AdaptiveClick: Click-Aware Transformer With Adaptive Focal Loss for Interactive Image Segmentation

Jiacheng Lin, Jiajun Chen, Kailun Yang, Alina Roitberg, *Member, IEEE*, Siyu Li, Zhiyong Li, *Member, IEEE*, and Shutao Li, *Fellow, IEEE*

***Abstract*— Interactive image segmentation (IIS) has emerged as a promising technique for decreasing annotation time. Substantial progress has been made in pre- and post-processing for IIS, but the critical issue of interaction ambiguity, notably hindering segmentation quality, has been under-researched. To address this, we introduce ADAPTIVECLICK — a click-aware transformer incorporating an adaptive focal loss (AFL) that tackles annotation inconsistencies with tools for mask- and pixel-level ambiguity resolution. To the best of our knowledge, AdaptiveClick is the first transformer-based, mask-adaptive segmentation framework for IIS. The key ingredient of our method is the click-aware mask-adaptive transformer decoder (CAMD), which enhances the interaction between click and image features. Additionally, AdaptiveClick enables pixel-adaptive differentiation of hard and easy samples in the decision space, independent of their varying distributions. This is primarily achieved by optimizing a generalized AFL with a theoretical guarantee, where two adaptive coefficients control the ratio of gradient values for hard and easy pixels. Our analysis reveals that the commonly used Focal and BCE losses can be considered special cases of the proposed AFL. With a plain ViT backbone, extensive experimental results on nine datasets demonstrate the superiority of AdaptiveClick compared to state-of-the-art methods. The source code is publicly available at https://github.com/lab206/AdaptiveClick.**

***Index Terms*— Adaptive focal loss, click-aware attention, interaction ambiguity, interactive segmentation, vision transformer.**

Manuscript received 7 May 2023; revised 29 December 2023; accepted 12 March 2024. Date of publication 28 March 2024; date of current version 1 March 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4701404, in part by the National Natural Science Foundation of China under Grant U21A20518 and Grant U23A20341, and in part by Hangzhou SurImage Technology Company Ltd. *(Corresponding authors: Kailun Yang; Zhiyong Li.)*

Jiacheng Lin and Zhiyong Li are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: jcheng_lin@hnu.edu.cn; zhiyong.li@hnu.edu.cn).

Jiajun Chen, Kailun Yang, and Siyu Li are with the School of Robotics and the National Engineering Research Center of Robot Visual Perception and Control Technology, Hunan University, Changsha 410082, China (e-mail: chenjiajun@hnu.edu.cn; kailun.yang@hnu.edu.cn; lsynn@hnu.edu.cn).

Alina Roitberg was with the Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany. She is now with the Institute for Artificial Intelligence, University of Stuttgart, 70569 Stuttgart, Germany (e-mail: alina.roitberg@ki.uni-stuttgart.de).

Shutao Li is with the College of Electrical and Information Engineering and the Key Laboratory of Visual Perception and Artificial Intelligence of Hunan Province, Hunan University, Changsha 410082, China (e-mail: shutao_li@hnu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2024.3378295, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3378295

I. INTRODUCTION

INTERACTIVE image segmentation (IIS) tasks are designed to efficiently segment a specified object in an image by utilizing a minimal number of user operations, such as drawing boxes [1], [2], scribbling [3], [4], and clicking [5], [6], [7], [8], [9]. Thanks to the unique interactivity and time-efficiency of this labeling paradigm, existing IIS models are widely or potentially used in fields such as medical image processing [10], [11], dataset production [8], [12], and industry perception [13], [14], [15].

A significant portion of interaction segmentation research targets *click-based* IIS. Existing studies explored click-based IIS tasks mainly from the perspective of data embedding [6], [16], [17], interaction ambiguity [18], [19], [20], segmentation network [21], [22], post-processing [8], [21], back-propagation [23], [24], and loss optimization [6], [25], which have effectively improved training convergence and stability, yielding impressive results. Although interaction ambiguity has been investigated in earlier studies [7], [19], [20], multiple deeper underlying issues, inter-class click ambiguity and intra-class click ambiguity, have hindered effective solutions.

On the one hand, interclass click ambiguity arises when a click may correspond to multiple potential objects in an image. For instance, in Fig. 1, the potential object could be a *human*, a *horse*, or a *combination of both* with a click. However, most conventional IIS methods are mask-fixed methods, which can only produce a single mask, rendering them ineffective in addressing the click ambiguity. Apart from the ambiguity introduced by user clicks, another significant factor exacerbating inter-class click ambiguity is the long-range propagation fading of click features. On the other hand, intraclass click ambiguity is induced by "gradient swamping" during the optimization process of focal loss (FL). This results in a substantial number of misclassified pixels around the decision boundary, as illustrated by the $\mathbf{P}_t$ plot of FL in Fig. 1, further exacerbating the interaction ambiguity. The "gradient swamping" refers to the phenomenon of FL focusing too much on the classification of hard pixels, significantly weakening the gradient values that many low-confidence easy pixels (also referred to as ambiguous pixels) should contribute.

In this article, we rethink the interaction ambiguity of IIS by addressing both inter-class and intra-class click ambiguities.

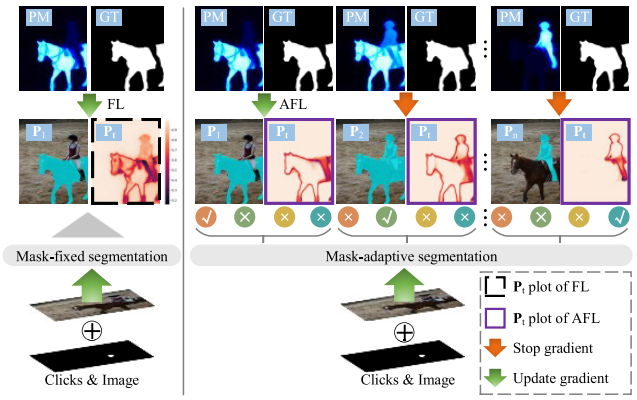


Fig. 1. Illustration of the proposed method compared with the existing mask-fixed IIS methods. In mask-fixed IIS methods, only a single mask is generated given the input. In contrast, our mask-adaptive AdaptiveClick can produce multiple candidate masks ($\mathbf{P}_1 \sim \mathbf{P}_n$) to address possible ambiguities introduced by user clicks. The model then selects the optimal combination between the GT and probability map (PM). Finally, AFL adaptively adjusts the optimal combination to produce a higher-quality mask. Here, $\mathbf{P}_t$ denotes the confidence of the pixel in the sample, with darker colors indicating more hard to segment and vice versa.

First, we propose a mask-adaptive AdaptiveClick, which is composed of a click-aware mask-adaptive transformer decoder (CAMD) with an adaptive focal loss (AFL). CAMD incorporates a click-aware attention module (CAAM) that generates distinct instance masks for each click by considering potential click ambiguities and subsequently selecting the optimal mask. This addresses inter-class click ambiguity and accelerates the convergence of the transformer. Next, we observe that the root of intraclass click ambiguity is "gradient swamping," which aggravates interaction ambiguity in IIS tasks. To tackle this, we put forward a novel AFL based on the gradient theory of BCE and FL. AFL adapts the training strategy according to the difficulty distribution of samples, improving intra-class click ambiguity problems (the $\mathbf{P}_t$ plot of AFL in Fig. 1 with clearer boundaries).

At a glance, the main contributions delivered in this work are summarized as follows.

1) A novel mask-adaptive segmentation framework is designed for IIS tasks. To the best of our knowledge, this is the first mask-adaptive segmentation framework based on transformers in the context of IIS.

2) We put forward a new clicks-aware mask-adaptive transformer decoder with a clicks-aware attention module to tackle the problem of inter-class click ambiguity arising from user ambiguity clicks and the long-range propagation fading of clicks features in the IIS methods.

3) A novel AFL is designed to overcome the "gradient swamping" problem specific to focal loss-based training in IIS, which accelerates model convergence and alleviates intraclass click ambiguity.

4) Experimental results from nine datasets showcase the clear benefits of AdaptiveClick and AFL, yielding state-of-the-art performance on IIS tasks.

The remainder of the article is structured as follows. Section II provides a brief overview of related work. The proposed methods are presented in Section III. We present experimental results in Section IV, followed by the discussion in Section V. Section VI summarizes the findings presented in this work.

II. Related Work

A. Architecture of IIS

In the IIS task, the goal is to obtain refined masks by effectively leveraging robust fusion features activated by click information. Interaction strategies in mainstream research works can be generally divided into three categories: pre-fusion [6], [16], [26], [27], secondary fusion [12], [16], [18], [28] and middle-fusion [29], [30], [31], [32].

Pre-fusion incorporates click as an auxiliary input, which includes click features [26] and previous mask [6], [8], [27]. Starting with click embedding via distance transformation [26], a common technique is to concatenate maps of clicks with raw image [19], [23], [33]. Subsequently, Benenson et al. [34] suggest that using clicks with an appropriate radius offers better performance. Nevertheless, due to the potential absence of pure interaction strategies in [26], another line of work [6], [35] constructively employs masks predicted from previous clicks as input for subsequent processing. Following this simple yet effective strategy, several studies have achieved remarkable results, such as incorporating cropped focus views [36] or exploiting IIS transformer [27]. Although the above methods strive to maximize the use of click, the click feature tends to fade during long-range propagation, exacerbating the interaction ambiguity of IIS methods.

To address the above issues, secondary fusion has been introduced as a potential solution. Lin et al. [16] emphasizing the impact of the first click, long-range propagation strategy for click [18], and utilizing multiscale strategies [20]. Inspired by [6], Chen et al. [21] further fuse click and previous mask for the refinement of the local mask. Recently, Wei et al. [28] proposed a deep feedback-integrated strategy that fuses coordinate features with high-level features. Moreover, recent works [23], [24], [37] have combined click-embedding strategies [26] with an online optimization approach. However, no existing research has explored overcoming the challenge of interaction ambiguity in vision transformers.

Recently, middle-fusion strategies have gained popularity in the IIS field. Kirillov et al. [29] propose a novel strategy that treats click as a prompt, supporting multiple inputs. Similarly, Zou et al. [32] concurrently suggest joint prompts for increased versatility. For better efficiency, Huang et al. [30] bypass the strategies in [6] related to the backbone to perform inference multiple times. Inspired by Mask2Former [38], a parallel study [31] models click as queries with the timestamp to support multiinstance IIS tasks. Although these models achieve remarkable success, they sacrifice the performance of a single IIS task to enhance multitask generalization.

In this article, we first explore mask-adaptive transformers for IIS, focusing on solving interaction ambiguity. Specifically, AdaptiveClick generates multiple candidate masks from ambiguity clicks and selects the optimal one for the final inference. Further, AdaptiveClick achieves long-range click propagation

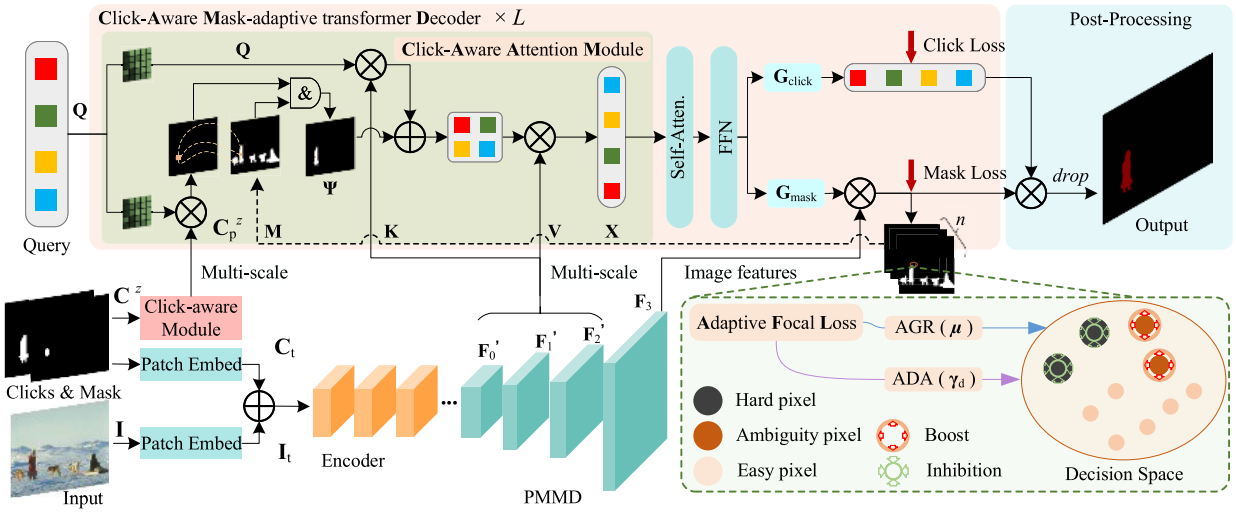


Fig. 2. Illustration of AdaptiveClick. First, clicks with the previous mask and image (**I**) features are obtained via patch embedding, then fused by addition. Second, the fusion features are obtained by Encoder, and the pixel features $\mathbf{F}_0$, $\mathbf{F}_1$, and $\mathbf{F}_2$ of different dimensions are obtained by PMMD. Third, $\mathbf{F}_0'$, $\mathbf{F}_1'$, $\mathbf{F}_2'$, and $\mathbf{F}_3$ and clicks are jointly input into the designed CAMD. Fourth, CAMD generates n corresponding masks $\hat{p}$ based on each click and then completes the optimization training process with ADA and AGR proposed in AFL. Finally, the obtained mask sequences are passed through post-processing to output the final mask.

within transformers through a CAAM, breaking the limitations of mask-fixed IIS methods.

