Growth Simulation Network for Polyp Segmentation

Hongbin Wei[1], Xiaoqi Zhao[1], Long Lv[2,3], Lihe Zhang[1,✉], Weibing Sun[2,3], Huchuan Lu[1]

[1] Dalian University of Technology, China
{weihongbin,zxq}@mail.dlut.edu.cn, {zhanglihe,lhchuan}@dlut.edu.cn,
[2] Affiliated Zhongshan Hospital of Dalian University, China
[3] Key Laboratory of Microenvironment Regulation and Immunotherapy of Urinary Tumors in Liaoning Province, China
lvlong113@126.com, weibingsun_dyfemw@163.com,

Abstract. Colonoscopy is a gold standard, while automated polyp segmentation can minimize missed rates and timely treatment of colon cancer at an early stage. But most existing polyp segmentation methods have borrowed techniques related to image semantic segmentation, and the main idea is to extract and fuse feature information of images more effectively. As we know, polyps naturally grow from small to large, thus they have strong rules. In view of this trait, we propose a Growth Simulation Network (GSNet) to segment polyps from colonoscopy images. First, the completeness map (i.e., ground-truth mask) is decoupled to generate Gaussian map and body map. Among them, Gaussian map is mainly used to locate polyps, while body map expresses the intermediate stages, which helps filter redundant information. GSNet has three forward branches, which are supervised by Gaussian map, body map and completeness map, respectively. What's more, we design a dynamic attention guidance (DAG) module to effectively fuse the information from different branches. Extensive experiments on five benchmark datasets demonstrate that our GSNet performs favorably against most state-of-the-art methods under different evaluation metrics. The source code will be publicly available at https://github.com/wei-hongbin/GSNet

Keywords: Colorectal cancer· Automatic polyp segmentation· Growth simulation· Dynamic attention guidance.

1 Introduction

Colorectal cancer (CRC) is the third most common cancer worldwide, and it is considered to be the second most deadly cancer, accounting for 9.4% of all cancer deaths [18]. It usually develops from small, noncancerous clumps of cells called polyps in the colon. Hence, the best way to prevent CRC is to accurately identify and remove polyps in advance. Currently, colonoscopy is the primary method for prevention of colon cancer [19], but the rate of missed polyps is high because of the large variation (shape, size and texture) of polyps and the

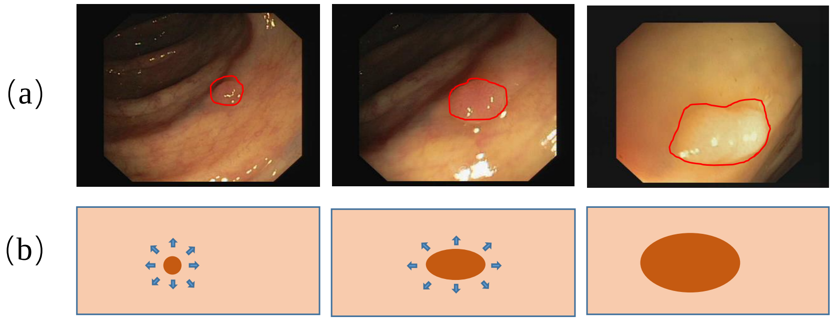

Fig. 1. Polyps in different stages. (a) From left to right, they indicate anterior, middle and late polyps, respectively. (b) Diagram of natural polyp growth.

high dependence on professional doctors. Clinical studies have shown that the rate of missing colon polyps during endoscopy may range from 20% to 47% [1]. Therefore, an accurate and efficient approach of automatically identifying and segmenting polyps is highly demanded to improve the outcome of colonoscopy.

Earlier automatic polyp segmentation models [22] mainly rely on hand-crafted features (e.g., color and texture), which are difficult to capture the global context information and have poor robustness. Recently, polyp segmentation algorithms based on deep learning have made good progress. For example, UNet [17], UNet++ [34] and ResUNet++ [9] have achieved higher accuracy than traditional methods. However, these methods only provide a better treatment of the polyp region but easily lead to a blurred boundary. To improve the boundary accuracy, PraNet [7] uses the reverse attention module to establish the relationship between region and boundary clues to obtain a more ideal boundary. To further improve model performance, many models consider targeted treatment of different levels of encoder features because there is a large semantic gap between high-level and low-level features. Polyp-PVT[3] obtains semantic and location information from high-level features and detail information from low-level features, and finally fuses them to improve the expression of features. Furthermore, MSNet [33,29] removes the redundant information in the every levels through multi-level subtraction, and effectively obtains the complementary information from lower order to higher order between different levels, so as to comprehensively enhance the perception ability.

