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

Abstract

In recent interactive segmentation algorithms, previous probability maps are used as network input to help predictions in the current segmentation round. However, despite the utilization of previous masks, useful information contained in the probability maps is not well propagated to the current predictions. In this paper, to overcome this limitation, we propose a novel and effective algorithm for click-based interactive image segmentation, called MFP, which attempts to make full use of probability maps. We first modulate previous probability maps to enhance their representations of user-specified objects. Then, we feed the modulated probability maps as additional input to the segmentation network. We implement the proposed MFP algorithm based on the ResNet-34, HRNet-18, and ViT-B backbones and assess the performance extensively on various datasets. It is demonstrated that MFP meaningfully outperforms the existing algorithms using identical backbones. The source codes are available at https://github.com/cwlee00/MFP.

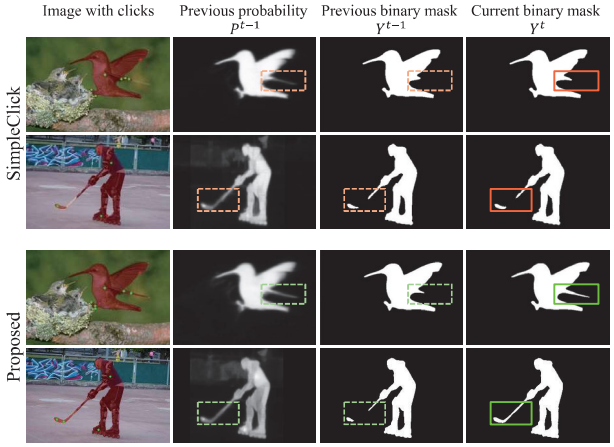


Figure 1. Utilization of previous probability information in the current segmentation round: The conventional click-based interactive segmentation algorithm SimpleClick [21] fails to capture the details contained in the previous probability maps. On the other hand, the proposed algorithm exploits the shape details in the previous probability maps to yield a better segmentation result in the current round.

1. Introduction

Interactive image segmentation is a task that aims to segment objects of interest given guidance through user annotations. It enables users to select objects and delineate them easily, so it is useful in many applications such as image editing and medical image analysis. With the rapid development of deep-learning-based algorithms for dense prediction tasks, demands for annotated data have increased significantly. However, obtaining pixel-level annotations is costly due to its laborious and time-consuming traits. With the employment of interactive segmentation techniques, these labeling costs could be reduced. It is hence essential to develop an effective interactive segmentation algorithm.

Various forms of user annotations have been adopted in interactive image segmentation, including bounding boxes [15, 25], scribbles [1, 10], and clicks [3, 7, 14, 17, 19–21, 26, 27, 29]. However, click-based interactions have become the mainstream methods due to their simplicity and well-established studies. In click-based methods, a user successively places foreground or background clicks to obtain a segmentation mask. Every time the user places a click, the segmentation mask is updated. Then, based on the segmentation mask, the user provides a new click on the mislabeled areas. This is repeatedly performed until a desired result is obtained. In this work, we attempt to develop a novel and more effective framework for click-based interactive image segmentation.

Recently, deep-learning-based techniques have shown promising results for interactive image segmentation. While early deep methods feed only an input image and click maps to the segmentation networks, Sofiiuk *et al.* [27] tried to exploit previous segmentation masks by taking them as additional input to their segmentation network. They demonstrated that making the network model aware of previous

*Corresponding author

masks improves the stability of the model. Since then, taking previous masks (or previous probability maps) as the network input has become a standard pipeline for click-based interactive segmentation. However, even though several algorithms use previous probability maps to generate current predictions, we observe that the information in the previous probability maps is not well propagated to the current predictions. Examples of these observations are presented in Figure 1.

To overcome these limitations, we propose a novel interactive segmentation framework, called **M**aking **F**ull use of **P**robability maps (MFP), to make better use of previous probabilities. Previous probabilities predicted by a segmentation network provide information such as the shape of a target object, while user clicks give accurate information for discerning foreground regions from background ones. In this paper, we first introduce the notion of probability map modulation to combine these two types of information and yield a better representation of the target object. We propose taking a modulated probability map as additional input to the network. Thus, we extend the existing interactive segmentation framework as shown in Figure 2. Experiments on four benchmark datasets demonstrate that the proposed MFP framework achieves excellent results when implemented on three different backbone networks.

The major contributions of this work can be summarized as follows.

- We propose the first modulation scheme for previous probability maps that enhances the representation of user-specified objects.
- We develop the novel MFP framework for click-based interactive segmentation, which propagates click information to unclicked locations effectively by making full use of previous probability maps.
- We implement the proposed MFP algorithm based on three backbone networks of ResNet-34 [13], HRNet-18 [28], and ViT-B [6] and assess the performance on various benchmark datasets. MFP meaningfully outperforms the existing algorithms using identical backbones.