B. Loss Function of IIS

The BCE loss [39] and its variants [36], [40], [41], [42], [43] are widely used in IIS tasks [20], [23], [25], [44] for training. However, BCE treats all pixels equally, resulting in a gradient of easy pixels inundated by hard pixels and blocking the model's performance. Previous efforts [36], [41], [45] attempt to solve this issue from the imbalance of positive/negative samples [46], [47] or easy/hard samples [36], [40].

To counteract the imbalance between positive and negative samples, WBCE [41] introduces a weighting coefficient for positive samples. Furthermore, Balanced CE [42] weights not only positive samples but also negative samples. These methods are used with data that satisfies a skewed distribution [48] but are blocked in balanced datasets due to their adjustable parameters, which influence the model's performance. To address the imbalance between hard and easy samples, the FL [36] formulates a difficulty modifier to enhance model training and down-weight the impact of easy examples, thereby allowing the model to focus more on the hard samples. Recently, Leng et al. [40] offer a new perspective and propose the poly loss (PL) as a linear combination of polynomial functions. Sofiiuk et al. [49] present the normalized focal loss (NFL), which expands an extra correction factor negatively correlated with the total modulate factor in FL. In addition, some other works [46], [47] also attempt to solve this problem. However, deeper reasons, such as the case that many low-confidence pixels are caused by gradient swamping, make ambiguous pixels impossible to be effectively classified.

In brief, it is crucial for the IIS tasks to address the issue of "gradient swamping" for FL-based losses to alleviate the interaction ambiguity. Unlike previous works [23], [24], we consider the interaction ambiguity from the perspective of intra-class click ambiguity optimization. Therefore, our model can focus more on ambiguous pixels for the IIS task than simply computing gradient values point-by-point equally during the gradient backpropagation process.

III. Method

At a high level, AdaptiveClick's primary components consist of four key stages: 1) data embedding; 2) feature encoding; 3) feature decoding; and 4) loss optimization. These stages are illustrated and summarized in Fig. 2. In our approach, we address the interaction ambiguity by focusing on aspects of feature decoding and loss optimization.

A. Deficiency of Existing IIS Methods

In this article, we categorize interaction ambiguity into inter-class and intra-class click ambiguity and tackle them by focusing on two distinct aspects: interclass click ambiguity resolution and intraclass click ambiguity optimization.

1) Inter-Class Click Ambiguity Resolution: Some IIS methods with fixed masks have explored interaction ambiguity by incorporating multiscale transformations [7], [19], [20] and investigating long-range propagation of clicks [16], [18], [50]. However, these studies primarily address interaction ambiguity by emphasizing click enhancement. While these methods effectively enhance the guiding role of clicks in the segmentation process, they still lack effective tools for handling click ambiguity. Thus, their primary focus is on maximizing the model's ability to generate a single ground-truth-compliant mask by emphasizing the click, but they fall short of effectively addressing click ambiguity.

2) Intraclass Click Ambiguity Optimization: Given the final prediction ($\mathbf{P}_n$) and ground truth (GT) ($\mathbf{y}_t$), BCE [39] widely

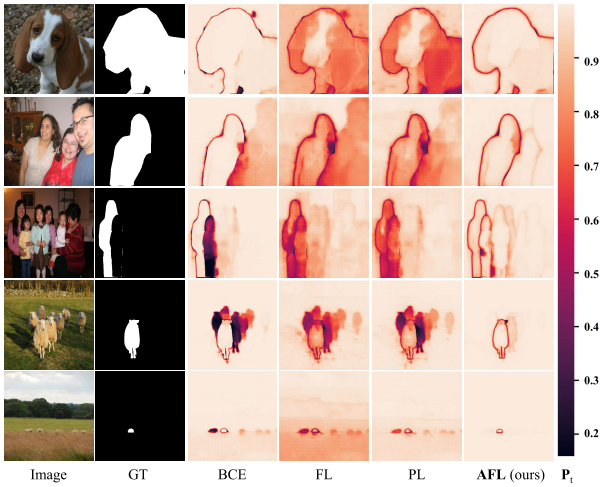


Fig. 3. Difficulty confidence visualization of different loss functions on the SBD [17] training dataset. From left to right are the image, the GT, and the $\mathbf{P}_t$ plot of BCE [39], FL [36], PL [40], and AFL, respectively.

applied in IIS [6], [8], [23] can be expressed as

$$\ell_{\text{bce}} = -\sum_{i=1}^{\mathcal{N}} \log\left(\mathbf{P}_t^i\right) \tag{1}$$

where $\mathbf{P}_t^i = \begin{cases} \mathbf{P}_n^i, & \text{if} \quad \mathbf{y}_t^i = 1 \\ 1 - \mathbf{P}_n^i, & \text{else} \quad \mathbf{y}_t^i = 0 \end{cases}$. $\mathbf{P}_t^i \in [0, 1]$ is difficulty confidence of each pixel and $\mathbf{P}_n^i \in \mathbf{P}_n$, $\mathbf{P}_t^i \in \mathbf{P}_t$ and $\mathbf{y}_t^i \in \mathbf{y}_t$. $\mathcal{N} = HW$, where H and W are the height and width of $\mathbf{P}_n$.

However, ℓ_{bce} treats hard and easy pixels equally [36], we call this property of BCE "difficulty-equal" [a theoretical proof in (13)], which will result in many hard pixels not being effectively segmented. To address those issues, FL (ℓ_{fl}) [36] is introduced in IIS works [27], [30], [31], [51], where

$$\ell_{\text{fl}} = -\sum_{i=1}^{\mathcal{N}} \left(1 - \mathbf{P}_t^i\right)^{\gamma} \log\left(\mathbf{P}_t^i\right) \tag{2}$$

where $(1 - \mathbf{P}_t^i)^{\gamma}$ is a difficulty modifier, $\gamma \in [0, 5]$, the larger of γ, the greater weight for hard pixels. ℓ_{fl} can solve the imbalance of hard and easy pixels due to its "difficulty-oriented" property [a theoretical proof in (14)].

However, study [40] points out that the coefficients of each Taylor term in (2) are not optimal, so the PL (ℓ_{pl}) is proposed as a correction scheme to improve the performance of ℓ_{fl}. The ℓ_{pl} can be expressed as

$$\ell_{\text{pl}} = -\sum_{i=1}^{\mathcal{N}} \left[\left(1 - \mathbf{P}_t^i\right)^{\gamma} \log\left(\mathbf{P}_t^i\right) + \alpha \left(1 - \mathbf{P}_t^i\right)^{\gamma+1} \right] \tag{3}$$

where the α is a coefficient of ℓ_{pl}.

Such improvements are beneficial, but as shown in Fig. 3, we observe that pixels around the click target boundaries (e.g., the person next to the lady in white in the second row) are still difficult to classify correctly. We define such pixels as low-confidence easy pixels or ambiguous pixels in the IIS task. This phenomenon occurs because both BCE [39] and FL [36] suffer from gradient swamping. The difference is that the gradient swamping of BCE is mainly the result of a large number of easy pixels swamping the gradients of hard pixels. In contrast, the gradient swamping of FL results from the gradients of ambiguous pixels being overwhelmed by those of hard pixels. This dilemma hinders the excellent performance of ambiguous pixels and further exacerbates the interaction ambiguity of the IIS model. We term this phenomenon "gradient swamping by FL-based loss of ambiguous pixels in IIS."

B. Interclass Click Ambiguity Resolution

Previous studies [6], [27] have experimented with using prior clicks as prior knowledge to help models converge faster and focus more precisely on local features. Motivated by this, we rethink interclass click ambiguity of mask-fixed-based IIS methods and address them by exploring a CAMD.

1) Pixel-Level Multiscale Mask Transformer Decoder: To obtain representations with more subtle differences of different clicks, we expand the fusion features $\mathbf{F}$ to 1/8, 1/16, and 1/32 resolution based on the feature pyramid networks [52], denoted as $\mathbf{F}_0$, $\mathbf{F}_1$, and $\mathbf{F}_2$. Then, $\mathbf{F}_0$, $\mathbf{F}_1$, and $\mathbf{F}_2$ are passed to a six multiscale deformable attention transformer [53], to obtain $\mathbf{F}_0'$, $\mathbf{F}_1'$, and $\mathbf{F}_2'$, respectively. Finally, a lateral upsampling layer is used on the 1/8 feature map to generate $\mathbf{F}_3$ with a resolution of 1/4 as a pixel-by-pixel embedding.

2) Click-Aware Mask-Adaptive Transformer Decoder: The proposed CAMD primarily consists of the CAAM, self-attention module, feed-forward network, and the click- and mask prediction heads.

a) Click-aware attention module: Given a set feature of positive clicks $\mathbf{C}_p' \in \{\mathbf{C}_p^1, \mathbf{C}_p^2, \ldots, \mathbf{C}_p^z\}$ and their corresponding different scales decoding features $\{\mathbf{F}_0', \mathbf{F}_1', \mathbf{F}_2'\}$, where z is the number of positive clicks among total m clicks and $z < m$. Then, the attention matrix of the proposed CAAM can be expressed as

$$\mathbf{X}_l = \text{softmax}\left(\Psi_{l-1} + \mathbf{Q}_l \mathbf{K}_l^{\mathsf{T}} \mathbf{V}_l\right) + \mathbf{X}_{l-1} \tag{4}$$

where l is the layer index, $\mathbf{X}_l \in \mathbb{R}^{n \times d}$ is the $n \times d$-dimensional query feature of the lth layer, and $\mathbf{Q}_l = f_Q(\mathbf{X}_{l-1}) \in \mathbb{R}^{n \times d}$. $\mathbf{X}_0$ denotes the input query feature of the transformer decoder. $\mathbf{K}_l$, $\mathbf{V}_l \in \mathbb{R}^{H_l W_l \times d}$ are the image features under the transformations f_k and $f_v(*)$ from $\{\mathbf{F}_0', \mathbf{F}_1', \mathbf{F}_2'\}$, respectively, and H_l and W_l are the spatial resolutions of the $\{\mathbf{F}_0', \mathbf{F}_1', \mathbf{F}_2'\}$. f_Q, f_k, and f_v are linear transformations. Ψ_{l-1} is click attention matrix and can be obtained via

$$\Psi_{l-1} = \psi\left[\omega_f\left(\mathbf{C}_p'\right)[\mathbf{Q}_l]_+^{\mathsf{T}}\right] \& \mathbf{M}_{l-1} \tag{5}$$

where ω_f is the click-aware module, which consists of a max pooling layer and a linear layer. ψ denotes the linear mapping layer, $[*]_+$ represents the max function, & represents the and operation. Ψ_0 is the binary click prediction obtained from X_0, i.e., before the query features are fed to the transformer decoder. $\mathbf{M}_{l-1}$ represents the attention mask of the previous transformer block, which can be calculated by the following equation:

$$\mathbf{M}_{l-1}(i, j) = \begin{cases} 0, & \text{if} \quad \mathbf{F}_3\left(\hat{\mathbf{y}}_t^{l-1}\right)^{\mathsf{T}}(i, j) = 1 \\ -\infty, & \text{otherwise} \end{cases} \tag{6}$$

where $\mathbf{F}_3(\hat{\mathbf{y}}_t^{l-1})^\top \in [0,1]^{n\times H_l W_l}$ is the binarized mask with a threshold of 0.5 resized by the (l–1)th transformer decoder.

b) Mask prediction head: The mask prediction head ($\mathbf{G}_{\text{mask}}$) consists of three MLP layers of shape $n\times d$. $\mathbf{G}_{\text{mask}}$ takes predicted object query $\mathbf{Q}_l$ as input and outputs a set of click-based prediction ($\hat{\mathbf{y}}_t$), where $\hat{\mathbf{y}}_t = \mathbf{G}_{\text{mask}}(\mathbf{Q}_l)$.

c) Clicks prediction head: The click prediction head ($\mathbf{G}_{\text{click}}$) consists of a linear layer of shape $n\times 2$ with a softmax. Given a $\mathbf{Q}_l \in \mathbb{R}^d$, $\mathbf{G}_{\text{click}}$ takes $\mathbf{Q}_l$ as input and outputs a click prediction ($\hat{c}$), where $\hat{c} = \mathbf{G}_{\text{click}}(\mathbf{Q}_l)$.

