— Supplemental Materials —

MFP: Making Full Use of Probability Maps for Interactive Image Segmentation

Chaewon Lee
Korea University
chaewonlee@mcl.korea.ac.kr

Seon-Ho Lee
Korea University
seonholee@mcl.korea.ac.kr

Chang-Su Kim
Korea University
changsukim@korea.ac.kr

A. Implementation Details

A.1. Network Architecture

Given an image $I \in \mathbb{R}^{H\times W\times 3}$ and clicks from a user, we encode the clicks into the click map $C^t \in \mathbb{R}^{H\times W\times 2}$ for segmentation round t, by representing each click with a fixed-size disk. We then concatenate the click map with the previous probability map $P^{t-1} \in \mathbb{R}^{H\times W\times 1}$ and the modulated probability map $\tilde{P}^{t-1} \in \mathbb{R}^{H\times W\times 1}$, making 4-channel input. Then, depending on the segmentation backbone, we embed the image I and the 4-channel input into tensors of the same size, respectively, through either patch embedding layers or convolution layers. These two tensors are element-wise summed and conveyed into the backbone, thus yielding a backbone feature $\mathcal{F}_B^t \in \mathbb{R}^{H/4\times W/4\times K}$. We also concatenate I, P^{t-1} and $\tilde{P}^{t-1}$ and process them through convolution layers to yield a probability-related feature $\mathcal{F}_P^t \in \mathbb{R}^{H/4\times W/4\times K}$. Next, we concatenate $\mathcal{F}_B^t$ and $\mathcal{F}_P^t$ and fuse them through convolution layers. Finally, the segmentation head takes the fused feature as input and generates the current probability map P^t. Thresholding P^t yields the final segmentation mask Y^t .

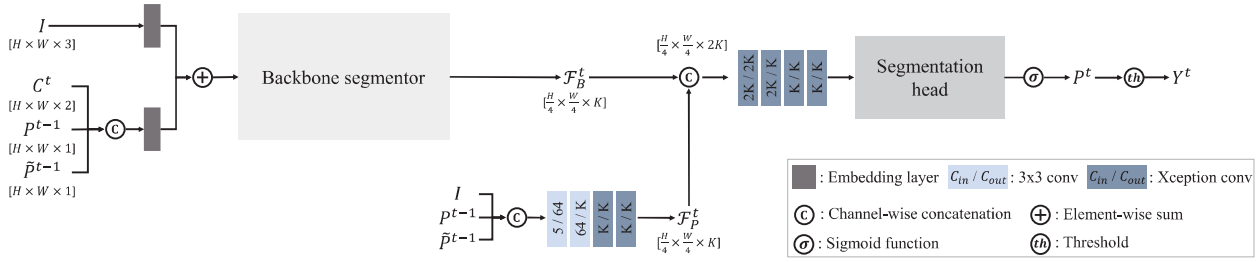


Figure S-1. Network architecture of the proposed MFP algorithm.

A.2. Training Details

For the ViT-B backbone, we initialize a plain ViT with MAE pretrained weights [4], as done in [10]. We train the network for 55 epochs using an initial learning rate of 5×10^{-5}. We reduce the learning rate by a factor of $\frac{1}{10}$ in the 10th epoch. For HRNet-18, we train the model for 130 epochs with an initial learning rate of 5×10^{-4}. We reduce the learning rate by a factor of $\frac{1}{10}$ in the 100th and 115th epochs, respectively. For ResNet-34, we train the model for 80 epochs with a learning rate of 5×10^{-4}. Table S-1 summarizes the training configurations.

Table S-1. Configurations for training network.

Backbone	$H \times W$	K	# of epochs	Learning rate	Batch size
ResNet-34	384x384	128	80	5×10^{-4}	16
HRNet-18	320x480	270	130	5×10^{-4}	16
ViT-B	448x448	256	55	5×10^{-5}	8

B. Additional Comparison Results

B.1. Comparison of IoU Performances of Models Trained on COCO-LVIS

Due to limited space, in the main paper, we compare the IoU results of the models trained on SBD only. Here, we also compare the models trained on the COCO+LVIS datasets [2, 9]. Figure S-2 shows the IoU ratios according to the number of clicks. We also report the area under the curve (AUC) in the legends. The proposed algorithm again provides the highest AUC scores on all four evaluation datasets.