2. Related Work

2.1. Interactive Image Segmentation

Extensive research has been carried out to solve the problem of interactive image segmentation. For instance, Rother *et al*. [25] proposed an early method, which takes interactive segmentation as a graph-based optimization problem. Such traditional methods rely on handcrafted features, thus they suffer from relatively low performance. With the emergence of deep learning, Xu *et al*. [29] first employed a convolutional neural network to perform interactive segmentation. They encoded user clicks into click maps via the distance transform, and used them with an RGB image as the network input. Since then, their idea of taking click maps as network input has become a de facto standard in deep-learning-based methods [3, 14, 26, 29]. Although click maps clearly represent the annotated labels in user-clicked locations, the output of the segmentation network is not guaranteed to have correct labels in those locations. To overcome this limitation, Jang and Kim [14] introduced the backpropagating refinement scheme (BRS), which is an inference-time optimization procedure that corrects the mislabeled clicks. Inspired by this work, Sofiiuk *et al*. [26] developed the f-BRS algorithm, which refines features instead of click maps. However, these BRS methods need to run backward gradient passes during the inference, demanding higher computational costs in general.

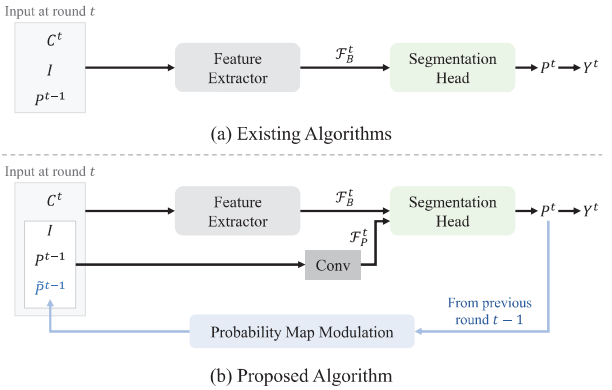
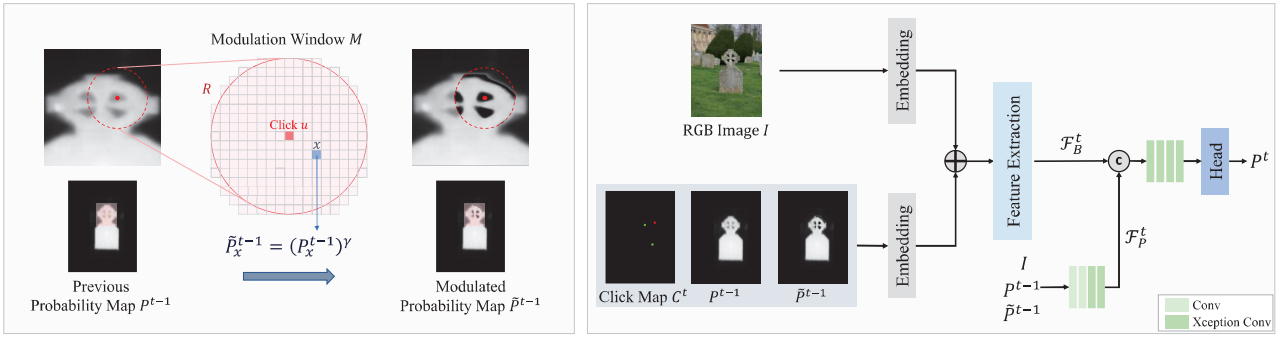


Figure 2. (a) Existing algorithms take an input image I, click map C^t, and previous probability map P^{t-1} as input to the segmentation network. From these input signals, they extract feature $\mathcal{F}_B^t$ and directly feed it into the segmentation head to obtain the current probability map P^t. Then, P^t is thresholded to the final object mask Y^t. (b) In contrast, the proposed MFP algorithm modulates P^{t-1} into $\tilde{P}^{t-1}$ and takes it as additional input to the network. Furthermore, MFP late-fuses probability-related feature $\mathcal{F}_P^t$ with backbone feature $\mathcal{F}_B^t$ before the segmentation head.

2.2. Utilization of Previous Masks

Forte *et al*. [8] and Sofiiuk *et al*. [27] started adding previous segmentation masks as network input in the current segmentation round. In particular, Sofiiuk *et al*. [27] demonstrated that even without additional optimization schemes, a simple feed-forward model using previous masks could achieve promising results. After their work, recent interactive segmentation methods in [4, 7, 19, 21, 31] all use the segmentation masks in previous segmentation rounds as network input. To further improve the segmentation performance, Chen *et al*. [4] and Lin *et al*. [19] proposed to refine segmentation results within a local window. After obtaining a global prediction, their methods also predict local results on cropped image regions around clicks and use the local predictions to refine the global prediction. Zhou *et al*.



(a) Probability map modulation

(b) Network architecture

Figure 3. Overview of the proposed MFP algorithm. In the click map, foreground and background clicks are depicted in green and red, respectively.

[31] formulated the problem of interactive segmentation as Gaussian process classification, and Du *et al*. [7] proposed to use self-attention and correlation modules for propagating click information to unclicked locations. Liu *et al*. [21] focused on developing a more effective backbone network, and leveraged a plain ViT backbone that could benefit from pretrained weights. Although all these methods use previous masks (or previous probability maps) in the current segmentation round, none of them attempt to explicitly extract beneficial information from the probability maps. Thus, they do not use the previous probability maps to their full potential.