Many above methods draw lessons from the general idea of semantic segmentation [31,32,14,30,15,13], that is, extracting and fusing feature information effectively. However, they are not targeted to the task of polyp segmentation. Polyps grow naturally from small to large, as shown in Figure 1, so this scene has strong laws. Compared with other scenes (e.g., animals and plants) of nature, the regions of polyps are internally continuous, and their boundaries are blurred, without sharp edges.

In the Encoder-Decoder architecture, there are some similarities between the learning process of decoder and the natural growth process of polyps, which progress from simple to complex and from rough to fine. Inspired by this, we propose a novel growth simulation network (GSNet) to achieve polyp segmen-

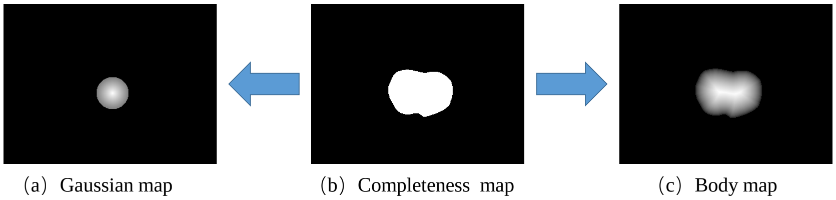

Fig. 2. Supervisions at different stages. Both Gaussian map and body map are obtained by decoupling completeness map.

tation. First, the completeness map is decoupled to generate Gaussian map and body map, as shown in Figure 2. Compared to the completeness map, the Gaussian map mainly characterizes the location cues and the body map describes more about the intermediate stages of polyp growth. Then, three feature learning branches are built, which are Gaussian branch, body branch and prediction branch. The Gaussian branch only uses high-level features to learn location information. The body branch uses more features and incorporates Gaussian branch features to learn main body representation. The prediction branch uses all-level features and body branch features to form the final prediction. The differentiated utilization of features in different branches can avoid introducing redundant information.

The information of the previous branch is used by the later branch. In order to effectively fuse it, we design a dynamic attention guidance (DAG) module, which forms the dynamic convolution kernel from the supervised feature of the previous branch to guide the feature learning of the later branch. Compared with conventional convolution, which takes the same parameters for all inputs, dynamic convolution can automatically change the convolution parameters and perform well when the model has strong correlation with input data. CondConv [28] and DyConv [4] are the earliest works about dynamic convolution, which are both implemented by generating different weights according to the input data, and then weighting and summing multiple convolution kernels.The above works all deal with one input. HDFNet [12] exploits dynamic convolution to fuse two strongly correlated features. It generates a convolution kernel based on one input and then performs convolution operation on the other input. Inspired by HDFNet, our DAG module makes substantial improvements. We first simplifies the generation process of dynamic convolution kernels, which greatly reduces the number of parameters and computational effort. Meanwhile, we borrow the non-local [25] idea and utilize dynamic convolution to generate the attention, thereby achieving information fusion of two input features.

In order to better restore its natural growth state, we further propose a Dynamic Simulation Loss (DSLoss), which makes the weight of each branch in the loss function change dynamically in different stages of training. Our main contributions can be summarized as follows:

- According to the characteristics of polyp images, we propose a novel growth simulation network, which is cascaded from Gaussian map to body map and finally forms a segmentation map. Meanwhile, the network avoids introducing redundant information and the training process is more smooth.

- To effectively fuse different branch features, we propose a dynamic attention guidance module, which can dynamically use features from the previous level to guide the feature generation of the next level.
- The GSNet can accurately segment polyps. Extensive experiments demonstrate that our GSNet achieves the state-of-the-art performance under different evaluation metrics on five challenging datasets.