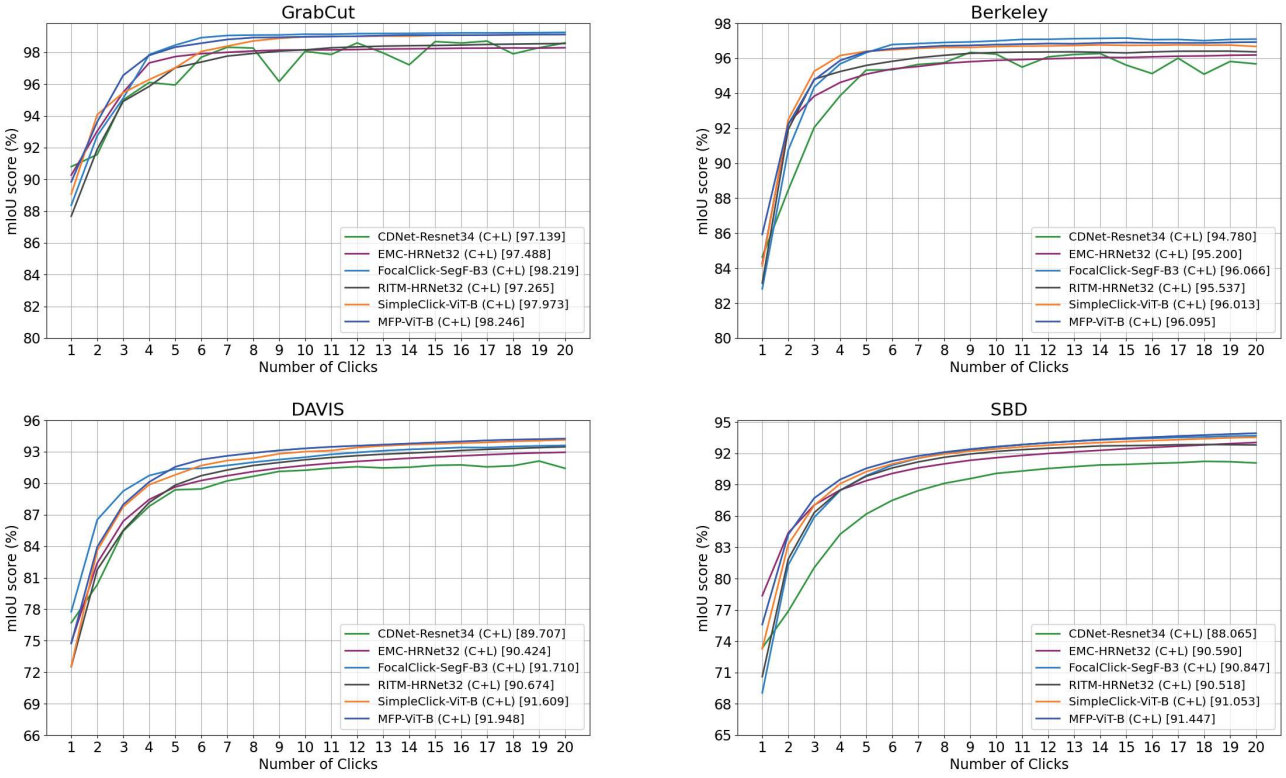


Figure S-2. Comparison of the mean IoU scores according to the number of clicks on the GrabCut[13], Berkeley[11], DAVIS[12], and SBD[3] datasets. The models are trained on the COCO+LVIS datasets [2, 9]. The legend contains the AUC score for each algorithm.

B.2. Comparison of algorithms using the ViT-B Backbone Trained on COCO-LVIS

Table S-2 provides comparison results with algorithms using the same ViT-B backbone network, trained on the COCO+LVIS datasets. Out of 12 NoC scores compared, MFP shows the best results in 8 tests. AdaptiveClick shows comparable results to the proposed MFP. However, it requires 19.3M more parameters and 88.51G more FLOPs than MFP, when the same backbone of ViT-B is used.

Table S-2. Comparison of the proposed MFP algorithm with conventional algorithms using the same ViT-B backbone. All algorithms are trained on the COCO + LVIS datasets.

