Interactive Image Segmentation with Temporal Information Augmented

Qiaoqiao Wei
wqq18@mails.tsinghua.edu.cn
Hui Zhang
huizhang@tsinghua.edu.cn
Jun-Hai Yong
yongjh@tsinghua.edu.cn

School of Software
BNRist
Tsinghua University
Beijing, China

Abstract

Interactive image segmentation aims to achieve pixel-wise localization of an object of interest in an image using minimal user annotations. Despite advances, existing methods suffer from accuracy fluctuations and notable constrained minimal errors due to annotation sparsity and neural network limitations. To improve segmentation quality and stability, this paper proposes the Temporal Information Augmentation (TIA) method. Informed by the concept of the proportional-integral-derivative (PID) controller, TIA integrates contextual information from multiple interaction rounds to enhance feature representations. Specifically, TIA strengthens the response of current feedback information through cosine feature similarities, fuses foreground and background instructive information from past interaction rounds with current features, and refines features in potential wrongly segmented areas by perceiving changes in the segmentation results. By incorporating current, past, and future contextual cues, TIA improves the discrimination ability of the segmentation model for target objects. Experimental results on the GrabCut, Berkeley, SBD, and DAVIS datasets with SegFormer- and ViT-based backbones have demonstrated state-of-the-art performance, highlighting generalization capability, efficiency, and effectiveness of TIA.

1 Introduction

Interactive image segmentation is a technique that obtains a segmentation mask of a target object in an input image based on simple user-provided cues. It allows users to engage in multiple rounds of interaction, where they can provide new annotations on the wrongly segmented regions and receive a new segmentation mask. The objective of interactive image segmentation is to achieve precise segmentation results with minimal labeling. Because of the manipulability and flexibility, interactive image segmentation is a crucial technique directly used in applications such as photo editing [13] and medical imaging diagnosis [34]. Besides, it enables artificial-intelligence-assisted data labeling [28] and therefore becomes an auxiliary tool for many challenging segmentation-related tasks, such as scene parsing and autonomous driving [1].

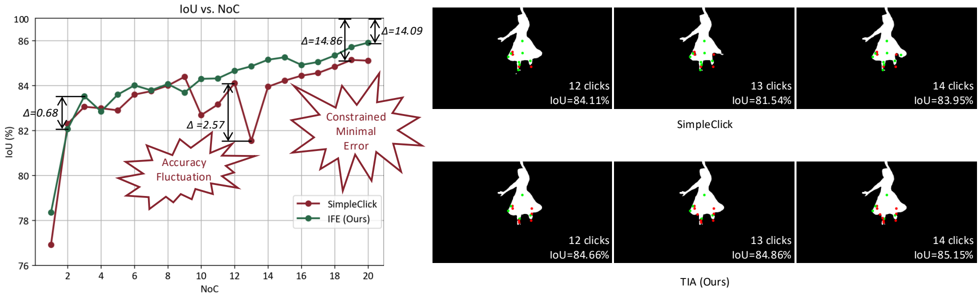

Figure 1: Two primary problems encountered by existing methods (taking SimpleClick [21] as an example), which are the accuracy fluctuation and the constrained minimal error.

We reckon that an interactive image segmentation model resembles a closed-loop control system in the field of automatic control. A closed-loop control system regulates a variable based on the error signal fed back [2]. Concretely, in a control loop, the difference between the target setpoint and the measured value of the variable is calculated and fed back to the system, and then the system adjusts the control signal to minimize the difference. An interactive image segmentation system operates similarly. It corrects a segmentation result according to user feedback. Since the feedback information is a bridge connecting adjacent interaction rounds, it is natural to exploit it to facilitate the segmentation process.

There are two types of feedback used in an interactive image segmentation system: user-annotated clicks and the segmentation mask generated in the last interaction round. Clicks are usually placed on the inaccurate segmentation areas of a predicted segmentation mask and thus help the segmentation network quickly locate those areas. To utilize feedback clicks, previous methods [14, 17, 32, 38] encoded all clicks into interaction maps using Gaussian filtering [38] or disk encoding [33], and jointly fed the input image and the maps into the segmentation network. The feedback segmentation result provides pixel-level semantic prior information about the foreground object, which helps the segmentation network recognize the position and shape of the foreground object. Starting from RITM [33], subsequent research [4, 12, 20, 21, 35] fed back the segmentation mask generated in the last interaction round as an additional channel of input.