2 The Proposed Method

2.1 Gaussian Map and Body Map

To eliminate redundant information and enable each branch to have a different focus, we decouple the completeness map to generate Gaussian map and body map, as shown in Figure 2. Firstly, we introduce the generation of Gaussian map. Let I be a binary GT map, which can be divided into two parts: foreground I_{fg} and background I_{bg}. For each pixel p, $I(p)$ is its label value. If $p \in I_{fg}$, $I(p)$ equals 1, and 0 if $p \in I_{bg}$. All the foreground pixels are critically enclosed by a rectangular box. We define a transformation function $I'(p)$ to measure the distance between pixel p and center point p_c of the rectangle.

$$I'(p) = e^{\frac{-0.5\times\|p-p_c\|_2}{R}}, \tag{1}$$

where $L2$-norm denotes the Euclidean distance between two pixels, and R denotes the minimum side length of the rectangle. We normalize the values I' of all pixels using a simple linear function:

$$\bar{I}' = \frac{I' - min(I')}{max(I') - min(I')}. \tag{2}$$

Thus, the Gaussian map is obtained by simple thresholding as follows:

$$O_{Gaussian} = \begin{cases} \bar{I}'(p), & \bar{I}'(p) > 0.5, \\ 0, & \bar{I}'(p) \leq 0.5. \end{cases} \tag{3}$$

For the body map, the transformation process refers to LDFNet [27]. We firstly compute the Euclidean distance $f(p,q)$ between pixels p and q. If pixel p belongs to the foreground, the function will first look up its nearest pixel q in the background and then use $f(p,q)$ to calculate the distance. If pixel p belongs to the background, their minimum distance is set to zero. The transformation can be formulated as:

$$I^*(p) = \begin{cases} \min_{q\in I_{bg}} f(p,q), & p \in I_{fg}, \\ 0, & p \in I_{bg}. \end{cases} \tag{4}$$

Then, the values I^* of all pixels are normalized by formula (2). To remove the interference of background, the body map is obtained as:

$$O_{body} = I \odot \bar{I}^*, \tag{5}$$

where $\odot$ denotes element-wise product.

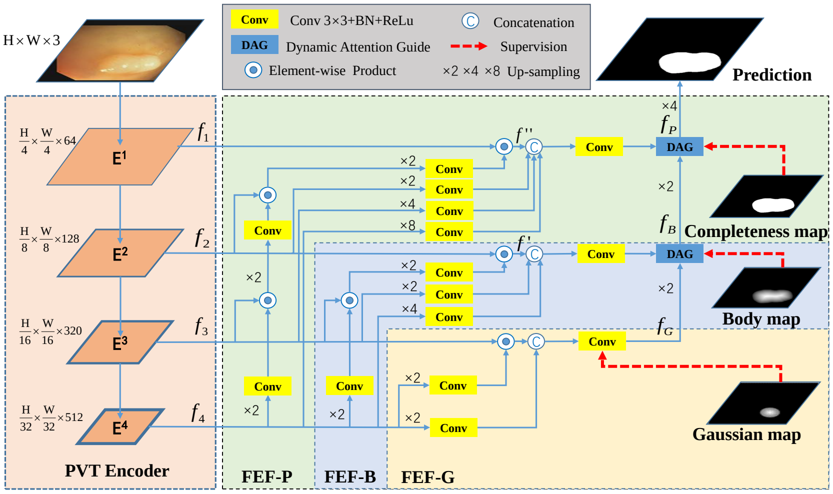

Fig. 3. The overall architecture of GSNet, which consists of a PVT encoder, features extraction and fusion (FEF) module including three branches FEF-G, FEF-B and FEF-P, and dynamic attention guidance (DAG) module for integrating and guiding features of different branches.

2.2 Overall Architecture

As shown in Figure 3, the proposed GSNet consists of three key modules: pyramid vision transformer (PVT) encoder, feature extraction and fusion (FEF) module and dynamic attention guidance (DAG) module. Specifically, the PVT is used to extract multi-scale long-range dependencies features from the input image. The FEF is employed to collect the semantic cues through aggregating features in a progressive manner, which contains three different branches FEF-G, FEF-B and FEF-P. The DAG is adopted to fuse the features provided by the different FEF branches, effectively using simpler features to dynamically guide the generation of more complex features.

We adopt the PVTv2 [24] as the encoder to extract pyramid features and adjust their channel number to 32 through four convolutional units to obtain $f_i, i \in \{1, 2, 3, 4\}$, which are then fed to FEF. First, FEF-G fuses high-level features (f_3, f_4) and is supervised by Gaussian map, which mainly focuses on location. Second, FEF-B fuses more features (f_2, f_3, f_4) and is guided by the FEF-G to generate the body map, which already contains main body information. Finally, FEF-P fuses all features (f_1, f_2, f_3, f_4) and is guided by FEF-B to generate the final prediction. In particular, the DAG plays a role of guidance for full information fusion of multiple segmentation cues. During training, we optimize the model with a dynamic simulation loss, which makes the weight of each branch loss change adaptively along iteration step.