		GrabCut			Berkeley			DAVIS			SBD		
Algorithm	Backbone	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
InterFormer [5]	ViT-B	1.38	1.50	2.68	1.99	3.14	6.69	4.10	6.19	12.51	3.78	6.34	12.43
iCMFormer [7]	ViT-B	1.42	1.52	-	1.40	1.86	-	3.40	5.06	-	3.29	**5.30**	-
AdaptiveClick [8]	ViT-B	**1.34**	1.48	**1.68**	1.40	**1.83**	5.10	3.40	**4.81**	9.26	**3.25**	5.37	11.83
SimpleClick [10]	ViT-B	1.38	1.48	1.80	1.36	1.97	5.05	3.66	5.06	10.04	3.43	5.62	11.92
MFP (Proposed)	ViT-B	**1.34**	**1.42**	1.70	**1.35**	1.90	**4.68**	**3.37**	**4.81**	**9.23**	3.26	5.34	**11.65**

B.3. Qualitative Comparsions

Figures S-3 and S-4 compare the proposed algorithm with SimpleClick[10]. As mentioned in the main paper, we adopt the automatic clicking strategy used in [1, 6, 14]. Hence, in the two algorithms, the clicks are placed at different locations. However, for an even fairer comparison, we provide segmentation results for the cases in which the two algorithms get clicks in as similar locations as possible. Also, the number of clicks in the proposed algorithm is either equal to or less than that in SimpleClick.