3) Mask-Adaptive Matching Strategy: To discriminate and generate a mask without ambiguity, we use the Hungarian algorithm [38], [54] to find the optimal permutation $\hat{\sigma}$ generated between the $\mathbf{P}_n$ and $\mathbf{y}_t$ during the mask-adaptive optimization process, and finally to optimize the object target-specific losses. Accordingly, we search for a permutation $\hat{\sigma} \in \mathbf{S}_{n_q}$ with the lowest total cost

$$\hat{\sigma} = \underset{\sigma\in\mathbf{S}_{n_q}}{\operatorname{argmin}} \sum_{i=1}^{n_q} \mathbb{1}_{\{\hat{\mathbf{y}}_p\neq\emptyset\}} \mathbb{L}_{\text{total}}\left(\hat{\mathbf{y}}_p\sigma(i), \mathbf{y}_p^i\right) \tag{7}$$

where $\mathbb{L}_{\text{total}}$ is total loss can be obtained in (22), $\hat{\mathbf{y}}_p$ is the predicted via click instance and $\hat{\mathbf{y}}_p = \{\hat{\mathbf{y}}_t^i\}_{i=1}^{n_q} = (\hat{\mathbf{y}}_t, c_m)$. n_q is the number of predicted via the click instance and the m is the number of click. Similarly, $\mathbf{y}_p$ is the $\mathbf{y}_t$ via click instance, and $\mathbf{y}_p = \{\mathbf{y}_t^i\}_{i=1}^{n_q} = (\mathbf{y}_t, c_m)$.

Inspired by [38], [54], CAMD uses a multiscale strategy to exploit high-resolution features. It feeds continuous feature mappings from the pixel-level multiscale mask transformer decoder (PMMD) to the continuous CAMD in a cyclic manner. It is worth mentioning that the CAMD enables clicks to propagate over a long range and accelerates the model's convergence. This is because, in previous works [38], [54], the optimization of the query had a random property. However, CAMD can correctly pass the click to the fine mask in $\hat{\mathbf{y}}_p$, thus achieving the convergence of the model faster and locating the objects effectively.

C. Intra-Class Click Ambiguity Optimization

There are no studies focused on solving the "gradient swamping" in image segmentation. Existing methods only explored the imbalance of positive and negative pixels [41], [42], and the imbalance of hard and easy pixels [36], [40], [49] of BCE. Unlike these works, we rethink the gradient swamping of FL and propose AFL based on the gradient theory of BCE and FL. It enables the model to adapt the learning strategy according to the global training situation, mitigating the gradient swamping.

1) Adaptive Difficulty Adjustment: Observing (2) and (3), ℓ_{fl} and ℓ_{pl} use $(1-\mathbf{P}_t^i)^\gamma$ as the difficulty modifier, and adjust the value of $(1-\mathbf{P}_t^i)^\gamma$ using initial γ according to the difficulty distribution of the dataset empirically (i.e., generally taken as 2 in previous works). However, it is worth noting that the distribution of difficulty varies among samples even in the same dataset (see Fig. 3). Therefore, giving a fixed γ to all training samples of a dataset is a suboptimal option.

To give the model the ability to adaptively adjust $(1-\mathbf{P}_t^i)^\gamma$ according to the different difficulty distributions of the samples and learning levels, we introduce an adaptive difficulty adjustment (ADA) factor γ_a. With γ_a, the $(1-\mathbf{P}_t^i)^\gamma$ of each pixel can be adjusted according to the overall training of the samples, giving the model the capability of "global insight."

Given $\mathbf{I}$, we refer to the foreground as the hard pixels and the number as $\mathcal{H}$ (obtained through $\mathbf{y}_t$). To estimate the overall learning difficulty of the model, we use the $\mathbf{P}_t^i$ of each pixel to represent its learning level. Then, the learning state of the model can be denoted as $\sum_{i=1}^{\mathcal{H}}(\mathbf{P}_t^i)$. Ideally, the optimization goal of the model is to classify all hard pixels completely and accurately. In this case, the optimal learning situation of the model can be represented as $\sum_{i=1}^{\mathcal{H}}(\mathbf{y}_t^i)$. Given the FL properties and the actual learning situation of the model, γ_a can be represented as

$$\gamma_a = 1 - \frac{\sum_{i=1}^{\mathcal{H}}(\mathbf{P}_t^i)}{\sum_{i=1}^{\mathcal{H}}(\mathbf{y}_t^i)}. \tag{8}$$

Then, the difficulty modifier for AFL is

$$(1-P_t)^{\gamma_d}, \text{where} \quad \gamma_d = \gamma + \gamma_a. \tag{9}$$

In summary, AFL can be expressed as

$$\ell_{\text{afl}} = \sum_{i=1}^{\mathcal{N}} -\left(1-\mathbf{P}_t^i\right)^{\gamma_d}\log\left(\mathbf{P}_t^i\right) + \alpha\left(1-\mathbf{P}_t^i\right)^{\gamma_d+1}. \tag{10}$$

2) Adaptive Gradient Representation: For the proposed AFL, a larger γ_d is suitable when a more severe hard-easy imbalance is present. However, for the same $\mathbf{P}_t$, the larger of γ_d, the smaller the loss. It leads to the fact that when one wants to increase the concentration on learning with a severe hard-easy imbalance, it tends to sacrifice a portion of the low-confidence easy pixel's loss in the overall training process.

To address this issue, we first explore the gradient composition of ℓ_{afl} and its properties based on Taylor approximation theory [55] of (1), we can obtain (12)

$$\ell_{\text{bce}} = \sum_{i=1}^{\mathcal{N}}\left[\left(1-\mathbf{P}_t^i\right) + \frac{1}{2}\left(1-\mathbf{P}_t^i\right)^2 + \cdots\right]. \tag{11}$$

To further explore the theoretical connection between the ℓ_{bce} and proposed ℓ_{afl}, we conducted an in-depth study at the gradient level of (10) and obtained

$$\ell_{\text{afl}} = \sum_{i=1}^{\mathcal{N}}\left[(1+\alpha)\left(1-\mathbf{P}_t^i\right)^{\gamma_d+1} + \frac{1}{2}\left(1-\mathbf{P}_t^i\right)^{\gamma_d+2} + \cdots\right]. \tag{12}$$

Next, we derive (11) of ℓ_{bce}, which yields

$$-\frac{\partial\ell_{\text{bce}}}{\partial\mathbf{P}_t} = -\sum_{i=1}^{\mathcal{N}}\nu_i = \sum_{i=1}^{\mathcal{N}}\left[1+\left(1-\mathbf{P}_t^i\right)^1 + \left(1-\mathbf{P}_t^i\right)^2 + \cdots\right] \tag{13}$$

where $\nu_i = \partial\ell_{\text{bce}}/\partial\mathbf{P}_t^i$ is the gradient of BCE for each pixel in the sample. The first term of the ν_i in (13) is always 1, regardless of the difficulty of the pixel, which indicates that the gradient of ℓ_{bce} is "difficulty-equal," treating all pixels equally. Because of this property, ℓ_{bce} can perform competitively in pixels with uniform difficulty distribution, but often performs poorly when the difficulty pixels are imbalanced (see Fig. 3, the visualization of BCE [39]).

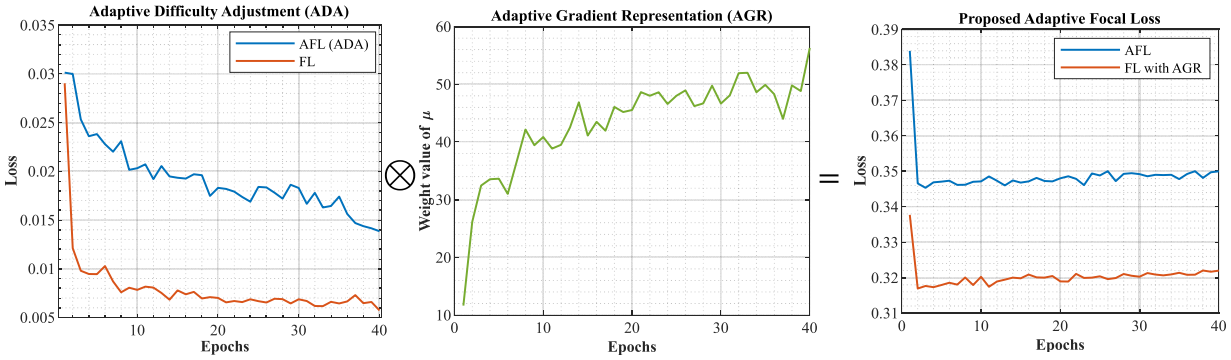


Fig. 4. Visualization plot of the gradient correction of FL [36] by the ADA and AGR is shown in this paper. The left figure is the FL after the ADA adjustment. The middle figure is the gradient adjustment value generated by AGR. The right figure is the FL and AFL being adjusted.

Interestingly, when deriving (12) we can obtain

$$-\frac{\partial \ell_{\text{afl}}}{\partial \mathbf{P}_{\text{t}}}=\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}\left[(1+\alpha)(1+\gamma_{\text{d}})+\left(1+\frac{\gamma_{\text{d}}}{2}\right)\left(1-\mathbf{P}_{\text{t}}^{i}\right)+\cdots\right]. \tag{14}$$

In contrast to ℓ_{bce}, the gradients of ℓ_{afl} and ℓ_{fl} are "difficulty-oriented," so the gradient swamping of ℓ_{afl} refer to the ambiguous pixels being swamped by the hard pixels.

In view of the above analysis, we expect that ℓ_{afl} can have the ability of ℓ_{fl} to classify hard and easy pixels while taking into account the focus on ambiguous pixels like ℓ_{bce}. Observing (13) and (14), we can see that the gradient of ℓ_{afl} can be approximated as the ℓ_{bce} gradient discarding the first γ_{d} terms. Therefore, there are two ways to make the ℓ_{afl}'s gradient approximate to ℓ_{bce}.

1) Adding the discarding gradients of ℓ_{bce} to make ℓ_{afl} to approximate ℓ_{bce}.
2) Multiplying by an adaptive gradient representation (AGR) factor μ that forces the ℓ_{afl} approximate to ℓ_{bce}.

Unfortunately, the scheme (1) relies on adding the missing terms $\sum_{i=1}^{\mathcal{N}}\sum_{j=1}^{\gamma_{\text{d}}}(1-\mathbf{P}_{\text{t}}^{i})^{j+1}$ in (14), which still retains the stiff "difficulty-equal" properties of ℓ_{bce}. Therefore, our primary focus is on exploring the gradient approximation scheme of (2).

Based on this conjecture, we summarized the gradient of ℓ_{bce} with respect to that of ℓ_{afl} at the gradient level. Then, we find that the adjustment of ℓ_{afl} to the ℓ_{bce} is mainly focused on the vertical direction of the gradient, e.g., in the following equation:

$$-\frac{\partial \ell_{\text{afl}}}{\partial \mathbf{P}_{\text{t}}}=\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}\begin{bmatrix}1 & +\gamma_{\text{d}}\left(1+\alpha+\frac{\alpha}{\gamma_{\text{d}}}\right)\\ +\left(1-\mathbf{P}_{\text{t}}^{i}\right) & +\frac{\gamma_{\text{d}}}{2}\left(1-\mathbf{P}_{\text{t}}^{i}\right)\\ +\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{2} & +\frac{\gamma_{\text{d}}}{3}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{2}\\ +\ldots & +\ldots\end{bmatrix}. \tag{15}$$

In (15), the left column of the polynomial is ν_i. The right column of the polynomial is the correction gradient of vertical generated by $(1-\mathbf{P}_{\text{t}}^{i})^{\gamma_{\text{d}}}$, which we define as $\nabla_{\mathcal{B}}$.

Since there is only a quantitative difference between $\nabla_{\mathcal{B}}$ and ν_i, we consider $\forall\delta, -\nu_i=\delta\nabla_{\mathcal{B}}$, where $\delta\in[0,1]$. Based on the above analysis, we can obtain the following equation:

$$\begin{aligned}\frac{\partial \ell_{\text{afl}}}{\partial \mathbf{P}_{\text{t}}}&=\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}[\nu_i+\delta\nabla_{\mathcal{B}}]\\&=\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}\left[(1+\delta\gamma_{\text{d}})\nu_i\right].\end{aligned} \tag{16}$$

The model classification reaches optimality under the condition that $\sum_{i=1}^{\mathcal{N}}(1-\mathbf{P}_{\text{t}}^{i})^{\gamma_{\text{d}}}\nu_i$ satisfies each $(1-\mathbf{P}_{\text{t}}^{i})^{\gamma_{\text{d}}}\in\sum_{i=1}^{\mathcal{N}}(1-\mathbf{P}_{\text{t}}^{i})^{\gamma_{\text{d}}}=0$, and each $\nu_i\in\sum_{i=1}^{\mathcal{N}}\nu_i=1$. Then, based on Chebyshev's inequality [56], we can obtain

$$\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}\nu_i=\frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}\sum_{i=1}^{\mathcal{N}}\nu_i. \tag{17}$$

Collating (13), (16), and (17), we can get

$$\frac{\partial \ell_{\text{bce}}}{\partial \mathbf{P}_{\text{t}}}=\frac{\partial \ell_{\text{afl}}/\partial \mathbf{P}_{\text{t}}}{\frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}(1+\delta\gamma_{\text{d}})}. \tag{18}$$

Observing (16) and (18), the gradient of ℓ_{afl} can be represented as an approximate total gradient of ℓ_{bce} by introducing μ, where

$$\mu=\frac{\mathcal{N}}{\sum_{i=1}^{\mathcal{N}}\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}(1+\delta\gamma_{\text{d}})}. \tag{19}$$

Based on the above analysis, μ allows the gradient of the ℓ_{afl} to be balanced between "difficulty-equal" and "difficulty-oriented," which can increase the weight of ambiguity pixels in the training process, thus making the ℓ_{afl} notice the ambiguity pixels rather than focusing more on extremely hard pixels.