However, previous work has primarily encountered two main issues, as depicted in Figure 1. Firstly, there is a potential for a significant decrease in segmentation accuracy when a new click is added, leading to a large fluctuation in the accuracy curve as the number of annotated clicks increases, referred to as **accuracy fluctuation**. This is akin to the overshoot problem encountered in dynamic system controls [24]. Secondly, there is disparity between the accuracy obtained under a restricted number of annotated clicks after segmentation reaches a steady state and the 100% accuracy, referred to as **constrained minimal error**. This problem is analogous to the concept of steady-state error in control theory [24]. Possible reasons for these two issues are as follows. Firstly, both types of feedback information have their limitations. Annotated clicks, although accurate, are scarce and provides partial guidance. The feedback segmentation results are dense but noisy. Secondly, existing methods establish long-range dependencies through stacked convolutional layers and non-local attention without imposing additional constraints related to semantic information, potentially causing the new clicks to affect the predicted labels of pixels that are spatially distant [35]. Thirdly, previous methods focused on manipulating information flow in the spatial perspective of the

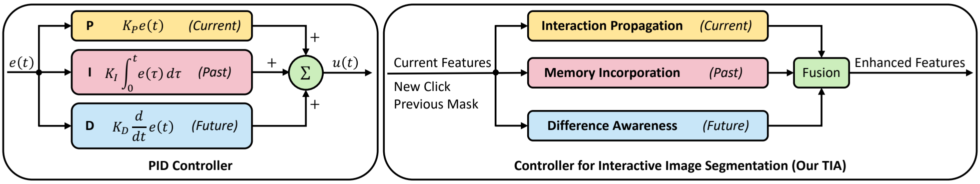
$e(t)$
P $K_P e(t)$ (Current)
I $K_I \int_0^t e(\tau)\, d\tau$ (Past)
D $K_D \frac{d}{dt} e(t)$ (Future)
+
+
+
Σ
$u(t)$
PID Controller
Current Features
New Click
Previous Mask
Interaction Propagation (Current)
Memory Incorporation (Past)
Difference Awareness (Future)
Fusion
Enhanced Features
Controller for Interactive Image Segmentation (Our TIA)

Figure 2: A comparison between a common PID controller and the proposed controller.

current state while neglecting instructive temporal information from historical states.

To tackle the aforementioned challenges, we introduce the idea of automatic control into the realm of interactive image segmentation. A typical closed-loop controller structure is the proportional-integral-derivative (PID) controller [23] (Figure 2). It utilizes three adjustment parameters—the proportional, the integral, and the derivative terms—to determine the next control action based on the current, past, and anticipated future errors, which accelerates the process of reaching the desired setpoint and dampen the overshoot issue.

Inspired by the PID control, we propose Temporal Information Augmentation (TIA) method, which incorporates temporal feedback information from multiple interaction rounds to fit the ideal segmentation result with fewer clicks. The proposed method is composed of three branches: Interaction Propagation (IP), Memory Incorporation (MI), and Difference Awareness (DA), which respectively correspond to the three terms P, I, and D in a PID controller. The IP branch produces a response that is proportional to the label of a new click and utilizes this response to propagate the correction intention, derived from the new click, across the entire feature map. The MI branch maintains a memory bank that stores past error signals—i.e., feedback information from previous interaction rounds—and consolidates this history information into the current state, which mitigates the issue of accuracy fluctuation. Meanwhile, the DA branch models the difference between the previous segmentation mask and the newly predicted one, thus enabling the segmentation network to focus on potential erroneous areas and improving segmentation results. By integrating present information processing, history knowledge integration, and future trend prediction,the proposed method can achieve high segmentation accuracy with fewer annotations.

2 Related Work

2.1 Interaction Feedback Exploitation

Originally, the feedback information in each interaction round comprised only newly annotated clicks [17, 18?]. Sofiiuk *et al.* [33] introduced the generated segmentation mask from the last interaction round as additional feedback. To harness interaction feedback, certain methods [10, 35] have investigated the appropriate feedback fusion location, such as early fusion and late fusion, while others [7, 15] have focused on developing efficient or lightweight feedback processing structures. Nonetheless, prior interactive image segmentation methods only produced correction signals from the current state in each interaction round, thus resembling a proportional controller, and these methods easily led to the accuracy fluctuation issue and posed significant constrained minimal errors.

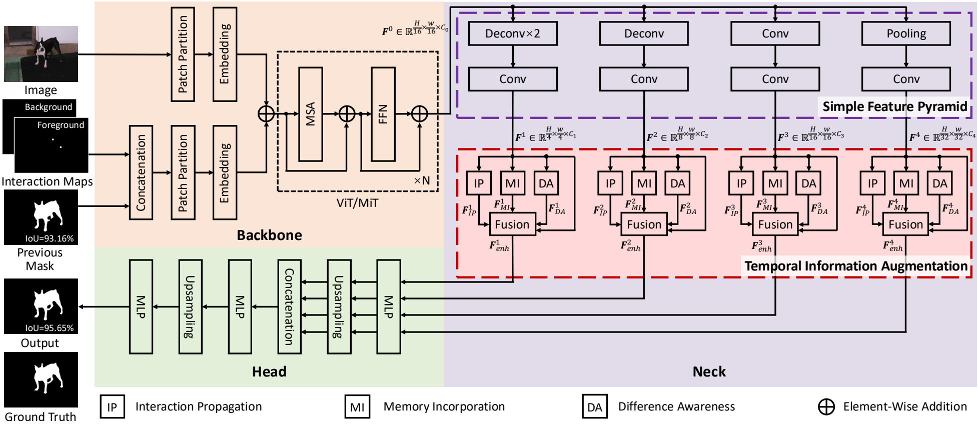

Figure 3: The overall network structure of TIA. The network basically comprises of three parts. The backbone extracts features from input images and feedback. The neck constructs a feature pyramid and incorporates temporal information. The head fuses multi-level features and outputs the segmentation mask.