2.3 Features Extraction and Fusion Module

It is well known that the different levels of encoder features contain different types of information, which is that high-level features contain more abstract information and low-level features often contain rich detail information. FEF has three branches FEF-G, FEF-B and FEF-P, which fuse features of different levels and are supervised by Gaussian map, body map and completeness map. Therefore, they focus on location information, body information and all information, respectively. Specific details are as follows.

FEF-G Its main purpose is to extract location information. To balance the accuracy and computational resources, only high-level features f_3 and f_4 are used. As shown in Figure 3, the process is formulated as follows:

$$f_G = F(Cat(F(up(f_4)) \odot f_3, F(up(f_4)))), \tag{6}$$

where $\odot$ denotes the element-wise product, $up(\cdot)$ indicates up-sampling, and $Cat(\cdot)$ is the concatenation operation. $F(\cdot)$ is defined as a convolutional unit composed of a 3×3 convolutional layer with padding of 1, batch normalization and ReLU, and it is parameter independent.

FEF-B Its main purpose is to extract body information. It combines high-level features (f_3 and f_4) and mid-level feature f_2. First, these features are preliminarily aggregated to obtain a rich semantic representation as

$$f' = F(F(f_4) \odot f_3) \odot f_2. \tag{7}$$

Then, DAG utilizes f_G to guide further information combination,

$$f_B = DAG(f_G, F(Cat(F(up(f_4)), F(up(f_3)), f'))). \tag{8}$$

In this process, the features f_3 and f_4 are up-sampled and processed through convolution operation, which has the effect of residual learning.

FEF-P It generates final prediction by using all level features (f_1, f_2, f_3 and f_4). Among them, low-level features are important to reconstruct details. Similar to FEF-B, the computation process can be written as

$$f'' = F(F(F(f_4) \odot f_3) \odot f_2) \odot f_1. \tag{9}$$

Then, DAG utilizes f_B to guide the combination of the above features,

$$f_P = DAG(f_B, F(Cat(F(f_4), F(f_3), F(f_2), f''))). \tag{10}$$

Finally, f_G, f_B and f_P are up-sampled to the same size as the input image and supervised by three different maps.

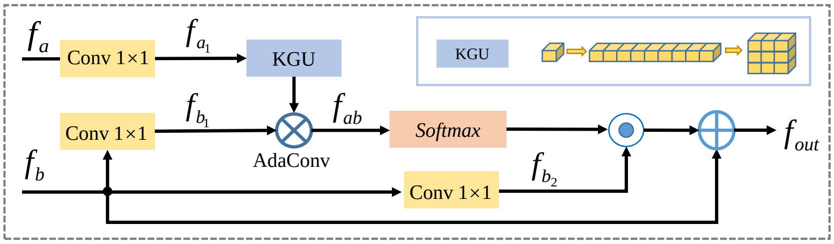

Fig. 4. Details of the DAG. KGU generates adaptive kernel tensors by adjusting convolutional channel, and then reshapes them to the regular form of convolution kernel.

2.4 Dynamic Attention Guidance Module

The proposed network generates predictions in sequence from Gaussian map to body map and finally to prediction map. They are closely related. The former is the basis of the latter and the latter is an extension of the former. Since dynamic convolution can automatically change the convolution parameters according to inputs, we propose a dynamic attention guidance (DAG) module combining dynamic convolutions and attention mechanisms, which can leverage the simpler feature map to guide the blended features to produce a more complete feature.

Given feature map f_a with guidance information and f_b with rich semantic information, we fuse them through DAG. First, they pass a linear mapping process of 1×1 convolution operation to generate f_{a_1}, f_{b_1} and f_{b_2}, respectively. Second, we use kernel generation unit (KGU) [12] to yield dynamic kernels based on f_{a_1}. Then, f_{b_1} is adaptively filtered to obtain guidance feature f_{ab},

$$f_{ab} = KGU(f_{a_1}) \otimes f_{b_1}, \tag{11}$$

where $KGU(\cdot)$ denote the operations of the KGU module. $\otimes$ is an adaptive convolution operation. Finally, the feature combination is achieved as follows:

$$f_{out} = f_b \oplus (softmax(f_{ab}) \odot f_{b_2}), \tag{12}$$

where $\oplus$ and $\odot$ denote the element-wise addition and multiplication, respectively.