3. Proposed Algorithm

Figure 3 is an overview of the proposed MFP algorithm. In each interactive segmentation round, we first modulate the probability map, predicted from the previous clicks, to make it closer to the actual segmentation result the user desires, as illustrated in Figure 3(a). Then, we feed the modulated map as additional input to the segmentation network in Figure 3(b). We train the segmentation network using a recursive training scheme.

3.1. Probability Map Modulation

To fully exploit the useful information in the previous prediction, we enhance the shape details of a target object in the probability map P^{t-1} from the previous round $t-1$. For regions that are likely to correspond to the target object, we enhance the probability values to make them closer to the foreground label of 1. On the contrary, for background regions, we lower the probabilities to make them closer to the background label of 0. To this end, we use the method of gamma correction [9].

Gamma correction: In round t, we use the probability map P^{t-1} that the segmentation head yielded in the last round $t-1$. We first modulate P^{t-1} into $\tilde{P}^{t-1}$ via gamma correction. We modify only the region that seems to need further refinement. More specifically, we apply gamma correction only to the pixels within a modulation window M, as shown in Figure 3(a). Typically, pixels nearer to the current click u are more likely to require further refinement. Hence, we define the modulation window as

$$M = \{x : \|x - u\| \leq R\} \tag{1}$$

where R is the radius of the window, representing the attention scope of the current click u.

Suppose that pixel x belongs to the modulation window, *i.e.* $x \in M$. Also, let P_x denote the probability of x in a probability map P. Then, P_x^{t-1} is modulated to $\tilde{P}_x^{t-1}$ by gamma correction. Let $l(u)$ be the label that the user provides at the current click u. Note that user clicks are provided on mislabeled areas. Therefore, if a foreground click is given with $l(u) = 1$, it indicates that a background label was assigned to u in the last round and $P_u^{t-1} < 0.5$. Since the user intends to assign a foreground label to u, we increase the probabilities of pixels in M. Therefore, when $l(u) = 1$, we perform the increasing modulation

$$\tilde{P}_x^{t-1} = (P_x^{t-1})^{\frac{1}{\gamma}} \tag{2}$$

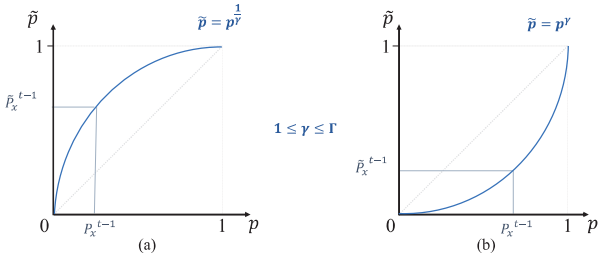


Figure 4. Illustration of the gamma correction for the probability modulation: (a) probabilities near a foreground click are increased with power $\frac{1}{\gamma}$ smaller than 1, and (b) those near a background click are decreased with power γ bigger than 1.

Table 1. Two gamma assignment schemes.

Method	Distance metric	Gamma assignment scheme
Euclidean distance	$d = \|x - u\|$	$\gamma = \Gamma \cdot (1 - \frac{d}{R}) + \frac{d}{R}$
Probability distance	$d = (P_x^{t-1} - P_u^{t-1})^2$	$\gamma = (\Gamma - 1) \cdot \max\left\{\frac{(\bar{d}-d)^3}{\bar{d}^3}, 0\right\} + 1$

for each $x \in M$, where $1 \leq \gamma \leq \Gamma$. Figure 4(a) shows a gamma correction curve for the increasing modulation. The exact opposite processing is done for a background click. When $l(u) = 0$, we perform the decreasing modulation

$$\tilde{P}_x^{t-1} = (P_x^{t-1})^{\gamma} \tag{3}$$

for each $x \in M$, as shown in Figure 4(b).

Assignment of γ: We assign different values of gamma for modulating each pixel x in M. The assignment is done according to how far x is from the given click u. For the click itself, the desired value $\tilde{P}_u^{t-1}$ is clear; it is desired that $\tilde{P}_u^{t-1} \approx 1$ if $l(u) = 1$, and $\tilde{P}_u^{t-1} \approx 0$ if $l(u) = 0$. It is hence natural to assign the biggest gamma Γ to the current click u. We set Γ so that $\tilde{P}_u^{t-1} = 0.99$ for a foreground click and $\tilde{P}_u^{t-1} = 0.01$ for a background click. On the other hand, pixels far away from u are less likely to belong to the same object as u, so we assign smaller gamma values for those pixels. For measuring how far a pixel is from the click, we propose two distance metrics: Euclidean distance and probability distance. We also develop different gamma assignment schemes for the two distance metrics. Table 1 summarizes these two schemes.

First, we measure how far pixel x is from click u by the Euclidean distance $d = \|x - u\|$. In this case, γ is determined by

$$\gamma = \Gamma \cdot (1 - \frac{d}{R}) + \frac{d}{R}. \tag{4}$$

Thus, γ linearly decreases from Γ to 0, as pixel x moves away from click u to the boundary of the modulation window M in (1).

