ClickAdapter: Integrating Details into Interactive Segmentation Model with Adapter

Shanghong Li[1], Yongquan Chen[1]*, Long Xu[1]*, Jun Luo[2], Rui Huang[1], Feng Wu[3], Yingliang Miao[4]

***Abstract*—Click-based interactive segmentation is the most concise and widely used data labeling method. While existing interactive segmentation methods excel in handling simple targets, they encounter challenges in obtaining high-quality masks from some complex scenes, even with a large number of clicks. Also, the cost of retraining the model from scratch for special scenarios is unacceptably high. To address these issues, we propose ClickAdapter, a simple yet powerful interactive segmentation model adapter without the need for no pre-training. Through introducing a small number of additional parameters and computations, the adapter module effectively enhanced the ability of interactive segmentation models to obtain high-quality prediction with limited clicks. Specifically, we incorporate a detail extractor that aims to extract spatial correlations and local detail features of images. These fine-grained data are then integrated into a model with our adapter to generate segmentation masks with sharp and precise edges. During the training process, only the parameters of our adapter are learnable, thereby reducing the training cost. Features in special scenarios can also be infused more efficiently. To verify the efficiency and performance advantages of the proposed method, a series of experiments on a wide range of benchmarks were conducted, demonstrating that the proposed algorithm achieved cutting-edge performance compared to current state-of-the-art (SOTA) methods.**

***Index Terms*—Human-computer interaction, interactive segmentation, adapter, spatial correlations, training cost.**

I. Introduction

INTERACTIVE segmentation aims to utilize limited user interaction information to achieve the segmentation of target instances in images. This method provides an efficient way to obtain large-scale annotated data, making it an important task in human computer interaction and computer vision. The current paper focuses on interactive segmentation based on click points, where foreground and background are defined by simple positive and negative clicks, respectively. After each click, a segmentation prediction of the target instance is returned. The objective is to achieve higher target accuracy with fewer parameters and click iterations.

Most early research on click-based interactive segmentation methods focused on developing more effective segmentation backbone networks or exploring novel refinement modules, FocalClick [1], EdgeFlow [2], f-BRS [3], etc., to obtain more refined segmentation results. SimpleClick [4] first introduced Vision Transformers (ViT) [5] into interactive segmentation approaches and surpassed those relying on Convolutional Neural Networks in performance, thereby developing rapidly and achieving SOTA results. Recently, the Segment Anything Model (SAM) [6] based on pre-trained MAE [7] attracted attention and reported its performance competiable with previous SOTA algorithms such as FocalClick [1]. We applied these methods in practical annotation processes and conducted extensive experiments. In many complex scenarios, we identified two key challenges with existing algorithms. First, existing methods still struggle to meet the requirements of high-precision mask annotation tasks, as shown in Figure 1. Specifically, 1) the segmentation edges of objects with fine structures are not accurately delineated and often require extensive

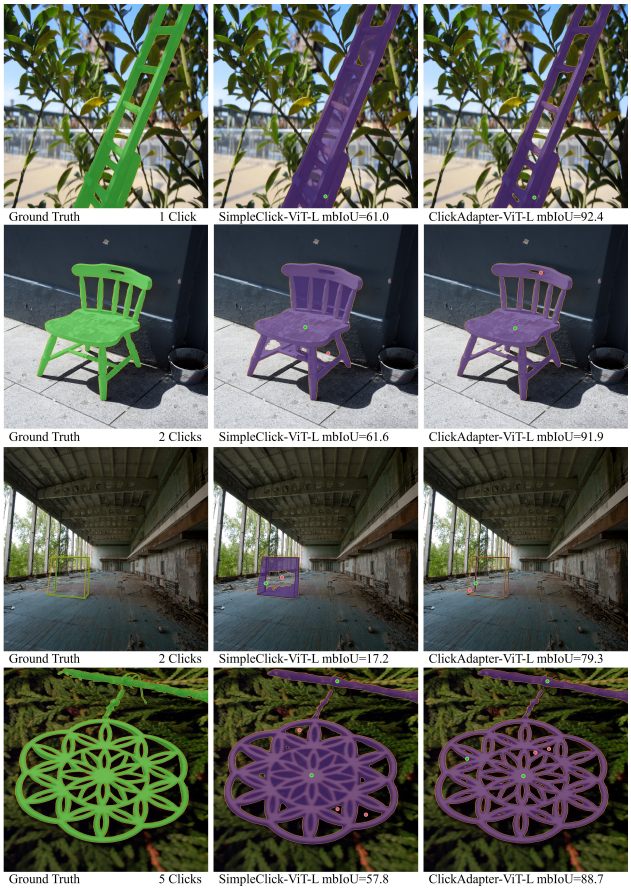


Fig. 1: Comparison of predicted masks between SimpleClick and our ClickAdapter using the same several click points on the object as input prompts. Our method yields significantly more detailed results with highly accurate boundaries.