3) Adaptive Focal Loss: In summary, ℓ_{afl} is as follows:

$$\ell_{\text{afl}}=\sum_{i=1}^{\mathcal{N}}\left[-\mu\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}}\log\left(\mathbf{P}_{\text{t}}^{i}\right)+\alpha\left(1-\mathbf{P}_{\text{t}}^{i}\right)^{\gamma_{\text{d}}+1}\right]. \tag{20}$$

As shown in Fig. 4, ADA can not only make the ℓ_{afl} focus more on ambiguity pixels but also produce a larger margin compared to FL [36], which indicates that AFL can adjust the training strategy based on different sample distributions and learning situations. On the other hand, the AGR allows the ℓ_{afl} to be balanced between "difficulty-equal" and "difficulty-oriented," which can increase the weight of ambiguity pixels, yielding solving the gradient swamping. In addition, the ADA

can not only make the model focus on ambiguous pixels but is also better at maintaining the stability of the model when using AGR directly on FL. This is reflected by the fact that the loss value of FL drifts more when AGR is used, while AFL does not in the right sub-figure of Fig. 4.

Moreover, it can be easily obtained that when the values of μ and γ_{d} are 0, the proposed AFL is equivalent to PL [40]; when μ, γ_{a}, and α are 0, AFL is equivalent to FL [36]; when μ, γ_{d}, and α are 0, AFL is equivalent to BCE [39]. This flexible quality allows AFL to achieve higher accuracy when compared to other approaches.

D. Model Optimization

1) Mask Loss: The proposed AFL and Dice Loss [47] are used as the mask loss in this work. Here

$$\ell_{\mathrm{mask}}\left(\hat{\mathbf{y}}_{\mathrm{p}}^{i}, \mathbf{y}_{\mathrm{p}}^{i}\right) = \lambda_{\mathrm{afl}}\ell_{\mathrm{afl}}\left(\hat{\mathbf{y}}_{\mathrm{p}}^{i}, \mathbf{y}_{\mathrm{p}}^{i}\right) + \lambda_{\mathrm{dice}}\ell_{\mathrm{dice}}\left(\hat{\mathbf{y}}_{\mathrm{p}}^{i}, \mathbf{y}_{\mathrm{p}}^{i}\right) \quad (21)$$

where $\hat{\mathbf{y}}_{\mathrm{p}}^{i}$ is the click-predicted instances permuted according to the optimal permutation $\hat{\sigma} \in \mathbf{S}_{n_q}$.

2) Click Loss: The cross-entropy loss is used as the click loss, and the click loss $\ell_{\mathrm{cli}}(\hat{c}_m, c_m)$ is intended to compute classification confidence for each click, which in turn ensures that each click can be computed efficiently.

3) Total Loss: Ultimately, the total loss can be expressed as

$$\mathbb{L}_{\mathrm{total}}\left(\hat{\mathbf{y}}_{\mathrm{p}}^{i}, \mathbf{y}_{\mathrm{p}}^{i}\right) = \sum_{i=1}^{\mathcal{N}} \lambda_{\mathrm{mask}}\ell_{\mathrm{mask}}\left(\hat{\mathbf{y}}_{\mathrm{p}}^{i}, \mathbf{y}_{\mathrm{p}}^{i}\right) + \lambda_{\mathrm{cli}}\ell_{\mathrm{cli}}\left(\hat{c}_m, c_m\right). \quad (22)$$

For predictions that match the $\hat{\mathbf{y}}_{\mathrm{p}}^{i}$ and 0.1 for "unclick," i.e., predictions that do not match any GT.

IV. Experiments

A. Datasets

We evaluate the proposed AdaptiveClick using the following well-recognized datasets widely used in IIS tasks.

1) SBD [17]: Comprising 8498 images for training and 2857 images for testing, this dataset is characterized by its scene diversity and is frequently utilized for IIS tasks.

2) COCO-LIVS [6]: This dataset comprises 118 K training images (with 1.2 M instances), which is widely adopted due to its diverse class distributions [6], [27], [51], [57].

3) GrabCut [58]: The GrabCut dataset contains 50 images, which have relatively simple appearances and have also been commonly used for evaluating the performance of different IIS methods [6], [27], [51], [57].

4) Berkeley [59]: This dataset includes 100 images, sharing some small object images with GrabCut [58]. It poses challenges for IIS models due to these similarities.

5) DAVIS [60]: Initially designed for video image segmentation, the 50 videos are divided into 345 frames for testing. The dataset features images with high-quality masks.

6) Pascal VOC [61]: It is composed of 1449 images (3427 instances). We assess segmentation performance on this validation set, as in [16], [18], and [27].

7) ssTEM [62]: The ssTEM dataset contains two image stacks, each with 20 medical images. We use the same stack as in [27] and [7] to evaluate model effectiveness.

8) BraTS [63]: The BraTS dataset includes 369 Magnetic Resonance Images (MRI) volumes. We test our model on the same 369 slices as in [27] and [7].

9) OAIZIB [64]: OAIZIB dataset consists of 507 MRI volumes. We test our model on the same 150 slices (300 instances) as in [22] and [27].

B. Implementation Details

For data embedding, given the clicks with previous mask $\{\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^m, \mathbf{M}_{\mathrm{pre}}\}$ and $\mathbf{I}$ to obtain encode features $\mathbf{C}_{\mathrm{t}}$ and $\mathbf{I}_{\mathrm{t}}$ via patch embedding, then they are fused like [6], [21], [27]. For feature encoding, we use ViT-Base (ViT-B) and ViT-Huge (ViT-H) as our backbone. For feature decoding, the designed PMMD is used to encode the fusion feature and then sent to the CAMD, where CAMD consists of three transformer blocks, i.e., $L = 3$ (nine layers in total), and $\mathbf{Q}$ is set to 10.0. For post-processing, we multiply the click and mask confidence to obtain the final confidence matrix, then generate the $\mathbf{P}_{\mathrm{n}}$ as done in [38] and [54]. In addition, $\mathbb{L}_{\mathrm{total}}$ is added to each CAMD layer to guarantee the performance of the model.

For AdaptiveClick training, we train for 60 epochs on the SBD [17] and the COCO-LVIS dataset [6]; the initial learning rate is set to 5×10^{-5}, and then reduced by ten times in the 40-th epoch. The image is cropped to size 448×448 pix. λ_{cli} is 2, λ_{mask} is 1, λ_{afl} is 5, and λ_{dice} is 5 in this work. Adam is used to optimize the training with the parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is 48 for training ViT-B and 24 for training ViT-H, respectively. All experiments are trained and tested on NVIDIA Titan RTX 6000 GPUs.

In addition, to assess the robustness and generalization of AFL, we have incorporated it into state-of-the-art IIS methods and ensured consistency in other parameter settings. The hyperparameters γ, δ, and α for the proposed AFL are fixed at 2, 0.4, and 1.0, respectively, throughout the training process.

C. Evaluation Metrics

Following previous works [6], [16], [21], we adopt the combination of number of clicks (NoCs) and Intersection over Union (IoU) at 0.85 (NoC85) and 0.9 (NoC90) as evaluation metrics. Here, NoC90 indicates that the IoU of the mask obtained by the IIS model is 90 when NoC is k. In particular, the IIS task sets the maximum NoC to 20 (samples with NoC > 20 are considered failures). Therefore, the lower the value of NoC85 and NoC90, the better. We also use the average IoU given k clicks (mIoU@k) as an evaluation metric to measure the segmentation quality given a fixed NoC.

D. Comparison Methods

We validate our approach against state-of-the-art IIS methods and loss functions, specifically.

1) IIS Methods: IIS methods including DIOS [26], Latent diversity [19], BRS [23], f-BRS-B [24], IA-SA [37], FCA-Net [16], PseudoClick [7], CDNet [18], RITM [6], FocalClick [21], FocusCut [8], GPCIS [51], SimpleClick [27], FCFI [28], InterFormer [30], DynaMITe [31], SAM [29], SEEM [32], EMC [65], FDRN [66], iCMFormer [67], and VTMR [68], all of which achieve competitive results.

TABLE I

COMPARISON IN NOC85 AND NOC90 BETWEEN ADAPTIVECLICK AND STATE-OF-THE-ART METHODS TRAINED ON AUGMENTED VOC [61] AND SBD [17] DATASETS AND TESTED ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS. TOP RESULTS WITHIN A GROUP ARE INDICATED IN UNDERLINE AND THE OVERALL TOP RESULTS IN **BOLD**

Method	Backbone	Train Data	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC		Average	
			NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
DIOS [26] CVPR2016	FCN	Augmented VOC	-	6.04	-	8.65	-	-	-	12.58	6.88	-	-	-
FCANet [16] CVPR2020	ResNet101	Augmented VOC	-	2.08	-	3.92	-	-	-	7.57	2.69	-	-	-
Latent Diversity [19] CVPR2018	VGG-19	SBD	3.20	4.79	-	-	7.41	10.78	5.05	9.57	-	-	-	-
BRS [23] CVPR2019	DenseNet	SBD	2.60	3.60	-	5.08	6.59	9.78	5.58	8.24	-	-	-	-
IA-SA [37] ECCV2020	ResNet101	SBD	-	3.07	-	4.94	-	-	5.16	-	-	-	-	-
f-BRS-B [24] CVPR2020	ResNet50	SBD	2.50	2.98	-	4.34	5.06	8.08	5.39	7.81	-	-	-	-
CDNet [18] CVPR2021	ResNet34	SBD	1.86	2.18	1.95	3.27	5.18	7.89	5.00	6.89	3.61	4.51	3.52	4.95
RITM [6] ICIP2022	HRNet18	SBD	1.76	2.04	1.87	3.22	3.39	5.43	4.94	6.71	2.51	3.03	2.89	4.09
PseudoClick [7] ECCV2022	HRNet18	SBD	1.68	2.04	1.85	3.23	3.38	5.40	4.81	6.57	2.34	2.74	2.81	4.00
FocalClick [21] CVPR2022	SegF-B0	SBD	1.66	1.90	-	3.14	4.34	6.51	5.02	7.06	-	-	-	-
FcousCut [8] CVPR2022	ResNet101	SBD	1.46	1.64	1.81	3.01	3.40	5.31	4.85	6.22	-	-	-	-
GPCIS [51] CVPR2023	SegF-B0	SBD	1.60	1.76	1.84	2.70	4.16	6.28	4.45	6.04	-	-	-	-
FCFI [28] CVPR2023	ResNet101	SBD	1.64	1.80	-	2.84	3.26	5.35	4.75	6.48	-	-	-	-
EMC [65] CVPR2023	HRNet18	SBD	1.74	1.84	-	3.03	3.38	5.51	5.05	6.71	2.37	-	-	-
FDRN [66] ACMMM2023	SegF-B0	SBD	1.58	1.78	-	3.08	4.18	6.20	4.78	6.66	-	-	-	-
SimpleClick [27] ICCV2023	ViT-B	SBD	1.40	1.54	1.44	2.46	3.28	5.24	4.10	5.48	2.38	2.81	2.52	3.51
AdaptiveClick (ours)	ViT-B	SBD	**1.38**	**1.46**	**1.38**	**2.18**	**3.22**	**5.22**	**4.00**	**5.14**	**2.25**	**2.66**	**2.45**	**3.33**

TABLE II

COMPARISON IN NOC85 AND NOC90 BETWEEN ADAPTIVECLICK AND STATE-OF-THE-ART METHODS TRAINED ON THE COCO-LVIS DATASET [6] AND TESTED ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS. † DENOTES THE RESULT IS FROM SEEM