Second, assuming that similar pixels in the same region would be assigned similar probabilities by the segmentation network, we define the probability distance $d = (P_x^{t-1} - P_u^{t-1})^2$ between x and u. In other words, pixel x is considered farther from click u when its probability P_x^{t-1} differs more from P_u^{t-1}. In this case, the maximum distance cannot be bounded by the radius R of the modulation window M. Instead, we measure the probability distances of all pixels in M from click u. Let $\bar{d}$ denote the median of these distances. Then, we determine γ by

$$\gamma = (\Gamma - 1) \cdot \frac{(\bar{d} - d)^3}{\bar{d}^3} + 1 \tag{5}$$

when $d \leq \bar{d}$. Therefore, $\gamma = \Gamma$ at click u, and $\gamma = 1$ when $d = \bar{d}$. When $d > \bar{d}$, we assume that x and u are unlikely to belong to the same region and set $\gamma = 1$.

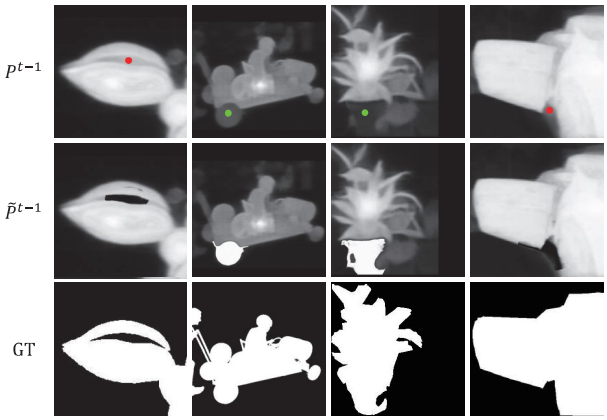


Figure 5. Examples of the probability map modulation via (5). From top to bottom: previous probability maps P^{t-1} before modulation with current clicks, modulated probability maps $\tilde{P}^{t-1}$, and ground-truth masks of target objects.

We adopt the probability-distance-based scheme in (5) for early clicks up to the Nth click and the Euclidean-distance-based scheme in (4) for later clicks. This is because, in early rounds, rough and large-scale object shapes tend to remain in P^{t-1}. On the other hand, in late rounds, the distinction between the target object and the background becomes clearer, and fine-scale information is required for better segmentation. However, P^{t-1} does not contain such fine-scale information in general. Thus, the Euclidean distance is used instead in late rounds. Figure 5 shows examples of the probability map modulation via the gamma assignment scheme in (5).

Assignment of R: We set the radius R of M to be $R_{\max}$. However, when there are previous clicks of the opposite type, we set R to be half the minimum distance of the current click u to an opposite click. Thus, we set

$$R = \min\left\{\frac{1}{2} \cdot \min_{c \in O} \|u - c\|,\ R_{\max}\right\} \tag{6}$$

where O is the set of previous clicks of the opposite type.

3.2. Network Architecture

After obtaining the modulated probability map $\tilde{P}^{t-1}$, we feed it together with the input image I, click map C^t, and original probability map P^{t-1} into the segmentation network, as depicted in Figure 3(b). For the interactive segmentation task, we adopt a common semantic segmentation network as the backbone for feature extraction. However, the semantic segmentation network takes only an RGB image as input. To handle the additional input C^t, P^{t-1}, and $\tilde{P}^{t-1}$, we concatenate them, and embed the concatenation

Table 2. The NoC scores of the proposed MFP algorithm and existing algorithms, which are trained on the SBD dataset [12].

		GrabCut			Berkeley			DAVIS			SBD		
Algorithm	Backbone	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
BRS [14]	DenseNet	2.60	3.60	-	-	5.08	-	5.58	8.24	-	6.59	9.78	-
f-BRS [26]	ResNet-101	2.30	2.72	-	-	4.57	-	5.04	7.41	-	4.81	7.73	-
RITM [27]	HRNet-18	1.76	2.04	3.66	1.87	3.22	8.35	4.94	6.71	13.87	3.39	5.43	11.65
CDNet [3]	ResNet-34	1.86	2.18	3.68	1.95	3.27	8.29	5.00	6.89	14.24	5.18	7.89	14.27
PsuedoClick [20]	HRNet-18	1.68	2.04	-	1.85	3.23	-	4.81	6.57	-	3.38	5.40	-
FocalClick [4]	SegF-B0	1.66	1.90	-	-	3.14	-	5.02	7.06	-	4.34	6.51	-
FocusCut [19]	ResNet-101	1.46	1.64	-	1.81	3.01	-	4.85	6.22	-	3.40	5.31	-
EMC-Click [7]	HRNet-18	1.74	1.84	-	-	3.03	-	5.05	6.71	-	3.38	5.51	-
GPCIS [31]	ResNet-50	1.64	1.82	2.62	1.60	2.60	6.77	4.37	5.89	12.42	3.80	5.71	**11.06**
iCMFormer [16]	ViT-B	**1.36**	**1.42**	-	1.42	2.52	-	4.05	5.58	-	3.33	5.31	-
SimpleClick [21]	ViT-B	1.40	1.54	2.16	1.44	2.46	6.70	4.10	5.48	12.24	3.28	**5.24**	11.24
MFP (Proposed)	ViT-B	1.38	1.48	**1.92**	**1.39**	**2.17**	**6.18**	**3.92**	**5.32**	**11.27**	**3.21**	**5.24**	11.20

and the input image I, respectively, into tensors of the same size, as done in [21, 27]. Then, we element-wise add these tensors and convey the sum to the feature extraction network. As the feature extraction backbone, we test ResNet-34 [13], HRNet-18 [28], and ViT-B [6] of different complexities, which are widely used for interactive image segmentation. The feature extractor yields a feature map $\mathcal{F}_B^t$.