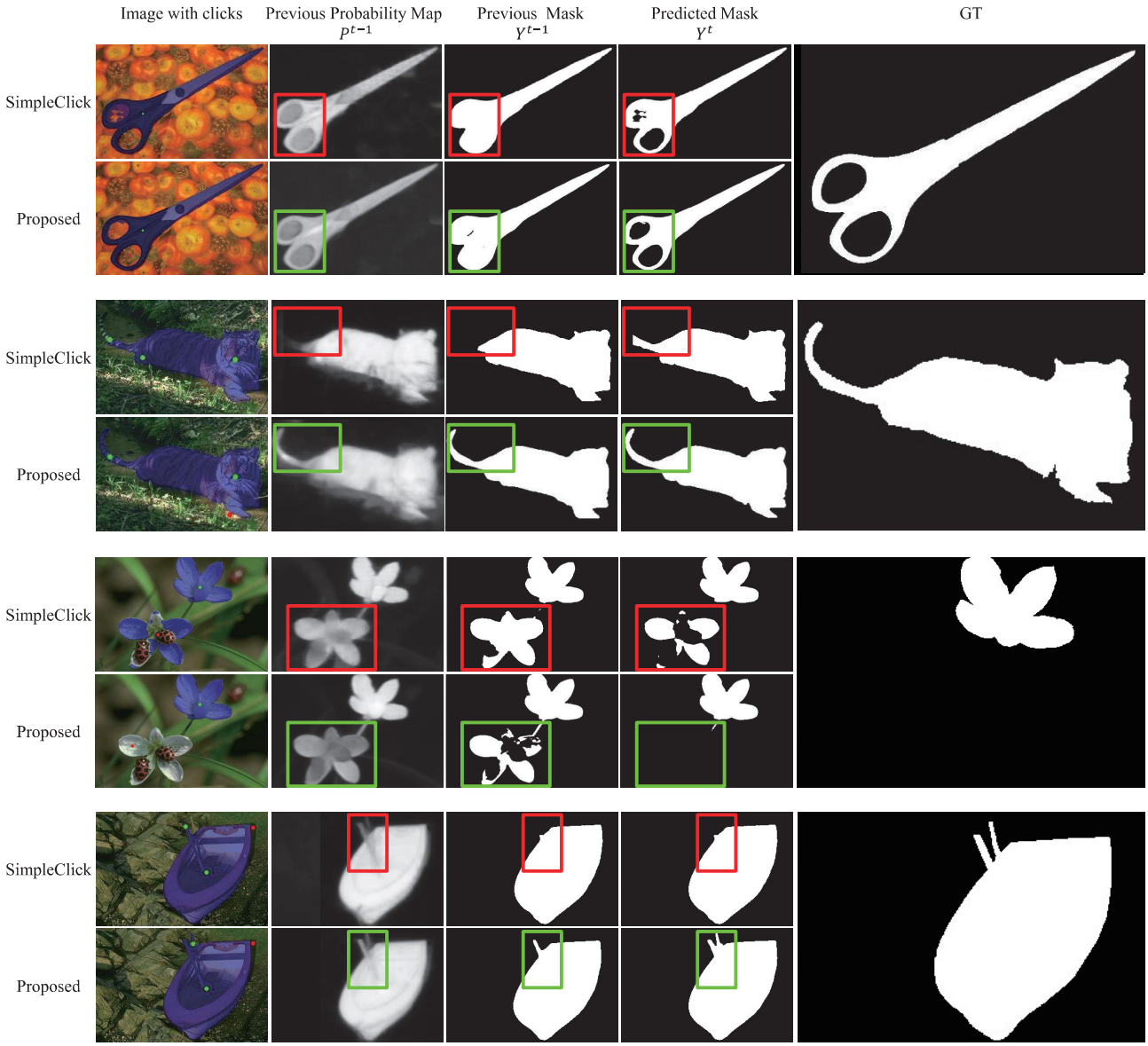


Figure S-3. Qualitative comparison of the proposed algorithm with SimpleClick [10]. Both algorithms are trained on COCO+LVIS.

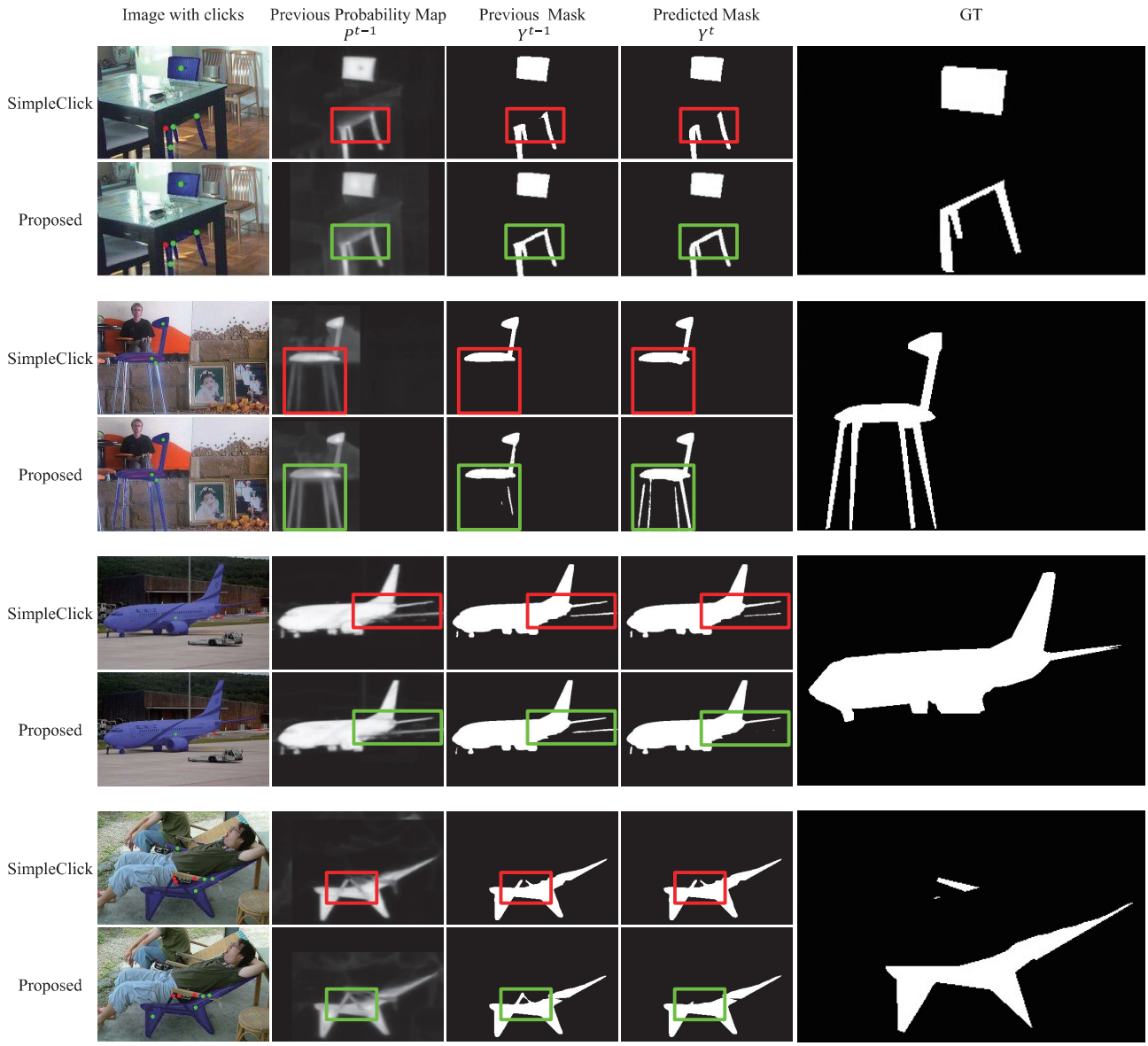


Figure S-4. Qualitative comparison of the proposed algorithm with SimpleClick [10]. Both algorithms are trained on COCO+LVIS.