2.2 PID Control in Computer Vision

PID controllers have been applied to various computer vision tasks. In object tracking, a PID controller was employed to manage the position and the orientation of a camera, enabling it to follow moving objects at each time step [25, 31]. More recently, PIDNet [37] was proposed to achieve real-time semantic segmentation. In the PIDNet, the P branch parses and retains details, the I branch integrates long-range context dependencies, and the D branch extracts high-frequency information from feature maps. Notably, as there is no feedback generated in the semantic segmentation task, all three branches of PIDNet exclusively operate in the spatial domain of feature maps, without involving any concepts about the past and the future. In contrast, our method aggregates present, past, and future information to boost segmentation performance.

3 Method

3.1 Preliminaries

At a time step t, a PID controller calculates an error value $e(t)$ and outputs a control variable $u(t)$ based on the proportional (P), integral (I), and derivative (D) terms. Mathematically, $u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{d}{dt}e(t)$, where K_P, K_I, and K_D denote the scaling coefficients for the P, I, and D terms respectively. The term P produces an instantaneous response proportional to the current error value $e(t)$. The term I fully accounts for the magnitude and the duration of errors and mitigates the steady-state error. The term D estimates the future trend of the error based on the current rate of change of $e(t)$, making the controller respond fast to a sudden change of the error and thus reducing the overshoot issue. Combining the three terms, a PID controller can apply accurate and responsive correction on many industrial control systems.

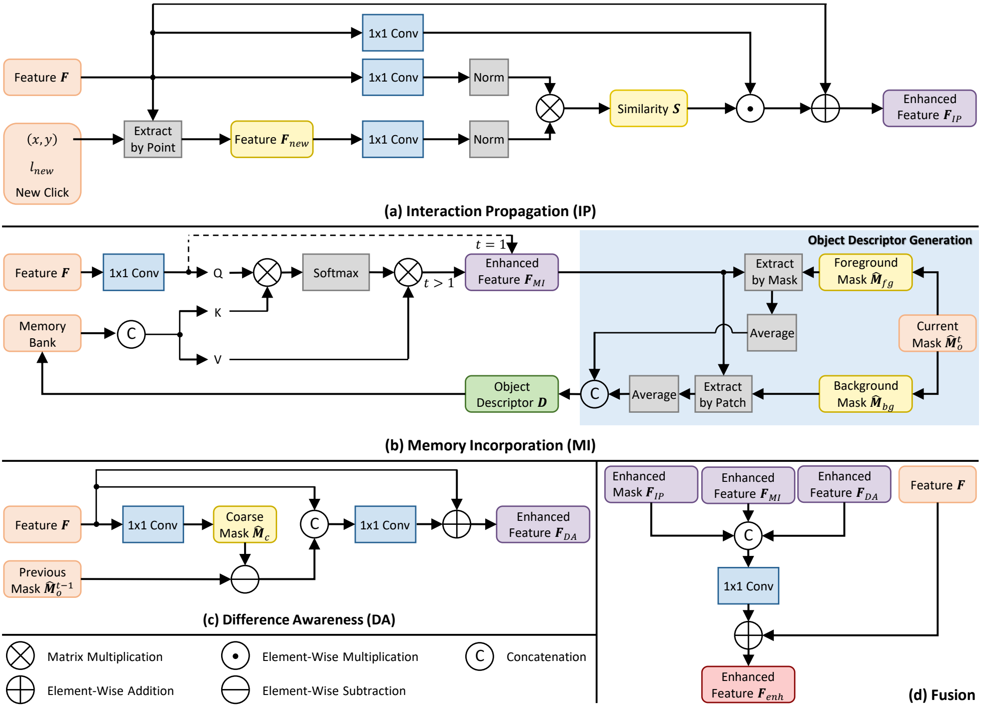
Feature $\boldsymbol{F}$
(x,y)
l_{new}
New Click
Extract by Point
Feature $\boldsymbol{F}_{new}$
1x1 Conv
1x1 Conv
1x1 Conv
Norm
Norm
Similarity $\boldsymbol{S}$
Enhanced Feature $\boldsymbol{F}_{IP}$
(a) Interaction Propagation (IP)
Feature $\boldsymbol{F}$
1x1 Conv
Q
K
V
Memory Bank
C
Softmax
$t = 1$
$t > 1$
Enhanced Feature $\boldsymbol{F}_{MI}$
Object Descriptor Generation
Extract by Mask
Foreground Mask $\widehat{\boldsymbol{M}}_{fg}$
Average
Current Mask $\widehat{\boldsymbol{M}}_o^t$
Background Mask $\widehat{\boldsymbol{M}}_{bg}$
Extract by Patch
Average
C
Object Descriptor $\boldsymbol{D}$
(b) Memory Incorporation (MI)
Feature $\boldsymbol{F}$
1x1 Conv
Coarse Mask $\widehat{\boldsymbol{M}}_c$
Previous Mask $\widehat{\boldsymbol{M}}_o^{t-1}$
C
1x1 Conv
Enhanced Feature $\boldsymbol{F}_{DA}$
(c) Difference Awareness (DA)
Matrix Multiplication
Element-Wise Multiplication
C Concatenation
Element-Wise Addition
Element-Wise Subtraction
Enhanced Mask $\boldsymbol{F}_{IP}$
Enhanced Feature $\boldsymbol{F}_{MI}$
Enhanced Feature $\boldsymbol{F}_{DA}$
Feature $\boldsymbol{F}$
C
1x1 Conv
Enhanced Feature $\boldsymbol{F}_{enh}$
(d) Fusion

Figure 4: The components of the proposed Temporal Information Augmentation, which are the Interaction Propagation module, the Memory Incorporation module, the Difference Awareness module, and the fusion module.