Existing interactive segmentation algorithms directly input the backbone feature $\mathcal{F}_B^t$ into a segmentation head to obtain a segmentation result. In contrast, we propose fusing the probability maps P^{t-1} and $\tilde{P}^{t-1}$ with $\mathcal{F}_B^t$. This late fusion strengthens the influence of the probability information on the final segmentation result. Specifically, we first concatenate I, P^{t-1}, $\tilde{P}^{t-1}$ and adjust their spatial resolutions and channel sizes using two convolution blocks. We then use two Xception conv blocks [5] to extract the probability-related feature $\mathcal{F}_P^t$. Then, we concatenate $\mathcal{F}_P^t$ with $\mathcal{F}_B^t$ and fuse them through four Xception conv blocks. The segmentation head processes this fused feature to generate the probability map P^t in the current round t. Thresholding this map yields the final segmentation mask Y^t.

3.3. Recursive Training

As the proposed MFP algorithm uses P^{t-1} to predict P^t, its training requires an ordered series of clicks and the corresponding probability maps. Sofiiuk *et al*. [27], one of the first methods to use previous probability maps in interactive segmentation, adopt an iterative sampling strategy. They first sample user clicks randomly, as done in [29]. After the random clicks, they add $0 \sim 3$ clicks iteratively based on the errors in the network prediction results. Although they partly train the model by mimicking user interactions, they still resort to random sampling initially. Thus, for initial clicks, there are no previous probability maps, and the probability modulation cannot be performed. Therefore, in this work, we develop a fully recursive training strategy to exploit the information in the ordered series of interactive clicks. More specifically, we start the training of an image by sampling the first click near the center of a target object. Then, by comparing the result of the segmentation network with the ground truth, we select the next click near the center of the biggest error region. This recursive selection and training is performed up to 24 clicks for an image.

4. Experiments

4.1. Experimental Settings

Datasets: We use four widely used datasets GrabCut [25], Berkeley [22], DAVIS [23], and SBD [12] to assess the proposed MFP algorithm. SBD and a combination of datasets COCO [18] and LVIS[11] are used for training.

- GrabCut: It consists of 50 images. For each image, a single object mask is provided.
- Berkeley: It contains 96 images with 100 object masks.
- DAVIS: Although constructed for video object segmentation, it can be also used for interactive image segmentation by sampling frames from the videos. We use the same 345 frames, sampled from 50 videos, as in [14].
- SBD: It is divided into training and validation sets. The training set has 8,498 images with 20,172 instance masks, and the validation set has 2,857 images with 6,671 masks.
- COCO + LVIS: COCO consists of 99K images with 1.2M instance masks, and LVIS has 100K images with 1.2M masks. As in [27], we combine these two datasets and use 104K images with 1.6M instance masks for training.

Evaluation Metrics: To evaluate the proposed algorithm, we use two performance measures. First, we report the NoC score, which is the average number of clicks required to achieve a certain intersection-over-union (IoU) ratio. Previous studies usually set the target IoU ratio as 90% and report NoC@85 and NoC@90 for assessment. However, recent methods yield high-quality segmentation results. Therefore, we set the target IoU score as 95% and report NoC@95 additionally. Second, we plot the mean intersection-over-

Figure 6. Qualitative comparison of algorithms trained on the COCO+LVIS dataset. We compare the proposed MFP with conventional algorithms that employ previous masks as input: FocalClick [4], EMC-Click [7], and SimpleClick [21]. Rows (a) show input images with clicks and prediction masks overlayed. Rows (b) and (c) show previous probability maps and current prediction masks, respectively. Note that the algorithms are compared fairly using the same automatic clicking strategy. Since the algorithms produce different segmentation results, click locations may differ accordingly.

union (mIoU) score according to the number of clicks and report the area under the curve (AUC).

Implementation Details: The proposed MFP can be implemented upon various backbone networks. In this work, we implement three versions based on the ViT-B [6], HRNet-18 + OCR [28, 30], and ResNet-34 + DeepLabv3+ [2, 13] backbones. Also, we use the SBD [12] and COCO + LVIS [11, 18] datasets for training. We apply random resizing, cropping, flipping, rotation, and brightness control for data augmentation. We minimize the normalized focal loss [27] using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We fix the hyperparameters in the probability map modulation to $N = 7$ and $R_{\max} = 100$ in all experiments. For evaluation, results can be greatly affected by how a user places clicks. Thus, for reliable assessment, we adopt the automatic clicking strategy used in [4, 14, 29].