Method	Backbone	Train Data	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC		Average	
			NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
f-BRS-B [24] CVPR2020	HRNet32	COCO-LVIS	1.54	1.69	-	2.44	4.37	7.26	5.17	6.50	-	-	-	-
CDNet [18] ICCV2021	ResNet34	COCO-LVIS	1.40	1.52	1.47	2.06	4.30	7.04	4.27	5.56	2.74	3.30	2.84	3.90
RITM [6] ICIP2022	HRNet32	COCO-LVIS	1.46	1.56	1.43	2.10	3.59	5.71	4.11	5.34	2.19	2.57	2.56	3.46
FocalClick [21] CVPR2022	SegF-B0	COCO-LVIS	1.40	1.66	1.59	2.27	4.56	6.86	4.04	5.49	2.97	3.52	2.91	3.96
FocalClick [21] CVPR2022	SegF-B3	COCO-LVIS	1.44	1.50	1.55	1.92	3.53	5.59	3.61	4.90	2.46	2.88	2.52	3.56
PseudoClick [7] ECCV2022	HRNet32	COCO-LVIS	1.36	1.50	1.40	2.08	3.38	5.54	3.79	5.11	1.94	2.25	2.37	3.30
FCFI [28] CVPR2023	HRNet18	COCO-LVIS	1.38	1.46	-	1.96	3.63	5.83	3.97	5.16	-	-	-	-
EMC [65] CVPR2023	SegF-B3	COCO-LVIS	1.42	1.48	-	2.35	3.44	5.57	4.49	5.69	2.23	-	-	-
FDRN [66] ACMMM2023	SegF-B3	COCO-LVIS	1.42	1.44	-	1.80	3.74	5.57	3.55	4.90	-	-	-	-
DynaMITe [31] ICCV2023	SegF-B3	COCO-LVIS	1.48	1.58	1.34	1.97	3.81	6.38	3.81	5.00	-	-	-	-
† SAM [29] ICCV2023	ViT-B	COCO-LVIS	-	-	-	-	6.50	9.76	-	-	3.30	4.20	-	-
† SEEM [32] NeurIPS2023	DaViT-B	COCO-LVIS	-	-	-	-	6.67	9.99	-	-	3.41	4.33	-	-
InterFormer [30] ICCV2023	ViT-B	COCO-LVIS	1.38	1.50	1.99	3.14	3.78	6.34	4.10	6.19	-	-	-	-
InterFormer [30] ICCV2023	ViT-L	COCO-LVIS	**1.26**	**1.36**	1.61	2.53	3.25	5.51	4.54	5.21	-	-	-	-
iCMFormer [67] ICCVW2023	ViT-B	COCO-LVIS	1.42	1.52	1.40	1.86	3.29	5.30	3.40	5.06	-	-	-	-
SimpleClick [27] ICCV2023	ViT-H	COCO-LVIS	1.38	1.50	1.36	1.75	2.85	4.70	3.41	4.78	1.76	1.98	2.15	2.94
VTMR [68] AAAI2024	SegF-B3	COCO-LVIS	1.38	1.42	1.44	1.72	3.55	5.53	3.26	4.82	-	-	-	-
AdaptiveClick (ours)	ViT-H	COCO-LVIS	1.32	1.38	**1.32**	**1.64**	**2.84**	**4.68**	**3.19**	**4.60**	**1.74**	**1.96**	**2.08**	**2.85**

2) Loss Functions: We compare the AFL with other loss functions, including BCE [39], Soft IoU [46], FL [36], NFL [49], WBCE [41], Balanced CE [42], and PL [40] losses, which are widely used in IIS and semantic segmentation tasks [48]. Specifically, in FL and NFL, the value of γ is 2, and the balanced weight is 0.5. For PL, the value of γ is 2, the balanced weight is 0.5, and the value of α is 1. Moreover, the positive weight for WBCE is 3, while the positive and negative weights for the Balanced CE are 3 and 1, respectively.

TABLE III

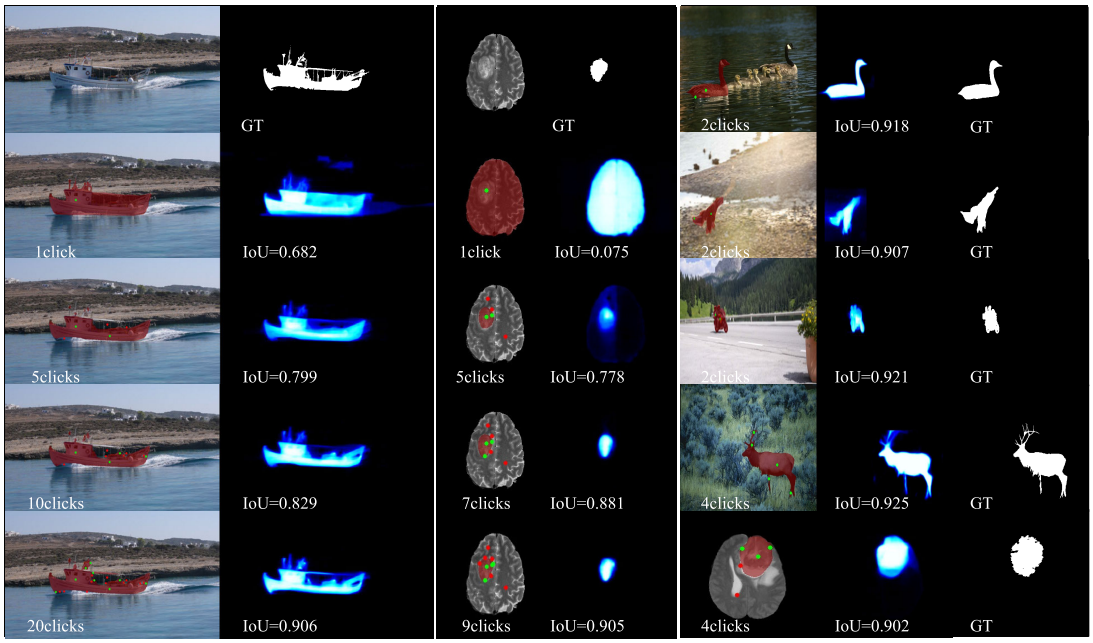
COMPARISON IN NOC85 AND NOC90 BETWEEN ADAPTIVECLICK AND STATE-OF-THE-ART METHODS TRAINED ON SBD [17] AND TESTED ON SSTEM [58], BRATS [59], AND OAIZIB [59]

Method	Backbone	Train Data	ssTEM		BraTS		OAIZIB	
			NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
CDNet [18] ICCV2021	ResNet34	SBD	11.10	14.65	17.07	18.86	19.56	19.95
RITM [6] ICIP2022	HRNet18	SBD	3.71	5.68	**8.47**	**12.59**	17.70	19.95
SimpleClick [27] ICCV2023	ViT-B	SBD	3.78	5.21	9.93	13.90	18.44	19.90
AdaptiveClick (ours)	ViT-B	SBD	**3.17**	**4.56**	10.79	14.39	**17.43**	**19.62**

E. Model Analysis

1) Comparison With State-of-the-Art Methods: Tables I and II empirically analyze the performance of AdaptiveClick and state-of-the-art methods using SBD and COCO-LIVS as training datasets. Likewise, Tables III and IV display the performance of AdaptiveClick and relevant published methods in medical images, respectively.

a) Evaluation on the natural images: In Tables I and II, we conduct a comparative analysis between AdaptiveClick and state-of-the-art IIS methods. First, we provide an experimental comparison of AdaptiveClick using ViT-B and ViT-H as the backbones, along with the proposed AFL as the loss function. AdaptiveClick achieves an average accuracy improvement of 0.074, 0.174, and 0.07, 0.09 on SBD and COCO-LIVS, respectively, when compared to existing state-of-the-art methods on NoC85 and NoC90. This underscores the robustness and superior accuracy of AdaptiveClick in addressing interaction ambiguity and gradient swamping challenges.



(a) (b) (c)

Fig. 5. Segmentation results on natural and medical datasets. The backbone is ViT-B trained on the SBD dataset [17]. The PMs are shown in blue; the masks are overlaid in red on the original images. The clicks are shown as green (positive click) or red (negative click) dots on the image. (a) Challenging case of natural images. (b) Challenging case of medical images. (c) Five normal cases of natural/medical images.

TABLE IV

COMPARISON IN NOC85 AND NOC90 BETWEEN ADAPTIVECLICK AND STATE-OF-THE-ART METHODS TRAINED ON COCO-LVIS [6] AND TESTED ON SSTEM [58], BRATS [59], AND OAIZIB [59]

Method	Backbone	Train Data	ssTEM		BraTS		OAIZIB	
			NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
CDNet [18] ICCV2021	ResNet34	COCO-LVIS	4.15	8.45	10.51	14.80	17.42	19.81
RITM [6] ICIP2022	HRNet32	COCO-LVIS	**2.74**	**4.06**	7.56	11.24	15.89	19.27
FocalClick [21] CVPR2022	SegF-B3	COCO-LVIS	3.95	5.05	7.17	11.19	12.93	19.23
SimpleClick [27] ICCV2023	ViT-H	COCO-LVIS	4.27	5.45	6.73	10.27	14.93	18.95
AdaptiveClick (ours)	ViT-H	COCO-LVIS	3.95	4.97	**6.51**	**9.77**	**12.65**	**16.69**

TABLE V

COMPUTATION COMPARISON FOR MODEL PARAMETERS, FLOPS, GPU MEMORY, AND SPEED WITH DIFFERENT STATE-OF-THE-ART IIS METHODS

Method (backbone, size)	Params/M	FLOPs/G	Mem/G	↓ SPC/ms
RITM (HRNet32, 400)$_{\text{ICIP2022}}$ [6]	30.95	83.12	0.50	54
iSegFormer (Swin-L, 400)$_{\text{MICCAI2022}}$ [22]	195.90	302.78	2.14	44
FocalClick (SegF-B3, 256)$_{\text{CVPR2022}}$ [21]	45.66	24.75	0.32	53
FocusCut (ResNet101, 384)$_{\text{CVPR2022}}$ [8]	59.35	100.76	0.89	355
SimpleClick (ViT-B, 448)$_{\text{ICCV2023}}$ [27]	96.46	169.78	0.87	54
InterFormer (ViT-B, 512)$_{\text{ICCV2023}}$ [30]	120.39	533.70	1.40	360
iCMFormer (ViT-B, 512)$_{\text{ICCVW2023}}$ [67]	124.81	297.54	-	78
AdaptiveClick (ViT-B, 448) (ours)	116.41	269.81	1.28	74

Simultaneously, we have conducted a statistical analysis of the average NoC85 and NoC90 for various IIS methods. The results demonstrate that AdaptiveClick outperforms all comparison methods, achieving more accurate segmentation with fewer clicks.

b) Evaluation on medical images: The results of AdaptiveClick and state-of-the-art IIS methods for medical images are presented in Tables III and IV. It is noteworthy that prevalent IIS methods generally demonstrate suboptimal performance when applied to medical image segmentation, attributable to two key factors. First, the majority of IIS methods have not undergone training on specialized medical datasets, despite the introduction of such datasets in some studies for training purposes. Second, these IIS methods often display limited generalization capabilities. In contrast, AdaptiveClick consistently achieves high accuracy across various medical datasets, highlighting the method's versatility. In addition, Fig. 5 is the result of some challenging and normal examples, and our AdaptiveClick can segment the specified object in the image across both natural and medical datasets.

c) Computational analysis: Table V provides computation statistics results for state-of-the-art IIS methods. As in [27], we evaluate the AdaptiveClick and compare it with existing methods on GrabCut [58]. The experimental results reveal that, although the proposed mask-adaptive method introduces some computational overhead, with a memory consumption of 1.28 G and a running speed of 74 ms, it still meets the requirements for real-time annotation. This demonstrates that AdaptiveClick not only effectively addresses the interaction ambiguity but also enhances segmentation accuracy while meeting real-time demands.

2) Effectiveness of AFL: The performance of different loss functions on the SBD [17] and COCO-LIVS [6] datasets is illustrated in Tables VI and VII, respectively. Tables VIII and IX report the experimental results of embedding AFL into existing methods on SBD and COCO-LIVS training datasets, respectively.

a) Comparison with state-of-the-art loss functions: In Table VI, the efficacy of AFL is demonstrated through an average improvement of 0.014∼0.07 and 1.136∼2.504 on NoC85 and NoC90, respectively (SBD training dataset). This improvement is observed in comparison to BCE, WBCE,

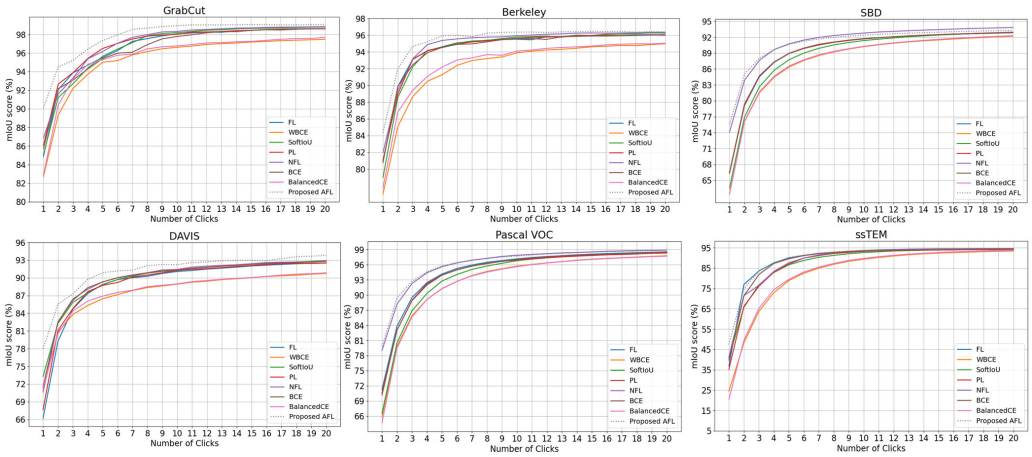


Fig. 6. Convergence analysis between AFL and state-of-the-art loss functions. AdaptiveClick is used as the baseline and SBD [17] as the training dataset. The test datasets are GrabCut [58], Berkeley [59], SBD [17], DAVIS [60], Pascal VOC [61], and ssTEM [58], respectively. The metric is the mean IoU given k clicks. Overall, our models require fewer clicks for a given accuracy level. The methods from top to bottom are FL [36], WBCE [41], Soft IoU [46], PL [40], NFL [49], BCE [39], Balanced CE [42], and AFL, respectively. Our AdaptiveClick in general requires fewer clicks for a given accuracy level.