2.5 Dynamic Simulation Loss

The total training loss function can be formulated as follows:

$$L_{total} = W_g * L_g + W_b * L_b + W_p * L_p, \tag{13}$$

where L_g, L_b and L_p separately represent loss of Gaussian branch, body branch and prediction branch. $L_g = L_b = L^w_{BCE}$ is standard binary cross entropy (BCE) loss, while $L_p = L^w_{IoU} + L^w_{BCE}$ is the weighted intersection over union (IoU) loss and weighted BCE loss, which has been widely adopted in segmentation tasks [16,26]. W_g, W_b and W_p are the weighting coefficients of three branch losses, respectively. The expression is as follows:

$$W_g = e^{\frac{-x}{(n/2)}}, W_b = 1/(e^{\frac{-x}{(n/5)}} + 1), W_p = 1, \tag{14}$$

Datasets	Methods	$mDice \uparrow$	$mIoU \uparrow$	$F_{\beta}^{\omega} \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi}^{max} \uparrow$	$MAE \downarrow$
ColonDB	U-Net(MICCAI'15)	0.512	0.444	0.498	0.712	0.696	0.061
	U-Net++(DLMIA'18)	0.483	0.410	0.467	0.691	0.680	0.064
	PraNet(MICCAI'20)	0.712	0.640	0.699	0.820	0.847	0.043
	SANet(MICCAI'21)	0.753	0.670	0.726	0.837	0.869	0.043
	Polyp-PVT(CAAI AIR'23)	0.808	0.727	0.795	0.865	0.913	0.031
	GSNet(Ours)	**0.822**	**0.746**	**0.805**	**0.874**	**0.922**	**0.026**
ETIS	U-Net(MICCAI'15)	0.398	0.335	0.366	0.684	0.643	0.036
	U-Net++(DLMIA'18)	0.401	0.344	0.390	0.683	0.629	0.035
	PraNet(MICCAI'20)	0.628	0.567	0.600	0.794	0.808	0.031
	SANet(MICCAI'21)	0.750	0.654	0.685	0.849	0.881	0.015
	Polyp-PVT(CAAI AIR'23)	0.787	0.706	0.750	0.871	**0.906**	**0.013**
	GSNet(Ours)	**0.802**	**0.724**	**0.763**	**0.871**	**0.906**	**0.013**
Kvasir	U-Net(MICCAI'15)	0.818	0.746	0.794	0.858	0.881	0.055
	U-Net++(DLMIA'18)	0.821	0.743	0.808	0.862	0.886	0.048
	PraNet(MICCAI'20)	0.898	0.840	0.885	0.915	0.944	0.030
	SANet(MICCAI'21)	0.904	0.847	0.892	0.915	0.949	0.028
	Polyp-PVT(CAAI AIR'23)	0.917	0.864	0.911	0.925	0.956	0.023
	GSNet(Ours)	**0.930**	**0.883**	**0.923**	**0.934**	**0.968**	**0.020**
CVC-T	U-Net(MICCAI'15)	0.710	0.672	0.684	0.843	0.847	0.022
	U-Net++(DLMIA'18)	0.707	0.624	0.687	0.839	0.834	0.018
	PraNet(MICCAI'20)	0.871	0.797	0.843	0.925	0.950	0.010
	SANet(MICCAI'21)	0.888	0.815	0.859	0.928	0.962	0.008
	Polyp-PVT(CAAI AIR'23)	0.900	0.833	0.884	0.935	**0.973**	**0.007**
	GSNet(Ours)	**0.909**	**0.844**	**0.889**	**0.939**	0.972	**0.007**
ClinicDB	U-Net(MICCAI'15)	0.823	0.755	0.811	0.889	0.913	0.019
	U-Net++(DLMIA'18)	0.794	0.729	0.785	0.873	0.891	0.022
	PraNet(MICCAI'20)	0.899	0.849	0.896	0.936	0.963	0.009
	SANet(MICCAI'21)	0.916	0.859	0.909	0.939	0.971	0.012
	Polyp-PVT(CAAI AIR'23)	0.937	0.889	0.936	0.949	**0.985**	**0.006**
	GSNet(Ours)	**0.942**	**0.898**	**0.937**	**0.950**	0.984	**0.006**