[1] Shenzhen Institute of Artificial Intelligence and Robotics for Society, The Chinese University of Hong Kong, Shenzhen, China.
[2] Northeastern University, Shenyang, China.
[3] University of Science and Technology of China, Hefei, China.
[4] Maxvision Technology Corp., Shenzhen, China.
* Corresponding author: Yongquan Chen and Long Xu (e-mail: yqchen@cuhk.edu.cn, xulong@cuhk.edu.cn).

manual corrections to meet annotation needs, and 2) for high-precision input images, the models are prone to noticeable prediction errors, which requires more clicks for corrections. Second, existing algorithms increasingly adopt large models with more parameters and train them on large-scale datasets to enhance the models' generalization capabilities, which poses challenges in model training. In some complex scenes, only medium-sized or small datasets are available for training, and due to the large parameter volume of ViT models, training from scratch is computationally expensive and may result in the loss of knowledge learned from large-scale datasets.

To address the aforementioned issues, we propose an additional module with a detail extractor to meet the demand for high-resolution segmentation capability in high-precision mask annotation tasks. This is a small-scale network that does not require pre-training and can be attached to the Transformer-structured backbone. Without adjustments of backbone networks, the adapter module can effectively adapting the model to complex application scenarios of interactive segmentation and reducing training costs.

Specifically, the detail extractor consists of two main components: 1) a spatial prior module based on deformable convolutions and 2) an injector module based on cross-attention layers. Recent studies [8], [9] have shown that convolutional neural networks can better assist ViT in capturing local spatial information; this is because the ViT backbone network treats the image as a series of patches and extensively models the interaction between patches. However, it struggles to perceive the intrinsic structural information within each patch and the spatial relationships among patches. On the other hand, convolutional neural networks naturally excel at modeling pixel-level local relationships and preserving spatial positional connections among pixels. The spatial prior module based on deformable convolutions captures local semantic information and spatial prior features from the input image, compensating for the missing local details in the predicted mask. Additionally, it is generally believed that early transformer layers capture lower-level features and have not established sufficiently rich global information [10]. Therefore, we employ an injector module based on Two-way Attention layers to inject spatial features into the shallow features obtained from the Transformer-structured backbone network, aiming to capture fine-grained features of the image.

Throughout the training process, only the parameters of the adapter are trainable, while all parameters of the backbone network remain fixed. Our proposed adapter not only demonstrates a notable reduction in the number of trainable parameters and associated training costs for the model but also ensures the retention of performance achieved by the backbone network on large-scale datasets through fine-tuning of the adapter parameters. To substantiate the effectiveness of the adapter structure, we conducted comprehensive quantitative and qualitative experiments, evaluating the performance of the proposed method across six widely recognized interactive segmentation benchmarks, utilizing backbone networks of varying scales. Moreover, we conducted a comparative analysis with state-of-the-art interactive annotation methods, alongside selected fine-tuning methods, to assess both performance and training efficiency.

In conclusion, our contributions can be summarized as follows:

- We introduced an adapter for existing interactive segmentation models, designed to undergo fine-tuning on the pre-trained backbone network of the Transformer architecture. This approach not only attains competitive performance in novel and complex scenarios but also markedly diminishes training costs, given the adapter's limited parameter count.
- We formulated the architecture of the adapter, incorporating a detail extractor and an injector with the introduction of a minimal number of parameters. Our approach adeptly amalgamates local spatial features of images with shallow semantic features, empowering the model to execute segmentation tasks in high-resolution scenes.
- Our proposed method has demonstrated competitive performance across multiple benchmarks, manifesting an average performance improvement of 11% with the addition of merely 15% learnable parameters.

II. Related Work

A. Interactive segmentation

Before the widespread application of deep learning in the field of computer vision, many works [11]–[14] were based on traditional optimization processes to design interactive segmentation methods. In recent years, due to its outstanding performance, deep learning has quickly surpassed traditional methods and become the mainstream research direction. Xu et al. [15] was the first to introduce deep learning into interactive segmentation, defining training strategies, evaluation protocols, modeling and generation methods for click points. Subsequent works were built upon this foundation with an investigation into more effective backbone networks and finer optimization modules. Regarding the research on backbone networks and training strategies: Lin et al. [16] emphasized the significance of the first click. Jang et al. [17] initially highlighted the iterative nature of interactive segmentation and optimized segmentation masks through backpropagation based on previously generated masks. Building on this foundation, Forte et al. [18]improved the network structure and for the first time propagated previously predicted masks as inputs for learning within the network. Sofiiuk et al. [19] further enhanced the network structure and refined the previously generated masks to improve prediction accuracy. Liu et al. [4] introduced ViT into the interactive segmentation task and achieved promising performance. In terms of optimization modules, Hao et al. [2] proposed an improved method based on target contour masks. Instead of using a global mask as prior information, an additional flow was used to predict the target's contour mask. However, due to the heterogeneity between contour features and original image features, an extra module was required to align the contour with the target, resulting in increased complexity and inference time. Zhang et al. [20] presented a contour-refinement network structure to handle previously predicted masks. It designed two networks to respectively predict the target's contour and optimize the