C. More Experiments

C.1. Hyper-Parameters

The probability map modulation scheme has two hyper-parameters: N and $R_{\max}$. In the main paper, we set $N = 7$ and $R_{\max} = 100$ for all our experiments. Here, we demonstrate how the hyper-parameters affect segmentation results. In this test, we use ViT-B as the backbone and train the model on the SBD dataset [3].

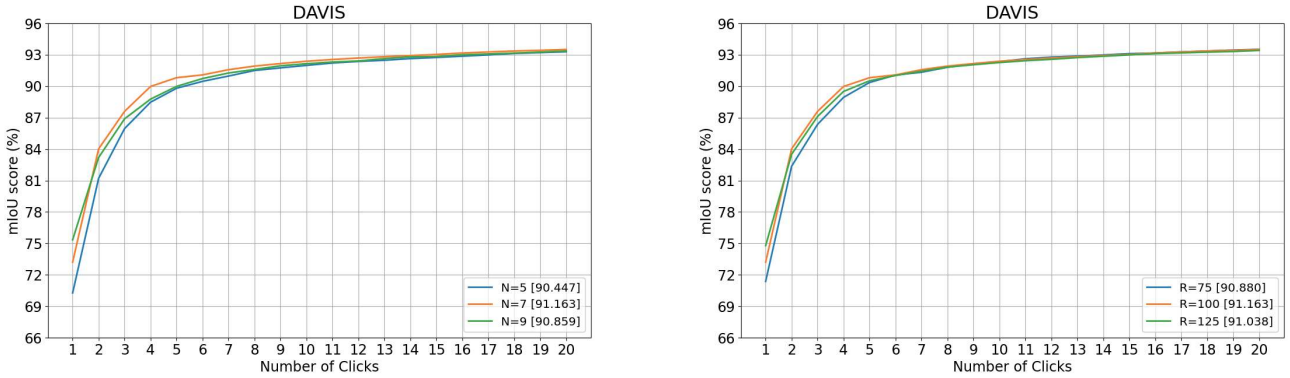
Table S-3 compares the NoC scores at different N's on the GrabCut, Berkeley, and DAVIS datasets, in which $R_{\max} = 100$. Table S-4 compares the NoC scores at different $R_{\max}$'s, in which $N = 7$. Also, Figure S-5 shows the IoU curves on the DAVIS dataset. From Table S-3, Table S-4, and Figure S-5, we see that decent results are obtained at $N = 7$ and $R_{\max} = 100$.

Table S-3. The NoC scores of the proposed MFP algorithm according to N.

N	GrabCut			Berkeley			DAVIS		
	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
5	1.42	1.50	2.26	1.51	2.31	6.58	4.38	5.69	12.01
7	1.38	1.48	1.92	1.39	2.17	6.18	3.92	5.32	11.27
9	1.38	1.46	2.04	1.48	2.33	6.12	4.03	5.44	11.66

Table S-4. The NoC scores of the proposed MFP algorithm according to $R_{\max}$.

$R_{\max}$	GrabCut			Berkeley			DAVIS		
	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
75	1.36	1.50	1.92	1.40	2.24	5.90	4.14	5.36	11.19
100	1.38	1.48	1.92	1.39	2.17	6.18	3.92	5.32	11.27
125	1.38	1.44	1.98	1.43	2.25	6.00	3.91	5.35	11.64



(a) mIoU performances according to N (b) mIoU performances according to $R_{\max}$

Figure S-5. The mean IoU performances according to the hyper-parameters N and $R_{\max}$ on the DAVIS dataset. $R_{\max} = 100$ in (a), while $N = 7$ in (b).