4.2. Comparative Assessment

Comparison with state-of-the-art methods: We first compare the proposed MFP (ViT-B) with conventional interactive segmentation algorithms using the NoC metrics. Some methods employ large transformers including ViT-B, ViT-L and ViT-H. The best performances by the recent methods are obtained using the ViT-H backbone. However, ViT-H suffers from high computational costs due to its large number of parameters. For this reason, we compare the proposed algorithm with the state-of-the-art methods by employing ViT-B. Early algorithms were trained on the SBD dataset only, while recent methods are trained on SBD and COCO + LVIS, respectively. We report the results of MFP under both training settings following the recent methods.

Table 2 compares the results using the SBD training data. For the methods [4, 7, 19, 20, 27] that presented multiple versions of their algorithms, we report their best NoC scores. Looking at the table results, we see that MFP outperforms the existing algorithms significantly. Note that MFP is the only algorithm that achieves a mean IoU of 85% with less than four clicks for all the datasets. Also, MFP yields the best results in 9 out of 12 tests, and the second-best results in the remaining three tests.

Figure 6 compares qualitative segmentation results of the proposed MFP with those of FocalClick [4], EMC-Click [7], and SimpleClick [21]. We can see that at the same number of clicks, MFP provides more accurate segmentation results. We also visualize previous probability maps $\tilde{P}^{t-1}$ in rows (b) of Figure 6. Unlike the existing algorithms, whose segmentation masks are inaccurate in regions far from the clicks (*e.g.* the bicycle wheels), MFP manages to make cor-

Table 3. The NoC scores of the proposed MFP algorithm and existing algorithms, which are trained on the COCO + LVIS datasets [11, 18].

		GrabCut			Berkeley			DAVIS			SBD		
Algorithm	Backbone	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
RITM [27]	HRNet-32	1.46	1.56	2.48	1.43	2.10	5.41	4.11	5.34	11.51	3.59	5.71	12.00
CDNet [3]	ResNet-34	1.40	1.52	1.84	1.47	2.06	5.42	4.27	5.56	11.90	4.30	7.04	14.17
PseudoClick [20]	HRNet-32	1.36	1.50	-	1.40	2.08	-	3.79	5.11	-	3.46	5.54	-
FocalClick [4]	SegF-B3	1.44	1.50	1.82	1.55	1.92	**4.63**	3.61	4.90	10.58	3.43	5.62	**11.55**
EMC-Click [7]	HRNet-32	**1.30**	**1.42**	1.84	1.48	2.35	6.95	4.29	5.33	11.82	3.55	5.65	12.26
DynaMITe [24]	Swin-L	1.62	1.72	-	1.39	1.90	-	3.80	5.09	-	3.32	5.64	-
iCMFormer [16]	ViT-B	1.42	1.52	-	1.40	**1.86**	-	3.40	5.06	-	3.29	**5.30**	-
SimpleClick [21]	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

Table 4. Comparison of the proposed MFP algorithm with conventional algorithms using identical backbones. The algorithms are grouped according to the backbone networks employed. All algorithms are trained on the SBD dataset [12].

		GrabCut			Berkeley			DAVIS			SBD		
Algorithm	Backbone	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95	NoC85	NoC90	NoC95
CDNet [3]	ResNet-34	1.86	2.18	3.68	1.95	**3.27**	8.29	5.00	**6.89**	**14.24**	5.18	7.89	14.27
MFP (Proposed)	ResNet-34	**1.70**	**1.92**	**3.00**	**1.71**	3.37	**7.94**	**4.97**	7.78	14.93	**3.92**	**6.21**	**12.47**
RITM [27]	HRNet-18	1.76	2.04	3.66	1.87	3.22	8.35	4.94	6.71	13.87	3.39	5.43	11.65
PsuedoClick [20]	HRNet-18	1.68	2.04	-	1.85	3.23	-	4.81	6.57	-	**3.38**	**5.40**	-
EMC-Click [7]	HRNet-18	1.74	1.84	-	-	**3.03**	-	5.05	6.71	-	**3.38**	5.51	-
MFP (Proposed)	HRNet-18	**1.52**	**1.60**	**2.90**	**1.68**	3.04	**7.94**	**4.77**	**6.36**	**13.45**	3.43	5.45	**11.58**
iCMFormer [16]	ViT-B	**1.36**	**1.42**	-	1.42	2.52	-	4.05	5.58	-	3.33	5.31	-
SimpleClick [21]	ViT-B	1.40	1.54	2.16	1.44	2.46	6.70	4.10	5.48	12.24	3.28	**5.24**	11.24
MFP (Proposed)	ViT-B	1.38	1.48	**1.92**	**1.39**	**2.17**	**6.18**	**3.92**	**5.32**	**11.27**	**3.21**	**5.24**	**11.20**

rect predictions in those regions. Specifically, the previous probability maps of both SimpleClick and MFP contain information about the wheel shapes, but only MFP can segment out the wheels. From this observation, we believe that the proposed MFP algorithm makes better use of the information in previous masks.

Table 3 compares the results using the COCO + LVIS training data. Out of 12 NoC scores compared, MFP outperforms the existing algorithms in seven tests and ranks second in the remaining five tests. This demonstrates that the proposed MFP algorithm exceeds or shows comparable results to the state-of-the-art methods.