3.2 Overview

The overview of the proposed architecture is depicted in Figure 3, which is primarily divided into three parts: backbone, neck, and head. Disk encoding [33] is employed to encode user-provided clicks into interaction maps. In the backbone, the input image $I \in \mathbb{R}^{H\times W\times 3}$ and the interaction feedback are separately transformed into a sequence of tokens, added together, and passed into a backbone to produce a feature map $\boldsymbol{F}^0$. The neck consists of a simple feature pyramid (SFP) [16] and the proposed TIA. The former extracts multi-scale features at $\{1/4, 1/8, 1/16, 1/32\}$ of the original image resolution across four stages. The latter enhances the features by incorporating both past and present feedback information and activating the segmentation difference between two adjacent interaction rounds, as described in Section 3.3.1-3.3.3. Lastly, the head, following SimpleClick [21], comprises multilayer perceptrons (MLPs), upsampling, and concatenation, which is lightweight and effective.

3.3 Temporal Information Augmentation (TIA)

3.3.1 Interaction Propagation (IP)

In click-based interactive image segmentation, user annotations are sparse relative to the number of pixels in the entire image. Consequently, a high-performing network needs to be able to infer global segmentation correction intentions from local hints. Additionally, existing methods [12, 20, 21] typically encode user-annotated clicks without considering their sequences, which may weaken the instruction of the new click. To enhance this capability, we propose the Interaction Propagation (IP) to propagate correction information provided by a new click from local to global scope.

The IP branch is shown in Figure 4(a). It enhances the features of potential pixels that may belong to the same category as the new annotated clicks. To identify these potential pixels, the feature similarities between all pixels and a new click are calculated as follows:

$$\boldsymbol{S} = ||\phi_1(\boldsymbol{F})||_2 \otimes ||\phi_2(\boldsymbol{F}_{new})||_2, \tag{1}$$

where ϕ_1 and ϕ_2 are two 1×1 convolutions, and $||\cdot||_2$ denotes 2-norm. This calculation is inspired by FCFI [35]. Then, we enhance the features in the potential positions in a soft way:

$$\boldsymbol{F}_{IP} = \phi_3(\boldsymbol{F}) \odot \boldsymbol{S} + \boldsymbol{F}, \tag{2}$$

where ϕ_3 is 1×1 convolution. To effectively identify the correct potential pixels based on this feature similarity, we construct an estimated segmentation mask $\hat{\boldsymbol{M}}_{IP} = l_{new}\boldsymbol{S} + (1 - l_{new})(1 - \boldsymbol{S})$ using the similarity matrix $\boldsymbol{S}$ and the label of the new click $l_{new} \in \{0, 1\}$. $\hat{\boldsymbol{M}}_{IP}$ is supervised by the mean square error (MSE) loss (Section 4.1), which also helps enhance the features generated by the SFP.

3.3.2 Memory Incorporation (MI)

For interactive image segmentation, steady-state error might be attributed to inherent model defects or information loss. To reduce the error, we propose the Memory Incorporation (MI), depicted in Figure 4(b). The MI branch abstracts historical features into object descriptors, which encode high-level information about an object instance and scene context, and interact them with current features, thereby enhancing the distinctiveness of foreground and background within the features.

Object descriptors, denoted as $\boldsymbol{D} \in \mathbb{R}^{(1+K)\times C_D}$, integrate C_D-dimension feature vectors for the target object and K (a perfect square) background patches. The generation process is illustrated in the blue block in Figure 4(b). At each interaction round, the final segmentation mask $\hat{\boldsymbol{M}}_o^t$ is divided into a foreground mask $\hat{\boldsymbol{M}}_{fg}$ and a background mask $\hat{\boldsymbol{M}}_{bg}$. In the foreground/background mask, pixels belonging to foreground/background are set to 1, while others are set to 0. The background mask is evenly divided into $\sqrt{K} \times \sqrt{K}$ patches. Each object descriptor is obtained by averaging the features of pixels that are assigned to 1 in the corresponding mask or patch. Take the foreground object descriptor as an example:

$$\boldsymbol{D}_{fg} = \sum(F \odot \hat{\boldsymbol{M}}_{fg})/(\sum \hat{\boldsymbol{M}}_{fg} + \varepsilon), \tag{3}$$

where ε is a small constant to avoid division by zero. A memory bank with a certain capacity is maintained to store the object descriptors from the past few interaction rounds.