Table 1. Quantitative comparison. ↑ and ↓ indicate that the larger and smaller scores are better, respectively. The best results are shown in **bolded font**.

where n represents the number of all epochs. The change is consistent with the growth law of polyp growth, that is, at the beginning, there is only the position information of small polyps. With the continuous growth of polyps, the position information remains basically stable, other information will be more important. Because the change of weight is very small when it is subdivided into each epoch, it will not affect the optimization of network back propagation.

3 Experiments

3.1 Settings

Datasets and Evaluation Metrics Following the experimental setups in PraNet, we evaluate the proposed model on five benchmark datasets: ClinicDB [2], Kvasir [8], ColonDB [21], CVC-T [23] and ETIS [20]. To keep the fairness of the experiments, we adopt the same training set with PraNet, that is, 550 samples from the ClinicDB and 900 samples from the Kvasir are used for training. The remaining images and other three datasets are used for testing.

We adopt six widely-used metrics for quantitative evaluation: mean Dice, mean IoU, the weighted F-measure (F_{β}^{ω}) [11], mean absolute error (MAE), the recently released S-measure S_{α} [5] and E-measure(E_{ϕ}^{max}) [6] scores. The lower value is better for the MAE and the higher is better for others.

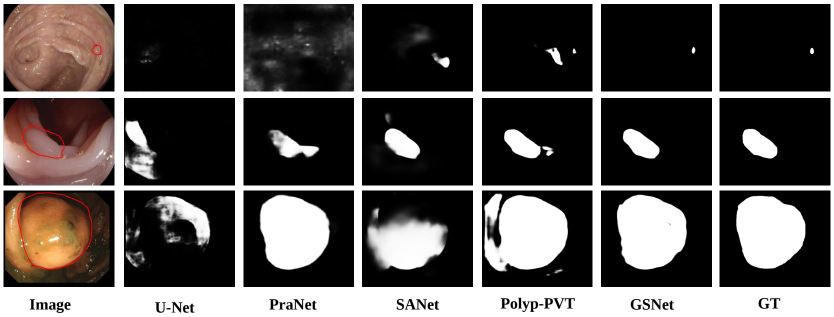

Fig. 5. Visual comparison of different methods.

Implementation Details Our model is implemented based on the PyTorch framework and trained on a single V100 GPU for 100 epochs with mini-batch size 16. We resize the inputs to 352 × 352 and employ a general multi-scale training strategy as the Polyp-PVT. For the optimizer, we adopt the AdamW [10], which is widely used in transformer networks. The learning rate and the weight decay all are adjusted to 1e-4. For testing, we resize the images to 352 × 352 without any post-processing optimization strategies.

3.2 Comparisons with State-of-the-art

To prove the effectiveness of the proposed GSNet, as shown in Table 1, five state-of-the-art models are used for comparison. On the five challenging datasets, our GSNet achieves the best performance against other methods. We also demonstrate qualitative results of different methods in Figure 5.

Bas.	FEF-P	FEF-B	FEF-G	DAG	ETIS				Kvasir			
					mDice	mIoU	F_β^ω	E_ϕ^{max}	mDice	mIoU	F_β^ω	E_ϕ^{max}
√					0.725	0.650	0.680	0.838	0.905	0.854	0.893	0.955
√	√				0.752	0.679	0.707	0.850	0.911	0.960	0.896	0.956
√	√	√		√	0.774	0.700	0.739	0.890	0.922	0.875	0.912	0.958
√	√	√	√		0.768	0.689	0.721	0.874	0.919	0.867	0.907	0.960
√	√	√	√	√	0.802	0.724	0.763	0.906	0.930	0.883	0.923	0.968
Only using CM as supervision					0.755	0.680	0.712	0.862	0.921	0.871	0.909	0.960

Table 2. Ablation study.

3.3 Ablation Study

We take the PVTv2 as the baseline to analyze the contribution of each component. The results are shown in Table 2.