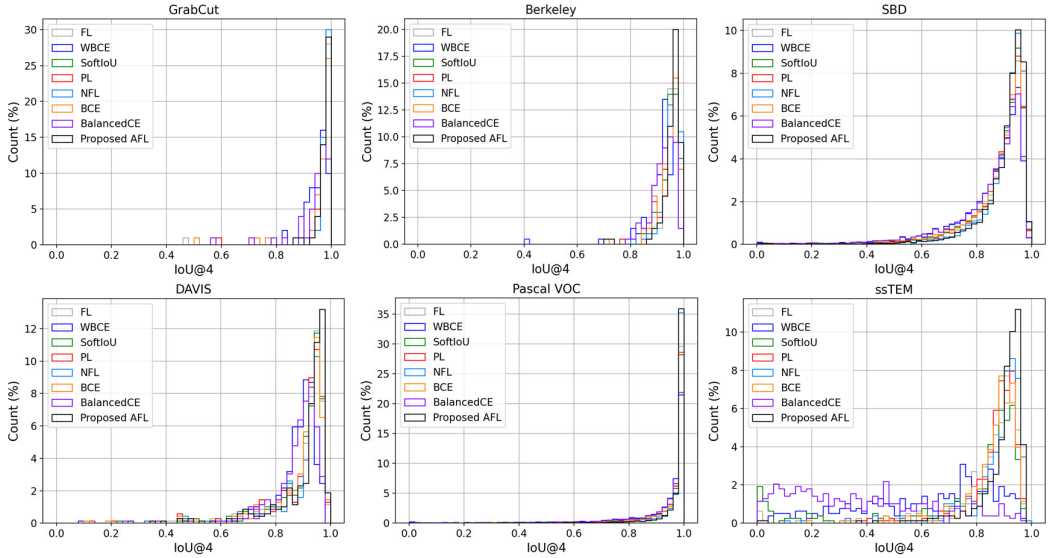


Fig. 7. Histogram analysis between AFL and state-of-the-art loss functions. In this experiment, AdaptiveClick is used as the baseline and COCO-LVIS [6] as the training dataset. The test datasets are GrabCut [58], Berkeley [59], SBD [17], DAVIS [60], Pascal VOC [61], and ssTEM [58], respectively. The methods from top to bottom are FL [36], WBCE [41], Soft IoU [46], PL [40], NFL [49], BCE [39], Balanced CE [42], and the proposed AFL, respectively. Given $k = 4$ clicks, AdaptiveClick obtains the best results (more instances with a higher IoU) than comparison loss functions.

TABLE VI

EVALUATION OF ADAPTIVECLICK OF VIT-B TRAINED ON THE SBD [17] WITH DIFFERENT LOSS FUNCTIONS. WE REPORT NOC85 AND NOC90 ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS

Method	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
BCE [39] ISIMP2004	1.64	1.78	1.69	2.99	4.06	6.39	4.55	6.08	2.85	3.37
WBCE [41] Bioinf.2007	1.66	2.58	2.25	5.20	5.18	8.00	5.43	9.28	3.39	4.12
Balanced CE [42] ICCV2015	1.62	1.84	1.88	3.59	4.55	7.00	5.57	7.35	3.44	4.18
Soft IoU [46] ISVC2016	1.62	1.99	1.78	2.29	4.55	6.48	4.69	6.32	3.13	3.71
FL [36] ICCV2017	1.62	1.84	1.69	2.82	4.11	6.51	4.94	6.50	2.81	3.35
NFL [49] ICCV2019	1.40	1.50	**1.32**	2.24	3.25	5.28	4.03	5.24	2.30	2.75
PL [42] ICLR2022	1.46	1.70	1.68	2.87	4.09	6.42	4.70	6.43	2.85	3.39
AFL (ours)	**1.38**	**1.46**	1.38	**2.18**	**3.22**	**5.22**	**4.00**	**5.14**	**2.25**	**2.66**

TABLE VII

EVALUATION OF ADAPTIVECLICK OF VIT-B TRAINED ON THE COCO-LVIS [6] WITH DIFFERENT LOSS FUNCTIONS. WE REPORT NOC85 AND NOC90 ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS

Method	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
BCE [39] ISIMP2004	1.54	1.72	1.79	2.89	4.30	6.74	4.29	5.76	2.71	3.19
WBCE [41] Bioinf.2007	1.60	1.88	2.15	4.29	5.60	8.61	5.26	8.79	3.32	4.03
Balanced CE [42] ICCV2015	1.74	2.10	2.13	4.32	5.53	8.58	4.90	8.02	3.24	3.93
Soft IoU [46] ISVC2016	1.62	1.70	1.64	2.63	4.19	6.66	4.19	5.60	2.69	3.18
FL [36] ICCV2017	1.54	1.60	1.58	2.62	4.19	6.60	4.12	5.60	2.64	3.12
NFL [49] ICCV2019	**1.34**	1.54	1.55	1.92	3.34	5.48	3.66	4.93	2.06	2.39
PL [42] ICLR2022	1.50	1.66	1.66	2.85	4.26	6.74	4.32	5.68	2.71	3.19
AFL (ours)	**1.34**	**1.48**	**1.40**	**1.83**	**3.29**	**5.40**	**3.39**	**4.82**	**2.03**	**2.31**

Balanced CE, FL, NFL, and PL. The results suggest that, as analyzed in (13) and (14), both BCE and FL-based losses exhibit a certain degree of gradient swamping. The proposed AFL proves effective in mitigating this issue. When comparing the experimental results of AFL with WBCE, Balanced CE, and Soft IoU, it becomes evident that while these losses address the pixel imbalance problem in segmentation, they do not excel in the IIS task, due to inadequate resolution of the more profound gradient swamping issue. This observation is supported by Fig. 6, which illustrates the mean IoU per-

TABLE VIII

RESULTS IN NOC85 AND NOC90 FOR STATE-OF-THE-ART METHODS WITH AND WITHOUT THE PROPOSED AFL. IN THESE EXPERIMENTS, WE TRAINED ON THE SBD DATASET [17] AND TESTED ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS. IT SHOULD BE NOTED THAT ADAPTIVECLICK + NFL INDICATES THE RESULTS OBTAINED BY USING ADAPTIVECLICK BUT USING NFL [49] AS THE TRAINING LOSS. ADAPTIVECLICK, ON THE OTHER HAND, INDICATES THE RESULTS OBTAINED BY USING BOTH THE FRAMEWORK OF THIS ARTICLE AND THE AFL AS LOSS FUNCTIONS. TOP RESULTS WITHIN A GROUP ARE INDICATED IN RED

Method	Backbone	Train Data	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
			NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
CDNet [18] CVPR2021	ResNet34	SBD	1.86	2.18	1.95	3.27	5.18	7.89	5.00	6.38	-	-
CDNet (ours)	ResNet34	SBD	1.82	2.14	1.86	2.88	4.53	7.15	4.78	6.27	-	-
RITM [6] ICIP2022	HRNet18	SBD	1.76	2.04	1.87	3.22	3.39	5.43	4.94	6.71	2.51	3.03
RITM (ours)	HRNet18	SBD	1.60	1.86	1.72	2.93	3.40	5.42	4.72	6.15	2.41	2.92
FocalClick [21] CVPR2022	SegF-B0	SBD	1.66	1.90	-	3.14	4.34	6.51	5.02	7.06	-	-
FocalClick (ours)	SegF-B0	SBD	1.64	1.84	-	3.12	4.36	6.44	4.79	6.50	-	-
FcousCut [8] CVPR2022	ResNet50	SBD	1.60	1.78	1.85	3.44	3.62	5.66	5.00	6.38	-	-
FcousCut (ours)	ResNet50	SBD	1.54	1.72	1.79	3.34	3.50	5.62	4.82	6.28	-	-
SimpleClick [27] ICCV2023	ViT-B	SBD	1.40	1.54	1.44	2.46	3.28	5.24	4.10	5.48	2.38	2.81
SimpleClick (ours)	ViT-B	SBD	1.36	1.52	1.44	2.34	3.10	5.04	4.10	5.51	2.23	2.63
AdaptiveClick + NFL	ViT-B	SBD	1.40	1.50	1.32	2.24	3.25	5.28	4.03	5.24	2.30	2.75
AdaptiveClick (ours)	ViT-B	SBD	1.38	1.46	1.38	2.18	3.22	5.22	4.00	5.14	2.25	2.66

TABLE IX

RESULTS IN NOC85 AND NOC90 FOR STATE-OF-THE-ART METHODS WITH AND WITHOUT THE PROPOSED AFL. IN THESE EXPERIMENTS, WE TRAINED ON COCO-LVIS [6] AND TESTED ON THE GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS

Method	Backbone	Train Data	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
			NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
RITM [6] ICIP2022	HRNet32	COCO-LVIS	1.46	1.56	1.43	2.10	3.59	5.71	4.11	5.34	2.19	2.57
RITM (ours)	HRNet32	COCO-LVIS	1.42	1.51	1.40	1.98	3.48	5.64	3.69	5.26	2.10	2.54
FocalClick [21] CVPR2022	SegF-B3	COCO-LVIS	1.44	1.50	1.55	1.92	3.53	5.59	3.61	4.90	2.46	2.88
FocalClick (ours)	SegF-B3	COCO-LVIS	1.40	1.44	1.52	1.87	3.51	5.49	3.57	4.85	2.44	2.86
SimpleClick [27] ICCV2023	ViT-H	COCO-LVIS	1.38	1.50	1.36	1.75	2.85	4.70	3.41	4.78	1.76	1.98
SimpleClick (ours)	ViT-H	COCO-LVIS	1.40	1.50	1.33	1.71	2.80	4.65	3.23	4.75	1.72	1.93
InterFormer [30] ICCV2023	ViT-L	COCO-LVIS	1.28	1.36	1.61	2.53	3.25	5.51	4.54	5.21	-	-
InterFormer (ours)	ViT-L	COCO-LVIS	1.26	1.32	1.56	2.50	3.21	5.46	4.48	5.18	-	-
AdaptiveClick + NFL	ViT-B	COCO-LVIS	1.34	1.54	1.55	1.92	3.34	5.48	3.66	4.93	2.06	2.39
AdaptiveClick (ours)	ViT-B	COCO-LVIS	1.34	1.48	1.40	1.83	3.29	5.40	3.39	4.82	2.03	2.31

formance for each loss function. The figure affirms that AFL consistently enhances performance across all six datasets.

We also report the segmentation quality achieved with AFL with the COCO-LIVS training dataset. The results in Table VII showcase that the proposed AFL significantly boosts the IIS model performance when compared to other state-of-the-art loss functions, which is consistent with the outcome observed using the SBD training dataset. Fig. 7 shows the IoU statistics on the test dataset with four clicks. It becomes evident that the proposed AFL always maintains the highest IoU and count values for the same number of hits, which confirms that the proposed AFL has stronger robustness and better accuracy compared with different loss functions.

Overall, the proposed AFL significantly enhances the performance of the IIS model and consistently outperforms existing loss functions designed for IIS tasks.

b) Embedding AFL into state-of-the-art IIS methods: As shown in Table VIII, the proposed AFL brings positive effect improvements for almost all compared methods in all experiments on the test dataset. This indicates that AFL is robust enough to be applicable to existing mainstream IIS models. Further, the proposed AFL can bring an average NoC85 and NoC90 performance gain of 0.137, 0.181, and 0.119, 0.103 for CNN- and transformer-based IIS methods, respectively. Comparing the performance of FocalClick, SimpleClick, and AdaptiveClick, it can be seen that the proposed AFL is able to deliver NoC85 and NoC90 improvements for FocalClick, SimpleClick, and AdaptiveClick by 0.23, 0.114, 0.014, 0.142, and 0.098, 0.07, respectively, on the five mainstream test datasets. Comparing the results of CDNet, RITM, and FocusCut, it is clear that the proposed AFL can not only significantly improve the accuracy of the transformer-based model, but also bring performance gains to the CNNs-based model. This reflects that AFL has more performance gains for transformer-based models and certifies that AFL has the potential to unleash the capacity of vision transformers in IIS tasks. Consistent with the above analysis, the same results are confirmed in Table IX on the metrics with COCO-LIVS as the training dataset. In summary, AFL can be applied to different training and test datasets and can adapt to different backbones, loss functions, and different architectures of IIS models with performance gains and strong generalizability.