The generated object descriptors are then employed to activate the current features $\boldsymbol{F}$ through attention. $\boldsymbol{F}$ is encoded by 1×1 convolution first. In the first interaction round, where the memory bank is empty, the object features are directly used to generate new object descriptors. In the subsequent interaction rounds, all the object descriptors in the memory bank are concatenated in depth. Then, cross attention is applied with the object features as *query* and the previous object descriptors as *key* and *value*:

$$\boldsymbol{F}_{MI} = \text{Softmax}(\frac{\boldsymbol{F}\boldsymbol{D}^T}{\sqrt{C_d}})\boldsymbol{D}. \tag{4}$$

The output $\boldsymbol{F}_{MI}$ is used to generate new object descriptors.

3.3.3 Difference Awareness (DA)

To improve segmentation quality with a reduced number of clicks, we introduce the Difference Awareness (DA) technique. Inspired by the PID controller, which anticipates future trends to alleviate overshoot, DA predicts the disparities between the current segmentation mask and that of the preceding interaction round. These disparities serve as heuristic cues to highlight potential segmentation errors. This allows the segmentation network to focus on the areas that needing correction and refines the features in those areas. Additionally, the "prediction-refinement" process enhances the network's self-perception ability regarding the segmentation trends and mitigates fluctuations in accuracy.

As shown in Figure 4(c), a series of 1×1 convolution blocks, denoted as ϕ_4, is utilized to predict a coarse segmentation mask $\hat{\boldsymbol{M}}_c$ from the features $\boldsymbol{F}$. Subsequently, the element-wise difference between $\hat{\boldsymbol{M}}_c$ and the segmentation mask generated in the last interaction round $\hat{\boldsymbol{M}}_o^{t-1}$ is calculated. The difference helps the segmentation network focus on wrongly segmented areas in the last interaction round and corrects the current feature. This process can be formulated as

$$\boldsymbol{F}_{DA} = \phi_5(\text{concat}(\phi_4(\boldsymbol{F}) - \hat{\boldsymbol{M}}_o^{t-1}, \boldsymbol{F})) + \boldsymbol{F}, \tag{5}$$

where ϕ_4 and ϕ_5 are convolution blocks, and "concat(·,·)" represents concatenation in depth.

3.3.4 Fusion

The fusion module (Fig. 4(d)) receives outputs from the IP, MI, and DA modules as inputs to enhance the features $\boldsymbol{F}$. Specifically, the inputs are concatenated along the channel dimension, followed by a convolutional block ϕ_6 that autonomously selects important information from the three inputs to generate correction signals. These correction signals are then fused into the feature $\boldsymbol{F}$ through a residual connection. This process can be represented as:

$$\boldsymbol{F}_{enh} = \phi_6(\text{concat}(\boldsymbol{F}_{IP}, \boldsymbol{F}_{MI}, \boldsymbol{F}_{DA})) + \boldsymbol{F}. \tag{6}$$

3.4 Loss

To supervise the training of the segmentation network, two kinds of losses were utilized: MSE loss and the normalized focal loss (NFL) [33], denoted as $\mathcal{L}_{nfl}$. The ground truth mask is denoted as $\boldsymbol{M}$. The total loss function is

$$\mathcal{L} = 0.25 \sum_{i=1}^{4} \mathcal{L}_{mse}(\hat{\boldsymbol{M}}_{IP}^{i}, \boldsymbol{M}) + 0.25 \sum_{i=1}^{4} \mathcal{L}_{nfl}(\hat{\boldsymbol{M}}_{c}^{i}, \boldsymbol{M}) + \mathcal{L}_{nfl}(\hat{\boldsymbol{M}}_{o}^{t}, \boldsymbol{M}), \tag{7}$$

where the superscript i denotes the index of stage.

4 Experiments

4.1 Experimental Setup

Dataset. We adopted four public benchmarks to evaluate the proposed method: 1) **GrabCut** [29]: 50 images each with a corresponding instance mask; 2) **Berkeley** [22]: 96 images and 100 instance masks in total; 3) **SBD** [11]: 8,498 images for training and 2,820 images in the validation set, including 6,671 instance-level masks; 4) **DAVIS** [27]: 345 frames sampled from 50 high-quality video sequences.

Implementation Details. During training, the following data augmentation techniques were adopted: random resizing with a scale ranging from 0.5 to 2.0, random flipping, random cropping, and random jittering of brightness, contrast, and RGB values. The size of an input image was 448 × 448. Plain ViT [6] and SegFormer [36] were employed as the backbone separately. The backbone was pretrained on ImageNet-1K [30]. The segmentation network was trained for 55 epochs using two NVIDIA GeForce RTX 3090 GPUs. An Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ was applied. The learning rate was initialized as 5×10^{-5} and multiplied by 0.1 after 50 epochs. For the hyper-parameters, we set $C_D = 256$, $K = 16$, and $\varepsilon = 10^{-6}$. TIA was implemented in PyThon and PyTorch [26].

