp. 484–496, 2010.

[8] R. Kafieh, H. Rabbani, M. D. Abramoff, and M. Sonka, "Intra-retinal layer segmentation of 3d optical coherence tomography using coarse grained diffusion map," *Med. Image Anal.*, vol. 17, no. 8, pp. 907–928, 2013.

[9] P. A. Dufour, L. Ceklic, H. Abdillahi, S. Schroder, S. De Dzanet, U. Wolf-Schnurrbusch, and J. Kowal, "Graph-based multi-surface segmentation of oct data using trained hard and soft constraints," *IEEE Trans. Med. Imag.*, vol. 32, no. 3, pp. 531–543, 2012.

[10] D. Xiang, H. Tian, X. Yang, F. Shi, W. Zhu, H. Chen, and X. Chen, "Automatic segmentation of retinal layer in oct images with choroidal neovascularization," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 5880–5891, 2018.

[11] L. Ngo, J. Cha, and J.-H. Han, "Deep neural network regression for automated retinal layer segmentation in optical coherence tomography images," *IEEE Trans. Image Process.*, vol. 29, pp. 303–312, 2019.

[12] A. G. Roy, S. Conjeti, S. P. K. Karri, D. Sheet, A. Katouzian, C. Wachinger, and N. Navab, "Relaynet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks," *Biomedical optics express*, vol. 8, no. 8, pp. 3627–3642, 2017.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*. Springer, 2015, pp. 234–241.

[14] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in Med. Image Anal. and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[15] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern recognition*, vol. 106, p. 107404, 2020.

[16] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021.

[17] H. Wang, S. Xie, L. Lin, Y. Iwamoto, X.-H. Han, Y.-W. Chen, and R. Tong, "Mixed transformer u-net for medical image segmentation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 2390–2394.

[18] 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:2105.05537*, 2021.

[19] X. Lin, Z. Yan, L. Yu, and K.-T. Cheng, "C2ftrans: Coarse-to-fine transformers for medical image segmentation," *arXiv preprint arXiv:2206.14409*, 2022.

[20] S. J. Chiu, X. T. Li, P. Nicholas, C. A. Toth, J. A. Izatt, and S. Farsiu, "Automatic segmentation of seven retinal layers in sdoct images congruent with expert manual segmentation," *Optics express*, vol. 18, no. 18, pp. 19 413–19 428, 2010.

[21] S. Karri, D. Chakraborthi, and J. Chatterjee, "Learning layer-specific edges for segmenting retinal layers with large deformations," *Biomedical optics express*, vol. 7, no. 7, pp. 2888–2901, 2016.

[22] A. Lang, A. Carass, M. Hauser, E. S. Sotirchos, P. A. Calabresi, H. S. Ying, and J. L. Prince, "Retinal layer segmentation of macular oct images using boundary classification," *Biomedical optics express*, vol. 4, no. 7, pp. 1133–1152, 2013.

[23] Y. He, A. Carass, Y. Liu, B. M. Jedynak, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Deep learning based topology guaranteed surface and mme segmentation of multiple sclerosis subjects from retinal oct," *Biomedical optics express*, vol. 10, no. 10, pp. 5042–5058, 2019.

[24] H. Zhang, J. Yang, K. Zhou, F. Li, Y. Hu, Y. Zhao, C. Zheng, X. Zhang, and J. Liu, "Automatic segmentation and visualization of choroid in oct with knowledge infused deep learning," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 12, pp. 3408–3420, 2020.

[25] J. Li, P. Jin, J. Zhu, H. Zou, X. Xu, M. Tang, M. Zhou, Y. Gan, J. He, Y. Ling *et al.*, "Multi-scale gcn-assisted two-stage network for joint segmentation of retinal layers and discs in peripapillary oct images," *Biomedical Optics Express*, vol. 12, no. 4, pp. 2204–2220, 2021.

[26] B. Wang, W. Wei, S. Qiu, S. Wang, D. Li, and H. He, "Boundary aware u-net for retinal layers segmentation in optical coherence tomography images," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 8, pp. 3029–3040, 2021.

[27] P. Jeihouni, O. Dehzangi, A. Amireskandari, A. Rezai, and N. M. Nasrabadi, "Multisdgan: translation of oct images to superresolved segmentation labels using multi-discriminators in multi-stages," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 4, pp. 1614–1627, 2021.

[28] A. Cazañas-Gordón and L. A. da Silva Cruz, "Multiscale attention gated network (magnet) for retinal layer and macular cystoid edema segmentation," *IEEE Access*, vol. 10, pp. 85 905–85 917, 2022.

[29] Y. Lee, J. Kim, J. Willette, and S. J. Hwang, "Mpvit: Multi-path vision transformer for dense prediction. in 2022 ieee," in *Proc. CVPR*, 2022, pp. 7277–7286.

[30] J. Li, T. Chen, R. Shi, Y. Lou, Y.-L. Li, and C. Lu, "Localization with sampling-argmax," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 236–27 248, 2021.

[31] W. Liu, T. Tian, W. Xu, H. Yang, X. Pan, S. Yan, and L. Wang, "Phtrans: Parallelly aggregating global and local representations for medical image segmentation," in *Proc. MICCAI*. Springer, 2022, pp. 235–244.

[32] W. Yu, M. Luo, and P. e. Zhou, "Metaformer is actually what you need for vision," in *Proc. CVPR*, 2022, pp. 10 819–10 829.

[33] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. CVPR*, 2021, pp. 15 750–15 758.

[34] Y. Tan, K.-F. Yang, S.-X. Zhao, and Y.-J. Li, "Retinal vessel segmentation with skeletal prior and contrastive loss," *IEEE Trans. Med. Imag.*, 2022.

[35] T. Li, P. Cao, Y. Yuan, L. Fan, Y. Yang, R. S. Feris, P. Indyk, and D. Katabi, "Targeted supervised contrastive learning for long-tailed recognition," in *Proc. CVPR*, 2022, pp. 6918–6928.

[36] S. J. Chiu, M. J. Allingham, P. S. Mettu, S. W. Cousins, J. A. Izatt, and S. Farsiu, "Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema," *Biomedical optics express*, vol. 6, no. 4, pp. 1172–1194, 2015.

[37] J. Tian, B. Varga, G. M. Somfai, W.-H. Lee, W. E. Smiddy, and D. Cabrera DeBuc, "Real-time automatic segmentation of optical coherence tomography volume data of the macular region," *PloS one*, vol. 10, no. 8, p. e0133908, 2015.

[38] Y. He, A. Carass, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Retinal layer parcellation of optical coherence tomography images: Data resource for multiple sclerosis and healthy controls," *Data in brief*, vol. 22, pp. 601–604, 2019.

[39] H. Fang, F. Li, H. Fu, J. Wu, X. Zhang, and Y. Xu, "Dataset and evaluation algorithm design for goals challenge," in *International Workshop on Ophthalmic Med. Image Anal.* Springer, 2022, pp. 135–142.

[40] S. J. Song, M. Y. Lee, and W. E. Smiddy, "Ganglion cell layer thickness and visual improvement after epiretinal membrane surgery," *Retina*, vol. 36, no. 2, pp. 305–310, 2016.