F. Ablation Studies

1) Influence of Components in AdaptiveClick: Tables X and XI present the performance changes of the AdaptiveClick with and without the proposed CAMD and AFL.

a) Influence of components in adaptiveclick: As in Table X, we first show the performance changes with and without the proposed CAMD component on the AdaptiveClick. The use of the CAMD is effective in improving the performance on the five test datasets. This positive performance

TABLE X

ABLATION STUDIES BY USING OR NOT USING CAMD AND AFL TRAINING ON SBD [17] AND TESTING ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61]

CAMD	AFL	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
		NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
		1.46	1.58	1.56	2.46	3.28	5.32	4.20	5.79	2.42	2.85
√		1.40	1.50	**1.32**	2.24	3.25	5.28	4.03	5.24	2.30	2.75
	√	1.42	1.60	1.58	2.35	3.29	5.26	4.15	5.55	2.33	2.73
√	√	**1.38**	**1.46**	1.38	**2.18**	**3.22**	**5.22**	**4.00**	**5.14**	**2.25**	**2.66**

TABLE XI

ABLATION STUDIES BY USING OR NOT USING ADA AND AGR OF AFL TRAINING ON SBD [17] AND TESTING ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61]

ADA	AGR	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
		NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
		1.62	1.84	1.69	2.82	4.11	6.51	4.94	6.50	2.81	3.35
√		1.50	1.60	1.64	2.57	3.83	6.03	4.36	5.78	2.63	3.12
	√	1.56	1.68	1.54	2.40	3.27	5.23	4.50	5.82	2.32	2.74
√	√	**1.38**	**1.46**	**1.38**	**2.18**	**3.22**	**5.22**	**4.00**	**5.14**	**2.25**	**2.66**

TABLE XII

IMPACT OF DIFFERENT HYPERPARAMETER SETTINGS ON ADAPTIVECLICK. WE TRAINED OUR ADAPTIVECLICK ON SBD [17] AND TESTED ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS

λ_{cli}	λ_{afl}	λ_{dice}	Q	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
				NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
2.0	3.0	5.0	5.0	1.40	1.50	1.40	2.39	3.30	5.32	4.08	5.35	2.31	2.76
4.0	3.0	5.0	5.0	1.40	1.50	1.43	2.33	3.31	5.31	4.10	5.37	2.31	2.76
2.0	4.0	5.0	5.0	1.46	1.60	1.43	2.29	3.33	5.33	3.97	5.38	2.43	2.84
2.0	5.0	5.0	5.0	**1.32**	**1.42**	1.52	2.19	3.27	5.29	4.08	5.27	2.30	2.73
2.0	5.0	5.0	5.0	1.40	1.48	**1.37**	2.21	3.30	5.35	**3.90**	5.29	2.31	2.73
2.0	5.0	3.0	5.0	1.40	1.46	**1.37**	2.21	3.29	5.35	4.08	5.40	2.44	2.81
2.0	5.0	5.0	10.0	1.38	1.46	1.38	**2.18**	**3.22**	**5.22**	4.00	**5.14**	**2.25**	**2.66**
2.0	5.0	5.0	20.0	1.48	1.64	1.47	2.28	3.27	5.24	3.99	5.28	2.28	2.73
2.0	5.0	5.0	100.0	1.40	1.48	**1.37**	2.21	3.29	5.35	4.08	5.40	2.29	2.74

TABLE XIII

PERFORMANCE CHANGE BY DIFFERENT WEIGHTS OF λ_{mask} TRAINED ON SBD [17] AND TESTED ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61] DATASETS

λ_{mask}	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
0.5	1.42	1.58	1.45	2.27	3.33	5.32	**3.81**	5.30	2.35	2.76
1.0	**1.38**	**1.46**	**1.38**	**2.18**	**3.22**	**5.22**	4.00	**5.14**	**2.25**	**2.66**
1.5	1.44	1.56	1.57	2.47	3.34	5.36	4.21	5.41	2.30	2.74
2.0	1.52	1.66	1.60	2.54	3.43	5.46	4.31	5.60	2.34	2.81

TABLE XIV

PERFORMANCE CHANGE BY USING AFL UNDER DIFFERENT WEIGHTS OF δ TRAINED ON SBD [17] AND TESTED ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61]

δ	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
0.0	**1.36**	**1.46**	1.41	2.31	3.21	5.18	4.01	5.20	2.26	2.69
0.1	1.46	1.56	1.50	2.33	3.26	5.26	4.18	5.29	2.32	2.74
0.2	1.38	1.48	1.45	2.31	3.24	5.18	4.17	5.30	2.32	2.74
0.4	1.38	**1.46**	**1.38**	**2.18**	3.22	5.22	**4.00**	**5.14**	**2.25**	**2.66**
0.6	1.42	1.60	1.58	2.35	**3.19**	**5.16**	4.15	5.25	2.30	2.73

justifies our analysis of the existing mask-fixed model in Section III-A. At the same time, it confirms that the CAMD can effectively solve the interaction ambiguity problem, and the designed CAAM component can provide for long-range propagation between clicks.

To further explore the effect of AFL on the IIS model, we report the performance of the AdaptiveClick when only AFL is used. As seen from the TABLEs, the model's performance on GrabCut and the DAVIS dataset is already close to that of the full AdaptiveClick when only AFL is used. This indicates that the proposed AFL has strong robustness. With both CAMD and AFL, the model's performance greatly improved on all five test datasets, which indicates that the proposed CAMD and AFL can effectively address the problems of "interaction ambiguity" in existing IIS tasks.

b) Influence of components in AFL: In Table XI, FL does not perform satisfactorily without using any of the proposed components. In contrast, the model's performance improves clearly on all five test datasets after using the proposed ADA. This indicates that the proposed ADA can help the model adjust the learning strategy adaptively according to differences in sample difficulty distribution. At the same time, the first hypothesis proposed in Section III-C1, that giving a fixed γ to all training samples of a dataset is a suboptimal option that may prevent achieving satisfactory performance on low-confidence easy pixels, is verified, which proves the rationality and effectiveness of ADA.

Also, we explore the case of using only the AGR component. The use of AGR results in more significant performance gains on the Berkeley, SBD, and PasvalVOC test datasets compared to the ADA-only. This indicates the effectiveness of the proposed AGR and validates the second hypothesis proposed in Section III-C1, that FL has the problem that when one wants to increase the concentration on learning with severe hard-easy imbalance, it tends to sacrifice part of low-confidence easy pixels' loss contribution in the overall training process. At the same time, this proves the validity of the theoretical analysis in Section III-C1. Namely, the gradient direction of the simulated BCE can help AFL classify low-confidence easy pixels.

Finally, when both ADA and AGR are used, AFL brings a minimum gain of 0.14, 0.34 (on GrabCut), and a maximum gain of 0.94, 1.23 (on DAVIS and SBD) over FL on the NoC85 and NoC90, respectively. Such significant performance improvements again demonstrate the effectiveness of the AFL.

2) Influence of Hyper-Parameters: In Tables XII and XIII, the effects of λ_{cli}, λ_{afl}, λ_{dice}, **Q**, and λ_{mask} on the model performance are explored. Then, Tables XIV and XV report the results of different δ and α weights.

a) Hyper-parameter analysis of adaptiveclick: In Table XII, we validate the performance of our model on the five test datasets when λ_{cli} is set to 2 and 4, respectively. We observe that the model is not sensitive to the value of λ_{cli}, but the model demonstrates better results when λ_{cli} is 2. Consequently, in this study, we select a click weight value of $\lambda_{cli} = 2$. Furthermore, when comparing the performance for varying AFL weight values of $\lambda_{afl} = \{5, 4, 3\}$, we observe significant fluctuations in the model's performance as λ_{afl} decreases. As a result, we opt for an AFL weight of $\lambda_{afl} = 5$. Then, observing the performance change when the dice weight of λ_{dice} is taken as 5, 4, and 3, we can see that the model performs most consistently when λ_{dice} is 5. Further, we tested the effect of the NoC **Q** on the new performance of the model, and the number of **Q** chosen was 5, 10, 20,

TABLE XV
PERFORMANCE CHANGE BY USING AFL UNDER DIFFERENT WEIGHTS OF α TRAINED ON SBD [17] AND TESTED ON GRABCUT [58], BERKELEY [59], SBD [17], DAVIS [60], AND PASCAL VOC [61]

α	GrabCut		Berkeley		SBD		DAVIS		Pascal VOC	
	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90	NoC85	NoC90
0.0	1.44	1.56	1.53	**2.15**	3.43	5.47	**3.93**	5.28	2.32	2.76
0.5	1.46	1.60	1.43	2.30	3.29	5.29	3.99	5.27	**2.23**	**2.66**
1.0	**1.38**	**1.46**	**1.38**	2.18	**3.22**	**5.22**	4.00	**5.14**	2.25	**2.66**
1.5	**1.38**	1.48	1.61	2.53	3.26	5.30	4.11	5.45	2.26	2.67
2.0	1.40	1.48	1.57	2.48	3.29	5.32	4.06	5.38	2.30	2.76

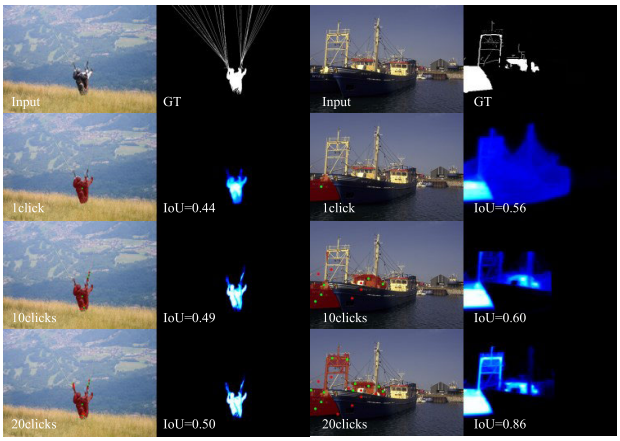


Fig. 8. Two illustrations of failure cases.

and 100, respectively. The experimental results show that the model has optimal performance when $\mathbf{Q} = 10$. Finally, the results of different mask loss weights, λ_{mask}, are listed in Table XIII. When $\lambda_{\text{mask}} = 1.0$, AdaptiveClick has the best performance. As a result, the values are set to $\lambda_{\text{mask}} = 1.0$, $\lambda_{\text{cli}} = 2.0$, $\lambda_{\text{afl}} = 5.0$, $\lambda_{\text{dice}} = 5.0$, and $\mathbf{Q} = 10.0$.

b) Hyper-parameter analysis of AFL: As shown in Tables XIV and XV, when δ is set to 0 and α is 0, AFL can be approximated as the value of NFL after applying ADA. Under these conditions, the NoC85 and NoC90 results of AFL are suboptimal. However, as δ increases from 0 to 0.2 and α rises from 0 to 0.5, there is a clear improvement in performance across all five datasets, providing compelling evidence for the efficacy of AFL. When $\delta = 0.4$ and $\alpha = 1.0$, the values of evaluation metrics start to fluctuate. Consequently, based on the outcomes of our experiments, the settings of $\delta = 0.4$ and $\alpha = 1.0$ emerge as the preferred parameters.

V. DISCUSSION

We introduce AdaptiveClick to effectively address the challenges associated with "interaction ambiguity," focusing on both inter-class click ambiguity resolution and intra-class click ambiguity optimization. Benefiting from CAMD's sensitivity to clicks and AFL's reduction of "gradient swamping," AdaptiveClick exhibits a more competitive performance on IIS across nine datasets. In addition to the IIS task, AdaptiveClick offers the following two advantages.

1) AdaptiveClick can offer a robust baseline for the IIS part from the generalized prompt segmentation models [29], [32], fostering the advancement of this task.
2) Whether referring [69], [70], interactive [27], [31], or prompt [29], [32] image/video segmentation tasks, "gradient swamping" is often present during the loss optimization process.

Our validation of AFL's effectiveness and generality for IIS tasks supports the notion that it can provide potential optimization guarantees in various fields.

However, AdaptiveClick has two limitations as follows.

1) As depicted in Fig. 8, AdaptiveClick may not be effective for slender objects with heavy occlusions, leading to potential segmentation failures.
2) Similar to other transformer-based IIS methods, our approach may not be efficient for low-power devices.

Consequently, enhancing the segmentation of slender objects with heavy occlusion through refined mask or pipeline optimization strategies and exploring lightweight transformer-based IIS frameworks through model compression and knowledge distillation represent two promising avenues. We defer these potential improvements to future work.

VI. CONCLUSION

In this article, we rethink the "interaction ambiguity" problem in the IIS task from the perspective of inter-class click ambiguity resolution to intra-class click ambiguity optimization. First, we designed CAMD to enhance the long-range interaction of clicks in the forward pass process and introduce a mask-adaptive strategy to find the optimal mask matching with clicks, thus solving the inter-class interaction ambiguity. Second, we observe that the root of the intra-class hard pixel misclassification is "gradient swamping" and propose a new AFL based on the gradient theory of BCE and FL to enforce the network to pay more attention to the ambiguous pixels, reducing the intra-class click ambiguity. Finally, experiments on nine datasets of the IIS task show that the proposed AdaptiveClick yields state-of-the-art performances.