Click Simulation. During each iteration of the training process, a random number of user interaction and segmentation rounds will be simulated, with a maximum of 24 rounds. Following InterFormer [12], a sampling approach with probability exponentially decaying with the number of rounds is used to select the number of simulated rounds. In both the training and inference stages, a single new click was utilized in each interaction round targeting the incorrectly segmented area. Following the standard protocol [21, 32, 33], this new click was positioned at the center of the largest false negative or positive connected component detected in the previous segmentation mask. The maximum number of simulated rounds was limited to 20 for each image during inference.

Evaluation Metrics. Our approach is accessed using eight evaluation metrics, which are classified into five categories. The specifics of these metrics are described as follows. (1) **Overall quality.** ①IoU@N [38]: the mean intersection over union (IoU) achieved for all images when provided N clicks; ②NoC@α [38]: the mean number of clicks (NoC) to reach a specific IoU threshold α for all images; (2) **Boundary Quality.** ③BIoU@N [20]: the mean boundary IoU (BIoU) [5] achieved for all images when provided N clicks; ④ASSD@N [20]: the average symmetric surface distance (ASSD) achieved when provided N clicks; (3) **Efficiency.** ⑤SPC [32]: the average running time for all images in seconds per click. To better validate the efficacy of our approach in enhancing the accuracy and stability of the interactive segmentation process, we propose the following metrics: (4) **Stability.** ⑥NoDC$_N$: the number of degeneration cases (NoDC) with N clicks, which are cases that obtain lower IoU after the addition of a new click; ⑦mDIoU: the mean decrease in IoU (mDIoU) of examples where the addition of a new click leads to a decrease in IoU; (5) **Performance Ceiling.** ⑧CME: the mean constrained minimal error (CME) for all images, which is the difference between the peak accuracy obtained under the constraint of a limited number of clicks and 100% accuracy.

4.2 Comparison with Previous Work

The quantitative segmentation results, segmented by different backbones, are tabulated in Table 1 and Table 2. The best and the second-best results for different backbones are written in bold and underlined, respectively. The experimental results indicate that our method:

Method	Backbone	GrabCut		Berkeley		SBD		DAVIS	
		NoC@85%	NoC@90%	NoC@85%	NoC@90%	NoC@85%	NoC@90%	NoC@85%	NoC@90%
†f-BRS-B [32]	ResNet-101	2.30	2.72	2.44	4.57	4.81	7.73	5.04	7.41
†CDNet [3]	ResNet-101	2.42	2.76	2.08	3.65	4.73	7.66	5.33	6.97
†FocusCut [20]	ResNet-101	1.46	1.64	1.81	3.01	3.40	5.31	4.85	6.22
†FCFI [35]	ResNet-101	1.64	1.80	1.56	2.84	3.26	5.35	4.75	6.48
‡RITM [33]	HRNet-18s	1.54	1.68	1.69	2.60	4.04	6.48	4.70	5.98
‡FocalClick [4]	HRNet-18s	1.48	1.62	1.72	2.66	4.43	6.79	3.90	5.25
‡FCFI [35]	HRNet-18s	1.50	1.56	1.54	2.05	3.88	6.24	3.70	5.16
‡RITM [33]	HRNet-18	1.42	1.54	1.47	2.26	3.80	6.06	4.36	5.74
‡FCFI [35]	HRNet-18	1.38	1.46	1.41	1.96	3.63	5.83	3.97	5.16
†GPCIS [39]	SegFormer-B0	1.60	1.76	1.84	2.70	4.16	6.28	4.45	6.04
‡FocalClick [4]	SegFormer-B0	1.40	1.66	1.59	2.27	4.56	6.86	4.04	5.49
‡VTMR [8]	SegFormer-B0	1.42	1.54	1.64	2.18	4.43	6.75	3.81	5.39
‡**TIA (Ours)**	SegFormer-B0	**1.38**	**1.42**	**1.40**	**1.97**	**4.12**	**6.35**	**3.68**	**5.14**
‡FocalClick [4]	SegFormer-B3	1.44	1.50	1.55	1.92	3.53	5.59	3.61	4.90
‡VTMR [8]	SegFormer-B3	1.38	1.42	1.44	1.72	3.55	5.53	3.26	4.82
‡**TIA (Ours)**	SegFormer-B3	**1.36**	**1.38**	**1.39**	**1.44**	**3.46**	**5.32**	**3.19**	**4.57**
*SAM [15]	ViT-B	1.56	1.68	**1.35**	1.91	6.53	10.38	4.81	6.44
‡InterFormer [12]	ViT-B	1.42	1.50	1.73	3.11	3.69	6.13	4.53	5.56
‡SimpleClick [21]	ViT-B	1.38	1.48	1.36	1.97	3.43	5.62	3.66	5.06
‡**TIA (Ours)**	ViT-B	**1.36**	**1.40**	**1.35**	**1.59**	**3.25**	**5.29**	**3.54**	**4.78**
*SAM [15]	ViT-L	1.72	1.91	1.37	2.01	5.74	9.32	5.04	6.48
‡InterFormer [12]	ViT-L	1.33	**1.40**	1.70	2.78	3.56	5.89	4.12	5.08
‡SimpleClick [21]	ViT-L	**1.32**	**1.40**	1.34	1.89	2.95	4.89	3.26	4.81
‡**TIA (Ours)**	ViT-L	**1.32**	1.42	**1.31**	**1.53**	**2.71**	**4.56**	**2.98**	**4.46**

Table 1: Evaluation results on four datasets. Training dataset notations: "†" denotes SBD [11] (8,498 images with 20,172 masks), "‡" denotes COCO+LVIS [9, 19] (118k images with 1.2M masks), and "*" denotes SA-1B [15] (11M images with 1.1B masks).