Effectiveness of FEF and DAG By the comparisons of Row 1 vs. Row 2, Row 2 vs. Row 3, and Row 3 vs. Row 5, we can see that all three branches of FEF-G,

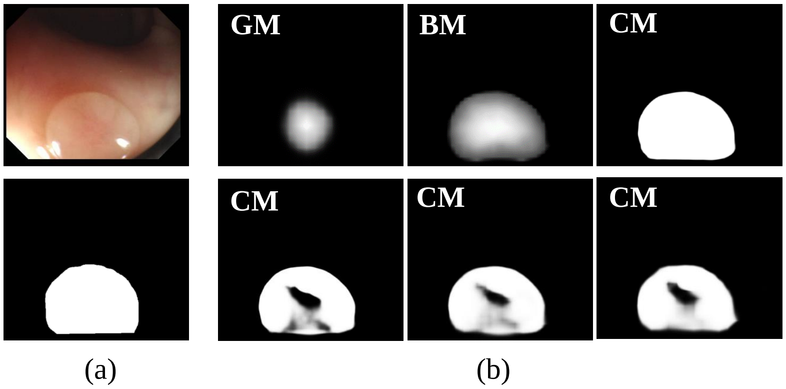

Fig. 6. Illustration of the benefits of using three decoupled maps. (a) Image and ground truth. (b) Visual comparison of different branches. The first row is the results of supervision by Gaussian, body and completeness maps, and the second row shows the result supervised by three completeness maps.

FEF-B and FEF-G are effective. We remove the DAG module and use simple fusion (element-wise addition and convolution) to combine these branches. The results are shown in Row 4. By comparing Row 4 and Row 5, we can see that the DAG works. Since no data in the ETIS is used for training, the significant gains on this dataset indicate that the designs of FEF and DAG greatly enhance the generalization of the model. Some images in the Kvasir are used as training data, hence the baseline model performs much better, which results in that the performance improvement on this dataset is not as obvious as that on the ETIS.

Effectiveness of Decoupled Maps We use the completeness map to supervise all three branches. The results are listed in Row 6 of Table 2. Comparing to Row 5, we find that the latter performs better than the former with a gain of 4.7% in terms of mDice on the ETIS.

The proposed GSNet structure is relatively simple and the different features are repeatedly introduced in the three branches. Gaussian branch and body branch mainly use high-level abstract features with small resolution, which does not learn the complete prediction perfectly. Enforcing the usage of the completeness map easily leads to the learning of broken features and introduces redundant information, as shown in Figure 6.

4 Conclusion

In this work, we propose a novel growth simulation network (GSNet) for polyp segmentation. According to the characteristics of polyp images, we propose to decouple the completeness map into Gaussian map and body map and use different map parallel branches to learn, respectively. These labels contain partial information, which reduces the learning difficulty of the model. The Gaussian branch mainly learns the location, body branch focuses more on the main information, and then the prediction branch to refine the details to get the final prediction map. In this way, the entire learning process progresses from simple to complex. Simulating the polyp growth process, DSLoss dynamically adjusts the weight to further boost the performance during training.

Acknowledgements. This work was supported by the National Natural Science Foundation of China # 62276046 and the Liaoning Natural Science Foundation # 2021-KF-12-10.

References

1. Ahn, S.B., Han, D.S., Bae, J.H., Byun, T.J., Kim, J.P., Eun, C.S.: The miss rate for colorectal adenoma determined by quality-adjusted, back-to-back colonoscopies. Gut and liver **6**(1), 64 (2012)
2. Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., Vilariño, F.: Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. Computerized medical imaging and graphics **43**, 99–111 (2015)
3. Bo, D., Wenhai, W., Deng-Ping, F., Jinpeng, L., Huazhu, F., Ling, S.: Polyp-pvt: Polyp segmentation with pyramidvision transformers (2023)
4. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11030–11039 (2020)
5. Fan, D.P., Cheng, M.M., Liu, Y., Li, T., Borji, A.: Structure-measure: A new way to evaluate foreground maps. In: Proceedings of the IEEE international conference on computer vision. pp. 4548–4557 (2017)
6. Fan, D.P., Gong, C., Cao, Y., Ren, B., Cheng, M.M., Borji, A.: Enhanced-alignment measure for binary foreground map evaluation. arXiv preprint arXiv:1805.10421 (2018)
7. Fan, D.P., Ji, G.P., Zhou, T., Chen, G., Fu, H., Shen, J., Shao, L.: Pranet: Parallel reverse attention network for polyp segmentation. In: International conference on medical image computing and computer-assisted intervention. pp. 263–273. Springer (2020)
8. Jha, D., Smedsrud, P.H., Riegler, M.A., Halvorsen, P., Lange, T.d., Johansen, D., Johansen, H.D.: Kvasir-seg: A segmented polyp dataset. In: International Conference on Multimedia Modeling. pp. 451–462. Springer (2020)
9. Jha, D., Smedsrud, P.H., Riegler, M.A., Johansen, D., De Lange, T., Halvorsen, P., Johansen, H.D.: Resunet++: An advanced architecture for medical image segmentation. In: 2019 IEEE International Symposium on Multimedia (ISM). pp. 225–2255. IEEE (2019)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
11. Margolin, R., Zelnik-Manor, L., Tal, A.: How to evaluate foreground maps? In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 248–255 (2014)
12. Pang, Y., Zhang, L., Zhao, X., Lu, H.: Hierarchical dynamic filtering network for rgb-d salient object detection. In: European Conference on Computer Vision. pp. 235–252. Springer (2020)