Comparison of IoU & AUC: Figure 7 compares the proposed MFP in terms of mean IoU ratios with four comparable algorithms trained on the SBD dataset: CDNet [3], RITM [27], GPCIS [31], SimpleClick [21]. We see that MFP generally achieves higher mean IoU ratios with the same number of clicks than the conventional algorithms. To numerically prove the superiority of MFP, we also report the AUC scores. MFP shows the highest AUC scores on all four datasets.

Comparison using identical backbones: Using stronger backbones greatly impacts the performance of interactive segmentation algorithms. Thus, to make fair comparisons with existing methods, we test the proposed MFP algorithm using three different backbones: ResNet-34 [13], HRNet-18 [28], and ViT-B [6]. We choose ViT-B as the main backbone following the state-of-the-art methods in [16, 21]. As ViT-B is a heavy network, we also choose relatively lighter backbones ResNet-34 and HRNet-18, which are employed by many other interactive segmentation methods. In Table 4, we compare the three versions of MFP with conventional algorithms that use the same backbone networks. Out of total 36 settings compared, MFP shows superiour results in 28 settings. Even for the cases where MFP does not achieve the highest scores, MFP yields comparable performance to the best results with margins less than 0.1. This demonstrates the effectiveness and generality of the proposed MFP framework.

4.3. Ablation Study

We conduct ablation studies to verify the efficacy of each component of the proposed algorithm, using the SBD training data. For evaluation, we choose the DAVIS dataset [23]. DAVIS has high-quality annotations and covers many different scenarios, thus we found the DAVIS results more convincing as compared with other datasets. The results are presented in Table 5. We first implement a baseline model (Method I) based on ViT-B. Method I does not take modulated probability maps as additional input and follows the

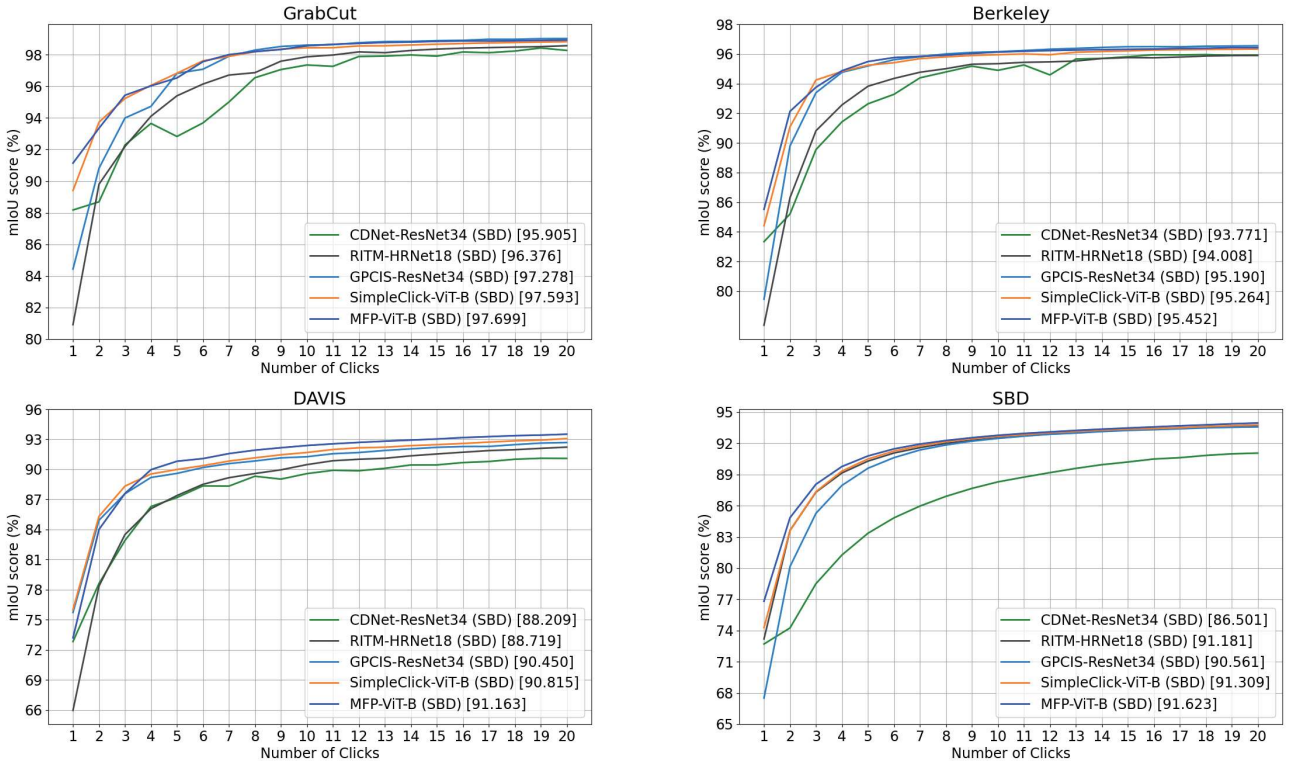


Figure 7. Comparison of the mean IoU scores according to the number of clicks on the GrabCut[25], Berkeley[22], DAVIS[23], and SBD[12] datasets. The models are trained on the SBD datasets. The legend of the graph contains the AuC score for each algorithm.