B	Method	Berkeley						DAVIS						SPC (s)
		NoDC$_{20}$	mDIoU (%)	IoU@5 (%)	BIoU@5 (%)	ASSD@5	CME (%)	NoDC$_{20}$	mDIoU (%)	IoU@5 (%)	BIoU@5 (%)	ASSD@5	CME (%)	
ViT-B	InterFormer	701	0.19	95.00	86.49	1.25	3.95	1756	0.45	88.13	80.92	8.58	9.63	0.32
	SimpleClick	723	0.24	96.13	88.78	0.92	2.99	1685	0.79	88.96	80.65	7.94	8.99	**0.03**
	TIA (Ours)	**680**	**0.09**	**96.53**	**90.24**	**0.81**	**2.60**	**1478**	**0.26**	**90.64**	**82.97**	**6.89**	**6.29**	0.04
ViT-L	InterFormer	**671**	0.18	95.40	87.58	0.99	3.70	1955	0.56	87.90	81.91	7.95	9.23	0.60
	SimpleClick	775	0.16	95.74	89.43	0.96	2.94	1547	0.52	89.25	83.22	6.47	8.81	**0.07**
	TIA (Ours)	737	**0.08**	**96.96**	**91.65**	**0.76**	**2.37**	**1493**	**0.32**	**91.36**	**84.85**	**6.28**	**5.83**	0.10

Table 2: Comparisons of previous methods on the seven metrics.

1) achieved better segmentation results with fewer clicks, which outperformed other counterparts on all four benchmarks; 2) exhibited less frequent and smaller average decrease in accuracy when a new click deteriorated the segmentation results, demonstrating the stability of our method; 3) attained higher peak accuracy within the constraint of no more than 20 clicks; 4) demonstrated a low computational budget.

The qualitative results of our method and previous methods are visualized in Figure 5. Compared to the previous methods [4, 12, 21, 39], our method generated more accurate segmentation results when provided with the same number of clicks. The contours of the foreground objects in our segmentation masks closely align with those in the ground truth. Please refer to the supplementary material for more visualization examples.

4.3 Ablation Study

Ablation study experiments were conducted to assess the effectiveness of each proposed component. We reported the results on the DAVIS dataset because it covers challenging scenarios, including unseen categories, motion blur, and occlusions, and has more fine-grained annotations. The results listed in Table 3 indicate that each component has contributed to the performance improvement related to segmentation quality and stability. Among them, MI had the most significant impact on the segmentation results, demonstrating that incor-

Backbone	IP	MI	DA	NoC@90%	IoU@5(%)	BIoU@5(%)	ASSD@5	NoDC_{20}	mDIoU(%)	CME(%)
ViT-B				5.06	88.96	80.65	7.94	1685	0.79	8.99
ViT-B		✓	✓	4.84	90.23	82.55	7.02	1502	0.32	6.74
ViT-B	✓		✓	4.93	89.19	81.24	7.58	1598	0.41	7.44
ViT-B	✓	✓		4.88	89.37	81.69	7.28	1526	0.49	7.39
ViT-B	✓	✓	✓	**4.78**	**90.64**	**82.97**	**6.89**	**1478**	**0.26**	**6.29**
ViT-L				4.81	89.25	83.22	6.47	1547	0.52	8.81
ViT-L		✓	✓	4.54	90.99	84.63	6.34	1501	0.34	6.29
ViT-L	✓		✓	4.67	89.86	83.87	6.39	1547	0.45	6.97
ViT-L	✓	✓		4.63	90.56	84.25	6.37	1532	0.36	6.53
ViT-L	✓	✓	✓	**4.46**	**91.36**	**84.85**	**6.28**	**1493**	**0.32**	**5.83**

Table 3: An ablation study for the core components on the DAVIS dataset.