13. Pang, Y., Zhao, X., Xiang, T.Z., Zhang, L., Lu, H.: Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In: CVPR. pp. 2160–2170 (2022)
14. Pang, Y., Zhao, X., Zhang, L., Lu, H.: Multi-scale interactive network for salient object detection. In: CVPR. pp. 9413–9422 (2020)
15. Pang, Y., Zhao, X., Zhang, L., Lu, H.: Caver: Cross-modal view-mixed transformer for bi-modal salient object detection. IEEE TIP (2023)
16. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., Jagersand, M.: Basnet: Boundary-aware salient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7479–7489 (2019)
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
18. Siegel, R.L., Miller, K.D., Goding Sauer, A., Fedewa, S.A., Butterly, L.F., Anderson, J.C., Cercek, A., Smith, R.A., Jemal, A.: Colorectal cancer statistics, 2020. CA: a cancer journal for clinicians **70**(3), 145–164 (2020)
19. Siegel, R.L., Torre, L.A., Soerjomataram, I., Hayes, R.B., Bray, F., Weber, T.K., Jemal, A.: Global patterns and trends in colorectal cancer incidence in young adults. Gut **68**(12), 2179–2185 (2019)
20. Silva, J., Histace, A., Romain, O., Dray, X., Granado, B.: Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. International journal of computer assisted radiology and surgery **9**(2), 283–293 (2014)
21. Tajbakhsh, N., Gurudu, S.R., Liang, J.: Automated polyp detection in colonoscopy videos using shape and context information. IEEE transactions on medical imaging **35**(2), 630–644 (2015)
22. Tajbakhsh, N., Gurudu, S.R., Liang, J.: Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks. In: 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI). pp. 79–83. IEEE (2015)
23. Vázquez, D., Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., López, A.M., Romero, A., Drozdzal, M., Courville, A.: A benchmark for endoluminal scene segmentation of colonoscopy images. Journal of healthcare engineering **2017** (2017)
24. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 568–578 (2021)
25. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
26. Wei, J., Wang, S., Huang, Q.: F^3net: fusion, feedback and focus for salient object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12321–12328 (2020)
27. Wei, J., Wang, S., Wu, Z., Su, C., Huang, Q., Tian, Q.: Label decoupling framework for salient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13025–13034 (2020)
28. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: Conditionally parameterized convolutions for efficient inference. Advances in Neural Information Processing Systems **32** (2019)
29. Zhao, X., Jia, H., Pang, Y., Lv, L., Tian, F., Zhang, L., Sun, W., Lu, H.: M2snet: Multi-scale in multi-scale subtraction network for medical image segmentation. arXiv preprint arXiv:2303.10894 (2023)
30. Zhao, X., Pang, Y., Zhang, L., Lu, H.: Joint learning of salient object detection, depth estimation and contour extraction. IEEE TIP **31**, 7350–7362 (2022)

31. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Suppress and balance: A simple gated network for salient object detection. In: ECCV. pp. 35–51 (2020)
32. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Towards diverse binary segmentation via a simple yet general gated network. arXiv preprint arXiv:2303.10396 (2023)
33. Zhao, X., Zhang, L., Lu, H.: Automatic polyp segmentation via multi-scale subtraction network. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 120–130. Springer (2021)
34. Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support, pp. 3–11. Springer (2018)