Table 5. Ablation studies of the proposed MFP algorithm conducted on the DAVIS dataset [23].

	$\tilde{P}^{t-1}$	Late fusion	Recursive training	NoC@85	NoC@90	NoC@95
I.				4.10	5.60	11.58
II.	✓		✓	4.05	5.30	11.54
III.	✓	✓		3.98	5.35	11.79
IV.	✓	✓	✓	3.92	5.32	11.27

same network architecture and training scheme as the previous algorithm [21]. In method II, we explore the effect of the late fusion strategy, by excluding the fusion layer. For method III, we follow the training procedure in [27], instead of the recursive training. When random sampling simulates clicks, since there is no order in the clicks, we randomly take any click and consider it as the current click to perform the probability map modulation. Method IV employs all components proposed in this work, so it is the same as the proposed algorithm.

Looking into the results in Table 5, we can see that method IV, which employs all components, yields the best performance. This indicates that all components contribute to performance improvements. Comparing the results between pairs of methods, we find that the performance gain from method II to IV is conspicuous for NoC@85, while that from method III to IV stands out for NoC@95. From this observation, we deduce that fusing probability information in a late layer has great impact for quickly obtaining coarse predictions, while the recursive training scheme aids in fine-level tuning for higher accuracy.

5. Conclusions

A novel click-based interactive segmentation framework, called MFP, which fully exploits previous probability maps was proposed in this paper. First, MFP modulates previous probability maps based on click information to obtain better representations of user-specified objects. Then, it propagates the additional information to the segmentation network, which was designed by extending the existing interactive segmentation framework. Experimental results demonstrated that MFP outperforms conventional algorithms when identical backbones are employed.

Acknowledgements

This work was supported by the NRF grants funded by the Korea government (MSIT) (No. NRF-2021R1A4A1031864 and No. NRF-2022R1A2B5B03002310), and by the IITP grant funded by the Korea government (MSIT) (No. 2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In *CVPR*, pages 392–399, 2014. 1

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 6

[3] Xi Chen, Zhiyan Zhao, Feiwu Yu, Yilei Zhang, and Manni Duan. Conditional diffusion for interactive segmentation. In *ICCV*, pages 7345–7354, 2021. 1, 2, 5, 7

[4] 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. 2, 5, 6, 7

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017. 5

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 5, 6, 7

[7] Fei Du, Jianlong Yuan, Zhibin Wang, and Fan Wang. Efficient mask correction for click-based interactive image segmentation. In *CVPR*, pages 22773–22782, 2023. 1, 2, 3, 5, 6, 7

[8] Marco Forte, Brian Price, Scott Cohen, Ning Xu, and François Pitié. Getting to 99% accuracy in interactive segmentation. *arXiv preprint arXiv:2003.07932*, 2020. 2

[9] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2008. 3

[10] Leo Grady. Random walks for image segmentation. *IEEE TPAMI*, 28(11):1768–1783, 2006. 1

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: a dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 5, 6, 7

[12] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011. 5, 6, 7, 8

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 5, 6, 7

[14] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *CVPR*, pages 5297–5306, 2019. 1, 2, 5, 6

[15] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *ICCV*, pages 277–284, 2009. 1

[16] Kun Li, George Vosselman, and Michael Ying Yang. Interactive image segmentation with cross-modality vision transformers. In *ICCV*, pages 762–772, 2023. 5, 7

[17] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, pages 577–585, 2018. 1

[18] 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. 5, 6, 7

[19] Zheng Lin, Zheng-Peng Duan, Zhao Zhang, Chun-Le Guo, and Ming-Ming Cheng. FocusCut: diving into a focus view in interactive segmentation. In *CVPR*, pages 2637–2646, 2022. 1, 2, 5, 6

[20] Qin Liu, Meng Zheng, Benjamin Planche, Srikrishna Karanam, Terrence Chen, Marc Niethammer, and Ziyan Wu. PseudoClick: interactive image segmentation with click imitation. In *ECCV*, pages 728–745, 2022. 5, 6, 7

[21] 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, 5, 6, 7, 8

[22] Kevin McGuinness and Noel E O'connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010. 5, 8

[23] 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. 5, 7, 8

[24] Amit Kumar Rana, Sabarinath Mahadevan, Alexander Hermans, and Bastian Leibe. Dynamite: Dynamic query bootstrapping for multi-object interactive segmentation transformer. *arXiv preprint arXiv:2304.06668*, 2023. 7

[25] 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. 1, 2, 5, 8

[26] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-BRS: Rethinking backpropagating refinement for interactive segmentation. In *CVPR*, pages 8623–8632, 2020. 1, 2, 5

[27] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *ICIP*, pages 3141–3145, 2022. 1, 2, 5, 6, 7, 8

[28] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 2, 5, 6, 7

[29] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016. 1, 2, 5, 6

[30] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, pages 173–190, 2020. 6

[31] Minghao Zhou, Hong Wang, Qian Zhao, Yuexiang Li, Yawen Huang, Deyu Meng, and Yefeng Zheng. Interactive segmentation as gaussion process classification. In *CVPR*, pages 19488–19497, 2023. 2, 3, 5, 7