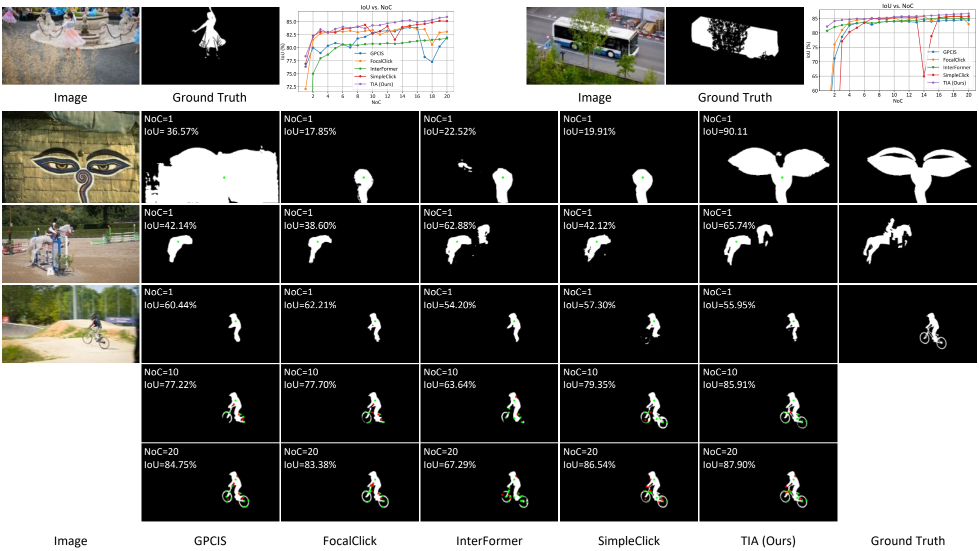

Figure 5: Qualitative comparisons of GPCIS [39], FocalClick [4], InterFormer [12], SimpleClick [21], and our method.

porating temporal information into the current state helps improve the network's ability to understand the foreground and background. Refer to the supplementary material for more ablation study results.

5 Conclusion

To mitigate accuracy fluctuations and minimize constrained minimal errors for interactive image segmentation, this paper leverages the concept of the PID control and introduces the TIA method. TIA enhances the dominant role of a new click by propagating the correction intention globally. It also leverages past round's feature descriptors to enhance the network's understanding of foreground and background, and perceives the trend of segmentation results changes to improve prediction and correction capabilities of the network. By incorporating past, current, and future information in multi-round interactions, the network can quickly respond and refine errors accurately. Experimental results with four different backbones on four datasets demonstrate that TIA effectively reduces user annotations, enhances segmentation quality and stability, and exhibits strong generalization capabilities.

Acknowledgements

This research was supported by NSFC (62021002) and Tsinghua-Kuaishou Institute of Future Media Data.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, pages 859–868, 2018.

[2] B. Wayne Bequette. *Process control: modeling, design, and simulation*. Prentice Hall Professional Technical Reference, Upper Saddle River, New Jersey, 2003.

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

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

[5] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, pages 15334–15342, 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, pages 1–9, 2021.

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

[8] Chaowei Fang, Ziyin Zhou, Junye Chen, Hanjing Su, Qingyao Wu, and Guanbin Li. Variance-insensitive and target-preserving mask refinement for interactive image segmentation. In *AAAI*, 2024.

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

[10] Yuying Hao, Yi Liu, Zewu Wu, Lin Han, Yizhou Chen, Guowei Chen, Lutao Chu, Shiyu Tang, Zhiliang Yu, Zeyu Chen, et al. Edgeflow: Achieving practical interactive segmentation with edge-guided flow. In *ICCVW*, pages 1551–1560, 2021.

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

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

[13] Chuong Huynh, Yuqian Zhou, Zhe Lin, Connelly Barnes, Eli Shechtman, Sohrab Amirghodsi, and Abhinav Shrivastava. Simpson: Simplifying photo cleanup with single-click distracting object segmentation network. In *CVPR*, pages 14518–14527, 2023.

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

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[16] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, pages 280–296. Springer, 2022.

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

[18] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. pages 2746–2754, 2017.

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

[20] Zheng Lin, Zheng-Peng Duan, Zhao Zhang, Chun-Le Guo, and Ming-Ming Cheng. Focuscut: Diving into a focus view in interactive segmentation. In *CVPR*, pages 2637–2646, 2022.

[21] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *ICCV*, pages 22290–22300, 2023.

[22] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, 2001.

[23] Nicolas Minorsky. Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, 34(2):280–309, 1922.

[24] Gordon J. Murphy. *Basic Automatic Control Theory*. D. Van Nostrand Company, 1958.

[25] Tolga Öztürk, Yalçın Albayrak, and Övünç Polat. Object tracking by pi control and image processing on embedded systems. In *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, pages 2178–2181. IEEE, 2015.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, pages 8026–8037, 2019.

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

[28] Marko Radeta, Ruben Freitas, Claudio Rodrigues, Agustin Zuniga, Ngoc Thi Nguyen, Huber Flores, and Petteri Nurmi. Man and the machine: Effects of ai-assisted human labeling on interactive annotation of real-time video streams. *ACM Transactions on Interactive Intelligent Systems*, 2024.

[29] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM TOG*, 23(3):309–314, 2004.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

[31] Ajeet Singh, S Chatterjee, and Ritula Thakur. Design of tracking of moving target using pid controller. *International Journal of Engineering Trends and Technology*, 15 (8):403–406, 2014.

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

[33] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *ICIP*, pages 3141–3145. IEEE, 2022.

[34] Guotai Wang, Wenqi Li, Maria A Zuluaga, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE TMI*, 37(7):1562–1573, 2018.

[35] Qiaoqiao Wei, Hui Zhang, and Jun-Hai Yong. Focused and collaborative feedback integration for interactive image segmentation. In *CVPR*, pages 18643–18652, 2023.

[36] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, volume 34, pages 12077–12090. Curran Associates, Inc., 2021.

[37] Jiacong Xu, Zixiang Xiong, and Shankar P Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. In *CVPR*, pages 19529–19539, 2023.

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

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