C.2. Modulation Schemes

RGB Distance vs. Probability Distance: In our modulation scheme, we assign different values of gamma according to how far the pixel is from a given click. For early clicks, we adopted the probability-distance-based scheme. Table S-5 also shows performances when the RGB distance is used instead for the modulation process. In the RGB-distance-based scheme, a pixel is less modulated (*i.e.* a smaller gamma value is assigned) when its RGB value differs more from the RGB value of the click. We see that using the probability-distance-based scheme achieves better NoC scores than the RGB-distance-based scheme.

Table S-5. Comparison of NoC scores using the RGB-distance-based modulation scheme with the probability-distance-based scheme.

Modulation strategy	GrabCut			Berkeley			DAVIS		
	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
RGB distance	**1.36**	**1.44**	2.08	1.46	2.37	6.22	3.95	5.46	11.95
Probability distance	1.38	1.48	**1.92**	**1.39**	**2.17**	**6.18**	**3.92**	**5.32**	**11.27**

Modulation Functions: For the probability map modulation scheme, we use the method of gamma correction. However, other functions can also be used for the modulation process. Figure S-6 illustrates two additional functions that could be used for the modulation scheme. f_1 is not a smooth function, and f_2 has a limited mapping range, thus they may cause undesirable modulation. We hence found f_3 (the gamma correction function) most suitable. Table S-6 shows how the different modulation functions affect the performances. We employ ViT-B as the backbone and use the SBD dataset for training. Note that f_3 generally shows the best performance.

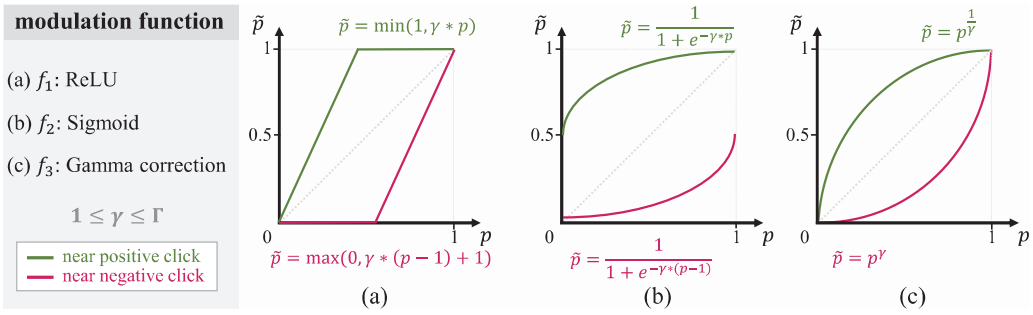


Figure S-6. Illustration of functions that could be used for the probability modulation process.

Table S-6. Comparison of NoC scores according to the modulation function used for the probability map modulation.

	GrabCut			Berkeley			DAVIS		
	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
f_1	**1.38**	1.50	2.06	**1.37**	2.25	**5.91**	3.97	**5.32**	11.36
f_2	1.42	1.56	2.00	1.42	2.28	6.07	4.00	5.38	11.90
f_3	**1.38**	**1.48**	**1.92**	1.39	**2.17**	6.18	**3.92**	**5.32**	**11.27**

References

[1] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. FocalClick: towards practical interactive image segmentation. In *CVPR*, pages 1300–1309, 2022. 3

[2] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: a dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 2

[3] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011. 2, 5

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1

[5] You Huang, Hao Yang, Ke Sun, Shengchuan Zhang, Liujuan Cao, Guannan Jiang, and Rongrong Ji. Interformer: Real-time interactive image segmentation. In *ICCV*, pages 22301–22311, 2023. 2

[6] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *CVPR*, pages 5297–5306, 2019. 3

[7] Kun Li, George Vosselman, and Michael Ying Yang. Interactive image segmentation with cross-modality vision transformers. In *ICCV*, pages 762–772, 2023. 2

[8] Jiacheng Lin, Jiajun Chen, Kailun Yang, Alina Roitberg, Siyu Li, Zhiyong Li, and Shutao Li. Adaptiveclick: Clicks-aware transformer with adaptive focal loss for interactive image segmentation. *arXiv preprint arXiv:2305.04276*, 2023. 2

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 2

[10] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. SimpleClick: interactive image segmentation with simple vision transformers. In *ICCV*, pages 22290–22300, 2023. 1, 2, 3, 4

[11] Kevin McGuinness and Noel E O'connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43 (2):434–444, 2010. 2

[12] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 2

[13] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004. 2

[14] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016. 3