REFERENCES

[1] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu, "MILCut: A sweeping line multiple instance learning paradigm for interactive image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 256–263.
[2] T. Wang, J. Yang, Z. Ji, and Q. Sun, "Probabilistic diffusion for interactive image segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 330–342, Jan. 2019.
[3] T. Wang, Z. Ji, J. Yang, Q. Sun, and P. Fu, "Global manifold learning for interactive image segmentation," *IEEE Trans. Multimedia*, vol. 23, pp. 3239–3249, 2021.
[4] X. Chen, Y. S. J. Cheung, S.-N. Lim, and H. Zhao, "ScribbleSeg: Scribble-based interactive image segmentation," 2303, *arXiv:2303.11320*.
[5] M. Jian and C. Jung, "Interactive image segmentation using adaptive constraint propagation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1301–1311, Mar. 2016.
[6] K. Sofiiuk, I. A. Petrov, and A. Konushin, "Reviving iterative training with mask guidance for interactive segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 3141–3145.
[7] Q. Liu et al., "PseudoClick: Interactive image segmentation with click imitation," in *Proc. ECCV*, 2022, pp. 728–745.
[8] Z. Lin, Z.-P. Duan, Z. Zhang, C.-L. Guo, and M.-M. Cheng, "FocusCut: Diving into a focus view in interactive segmentation," in *Proc. CVPR*, 2022, pp. 2637–2646.
[9] T. Wang, H. Li, Y. Zheng, and Q. Sun, "One-click-based perception for interactive image segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023, doi: 10.1109/TNNLS.2023.3274127.
[10] G. Wang et al., "DeepIGeoS: A deep interactive geodesic framework for medical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1559–1572, Jul. 2019.

[11] A. Diaz-Pinto et al., "DeepEdit: Deep editable learning for interactive segmentation of 3D medical images," in *Proc. MICCAI*, 2022, pp. 11–21.

[12] H. Ding, H. Zhang, C. Liu, and X. Jiang, "Deep interactive image matting with feature propagation," *IEEE Trans. Image Process.*, vol. 31, pp. 2421–2432, 2022.

[13] J. Deng and X. Xie, "3D interactive segmentation with semi-implicit representation and active learning," *IEEE Trans. Image Process.*, vol. 30, pp. 9402–9417, 2021.

[14] X. Gu, J. Li, K. Liu, Y. Zhu, X. Tao, and Y. Shang, "A precise minor-fault diagnosis method for lithium-ion batteries based on phase plane sample entropy," *IEEE Trans. Ind. Electron.*, pp. 1–9, 2024, doi: 10.1109/TIE.2023.3319717.

[15] Y. Li, G. Yang, Z. Su, S. Li, and Y. Wang, "Human activity recognition based on multienvironment sensor data," *Inf. Fusion*, vol. 91, pp. 47–63, Mar. 2023.

[16] Z. Lin, Z. Zhang, L.-Z. Chen, M.-M. Cheng, and S.-P. Lu, "Interactive image segmentation with first click attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13339–13348.

[17] S. Majumder and A. Yao, "Content-aware multi-level guidance for interactive instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11602–11611.

[18] X. Chen, Z. Zhao, F. Yu, Y. Zhang, and M. Duan, "Conditional diffusion for interactive segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7345–7354.

[19] Z. Li, Q. Chen, and V. Koltun, "Interactive image segmentation with latent diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 577–585.

[20] J. H. Liew, S. Cohen, B. Price, L. Mai, S. Ong, and J. Feng, "MultiSeg: Semantically meaningful, scale-diverse segmentations from minimal user input," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 662–670.

[21] X. Chen, Z. Zhao, Y. Zhang, M. Duan, D. Qi, and H. Zhao, "FocalClick: Towards practical interactive image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1290–1299.

[22] Q. Liu, Z. Xu, Y. Jiao, and M. Niethammer, "iSegFormer: Interactive segmentation via transformers with application to 3D knee MR images," in *Proc. MICCAI*, 2022, pp. 464–474.

[23] W.-D. Jang and C.-S. Kim, "Interactive image segmentation via backpropagating refinement scheme," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5297–5306.

[24] K. Sofiiuk, I. Petrov, O. Barinova, and A. Konushin, "F-BRS: Rethinking backpropagating refinement for interactive segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8623–8632.

[25] J. Lin et al., "Click-pixel cognition fusion network with balanced cut for interactive image segmentation," *IEEE Trans. Image Process.*, vol. 33, pp. 177–190, 2024.

[26] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 373–381.

[27] Q. Liu, Z. Xu, G. Bertasius, and M. Niethammer, "SimpleClick: Interactive image segmentation with simple vision transformers," in *Proc. ICCV*, 2023, pp. 22290–22300.

[28] Q. Wei, H. Zhang, and J.-H. Yong, "Focused and collaborative feedback integration for interactive image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 18643–18652.

[29] A. Kirillov et al., "Segment anything," in *Proc. ICCV*, 2023, pp. 4015–4026.

[30] Y. Huang et al., "InterFormer real-time interactive image segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 22301–22311.

[31] A. K. Rana, S. Mahadevan, A. Hermans, and B. Leibe, "DynaMITe: Dynamic query bootstrapping for multi-object interactive segmentation transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1043–1052.

[32] X. Zou et al., "Segment everything everywhere all at once," 2304, *arXiv:2304.06718*.

[33] J. Liew, Y. Wei, W. Xiong, S. Ong, and J. Feng, "Regional interactive image segmentation networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2746–2754.

[34] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11692–11701.

[35] M. Forte, B. Price, S. Cohen, N. Xu, and F. Pitié, "Getting to 99% accuracy in interactive segmentation," 2003, *arXiv:2003.07932*.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[37] T. Kontogianni, M. Gygli, J. Uijlings, and V. Ferrari, "Continuous adaptation for interactive object segmentation by learning from corrections," in *Proc. ECCV*, 2020, pp. 579–596.

[38] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1290–1299.

[39] Y.-D. Ma, Q. Liu, and Z.-B. Quan, "Automated image segmentation using improved PCNN model based on cross-entropy," in *Proc. Int. Symp. Intell. Multimedia, Video Speech Process.*, Oct. 2004, pp. 743–746.

[40] Z. Leng et al., "PolyLoss: A polynomial expansion perspective of classification loss functions," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[41] V. Pihur, S. Datta, and S. Datta, "Weighted rank aggregation of cluster validation measures: A Monte Carlo cross-entropy approach," *Bioinformatics*, vol. 23, no. 13, pp. 1607–1615, Jul. 2007.

[42] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[43] B. Li et al., "Equalized focal loss for dense long-tailed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6990–6999.

[44] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 616–625.

[45] Y. Song, J. Y. Teoh, K.-S. Choi, and J. Qin, "Dynamic loss weighting for multiorgan segmentation in medical images," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2023, doi: 10.1109/TNNLS.2023.3243241.

[46] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *Proc. Int. Symp. Vis. Comput.*, 2016, pp. 234–244.

[47] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

[48] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, Oct. 2020, pp. 1–7.

[49] K. Sofiiuk, O. Barinova, and A. Konushin, "AdaptIS: Adaptive instance selection network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7355–7363.

[50] Z. Ding, T. Wang, Q. Sun, and F. Chen, "Rethinking click embedding for deep interactive image segmentation," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 261–273, Jan. 2023.

[51] M. Zhou et al., "Interactive segmentation as Gaussian process classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19488–19497.

[52] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[53] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "DD deformable transformers for end-to-end object detection," in *Proc. ICLR*, 2021, pp. 3–7.

[54] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. NIPS*, Dec. 2021, pp. 17864–17875.

[55] W. Rudin et al., *Principles of Mathematical Analysis*, vol. 3. New York, NY, USA: McGraw-Hill, 1976.

[56] H. P. Heinig and L. Maligranda, "Chebyshev inequality in function spaces," *Real Anal. Exchange*, vol. 17, no. 1, p. 211, 1991.

[57] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[58] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut' interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[59] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognit.*, vol. 43, no. 2, pp. 434–444, Feb. 2010.

[60] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

[61] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, Jun. 2010.

[62] S. Gerhard, J. Funke, J. Martel, A. Cardona, and R. Fetter, "Segmented anisotropic ssTEM dataset of neural tissue," *Figshare*, Nov. 2013, doi: 10.6084/m9.figshare.856713.v1.
[63] U. Baid et al., "The RSNA-ASNR-MICCAI BraTS 2021 benchmark on brain tumor segmentation and radiogenomic classification," 2021, *arXiv:2107.02314*.
[64] F. Ambellan, A. Tack, M. Ehlke, and S. Zachow, "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative," *Med. Image Anal.*, vol. 52, pp. 109–118, Feb. 2019.
[65] F. Du, J. Yuan, Z. Wang, and F. Wang, "Efficient mask correction for click-based interactive image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22773–22782.
[66] H. Zeng, W. Wang, X. Tao, Z. Xiong, Y.-W. Tai, and W. Pei, "Feature decoupling-recycling network for fast interactive segmentation," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 6665–6675.
[67] K. Li, G. Vosselman, and M. Y. Yang, "Interactive image segmentation with cross-modality vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 762–772.
[68] C. Fang, Z. Zhou, J. Chen, H. Su, Q. Wu, and G. Li, "Variance-insensitive and target-preserving mask refinement for interactive image segmentation," 2023, *arXiv:2312.14387*.
[69] G. Feng, Z. Hu, L. Zhang, J. Sun, and H. Lu, "Bidirectional relationship inferring network for referring image localization and segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2246–2258, May 2023.
[70] J. Chen et al., "EPCFormer: Expression prompt collaboration transformer for universal referring video object segmentation," 2023, *arXiv:2308.04162*.

Jiacheng Lin received the B.S. degree from Lanzhou University of Technology (LUT), Lanzhou, China, in 2018, and the M.S. degree from Guizhou University (GZU), Guiyang, China, in 2021. He is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University (HNU), Changsha, China.

His research interests include computer vision, sense understanding, and privacy protection.

Jiajun Chen received the B.S. degree from Guangdong University of Technology (GDUT), Guangzhou, China, in 2022. He is currently pursuing the M.S. degree with the School of Robotics, Hunan University (HNU), Changsha, China.

His research interests include computer vision, machine learning, and image segmentation.

Kailun Yang received the B.S. degree in measurement technology and instrument from Beijing Institute of Technology (BIT), Beijing, China, the B.S. degree in economics from Peking University (PKU), Beijing, in 2014, and the Ph.D. degree in information sensing and instrumentation from the State Key Laboratory of Extreme Photonics and Instrumentation, Zhejiang University (ZJU), Hangzhou, China, in 2019.

He performed the Ph.D. internship with the Robotics and eSafety (RobeSafe) Research Group, University of Alcalá (UAH), Alcalá de Henares, Spain, from 2017 to 2018. He was a Post-Doctoral Researcher with the Computer Vision for Human-Computer Interaction (CV:HCI) Laboratory, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, from 2019 to 2023. He is currently a Professor with the School of Robotics and the National Engineering Research Center of Robot Visual Perception and Control Technology, Hunan University (HNU), Changsha, China. For more information, visit his website: https://yangkailun.com.

Alina Roitberg (Member, IEEE) received the B.Sc. and M.Sc. degrees (Hons.) in computer science from the Technical University of Munich (TUM), Munich, Germany, in 2015, and the Ph.D. degree in deep learning for driver observation from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, in 2021.

After working as a Data Science Consultant in automotive sector from 2016 to 2017, she joined Karlsruhe Institute of Technology (KIT), for Ph.D. studies. From 2021 to 2023, she was a Post-Doctoral Researcher with the Computer Vision for Human-Computer Interaction Laboratory, KIT. She did a research visit with the Intelligent Transport Systems Team, Johannes Kepler University Linz (JKU), Linz, Austria, in 2023. She is currently an Assistant Professor with the University of Stuttgart, Stuttgart, Germany. Her research interests include human activity recognition, uncertainty-aware deep learning, open-set, zero- and few-shot recognition, and applications in intelligent vehicles.

Dr. Roitberg received the Best Student Paper Runner-Up Award at IV 2020 and completed the Ph.D. internship at Facebook.

Siyu Li received the B.S. degree from Nanjing Normal University (NNU), Nanjing, China, in 2019, and the M.S. degree from Central South University (CSU), Changsha, China, in 2022. She is currently pursuing the Ph.D. degree with the School of Robotics, Hunan University (HNU), Changsha.

Her research interests include computer vision and BEV semantic mapping.

Zhiyong Li (Member, IEEE) received the M.Sc. degree in system engineering from the National University of Defense Technology (NUTD), Changsha, China, in 1996, and the Ph.D. degree in control theory and control engineering from Hunan University (HNU), Changsha, in 2004.

In 2004, he joined the College of Computer Science and Electronic Engineering, Hunan University, where he is currently a Full Professor. He has published more than 100 papers in international journals and conferences. His research interests include intelligent perception and autonomous mobile agents, machine learning and industrial big data, and intelligent optimization algorithms with applications.

Dr. Li is a member of the China Computer Federation (CCF) and the Chinese Association for Artificial Intelligence (CAAI).

Shutao Li (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees from Hunan University, Changsha, China, in 1995, 1997, and 2001, respectively.

In 2001, he joined the College of Electrical and Information Engineering, Hunan University, where he is currently a Full Professor. From May 2001 to October 2001, he was a Research Associate with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, where he was a Visiting Professor from April 2005 to June 2005. From November 2002 to November 2003, he was a Post-Doctoral Fellow with the Royal Holloway College, University of London, London, U.K. He has authored or coauthored more than 200 refereed articles. His current research interests include image processing, pattern recognition, and artificial intelligence.

Dr. Li is a member of the Editorial Board of *Information Fusion* and *Sensing and Imaging*. He received the two Second-Grade State Scientific and Technological Progress Awards of China in 2004 and 2006. He is an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.