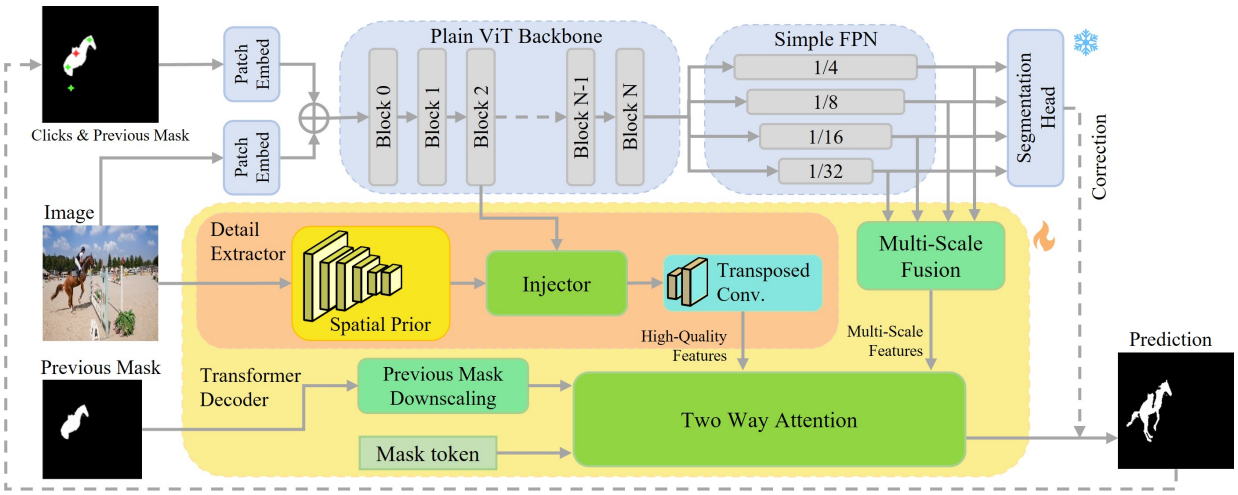


Fig. 2: Overview of our method. Our method takes the embedding image and user clicks map combined with the previous segmentation mask as input. Our method consists of three main modules: (a) a Plain ViT backbone that maintains single-scale feature maps throughout; (b) a detail extractor that extracts the spatial correlation and locally uses a Plain ViT model that maintains four-scale features throughout as our backbone; and (c) a transformer decoder that combines multiple different features. The parameters of the Plain ViT backbone network was frozen during the training process.

scaled target mask, thereby improving prediction accuracy. Chen et al. [1] proposed a method to optimize locally predicted masks through an optimization layer. These methods represented click points using click maps, which limited the model's understanding of click information. Additionally, they required additional networks to process previously predicted results, increasing model complexity and inference time. Chen et al. [3] investigated the encoding method for click points and proposed a fusion method that used convolutional layers to merge the segmentation network's predicted results with click points, optimizing the final prediction mask.

B. Vision Transformer and Adapter

In recent years, the transformer architecture has asserted its dominance in the realms of natural language processing (NLP) and speech recognition, primarily due to its outstanding attention mechanism. Dosovitskiy et al. [5] introduced this remarkable structure into the field of computer vision, where it outperformed traditional CNN models, particularly in image classification tasks. Subsequent advancements by PVT [21] and Swin [22] further incorporated a pyramid structure into ViT, resulting in superior performance attributed to its multi-level design tailored for handling image features. Segformer [23] adopts a straightforward and lightweight segmented backbone network through the hierarchical design of the Transformer structure. Nevertheless, recent studies, including those on BEiT [24], MAE [7], and PlainViT [25], have illustrated that the ViT structure, without the necessity of a multi-level design, still harbors substantial potential. Furthermore, this structure demonstrates increased flexibility and applicability, particularly in scenarios involving pre-training with multi-modal and masked data.

The concept of adapters found its initial widespread application in the field of Natural Language Processing (NLP). Stickland et al. [26] and Houlsby et al. [27] introduced novel modules into transformers for task-specific fine-tuning, allowing large pre-trained models to rapidly adapt to new downstream tasks. In the realm of computer vision, adapter structures have been proposed for progressive learning [28] and domain adaptation [29]. With the emergence of CLIP [30], numerous studies have explored the use of adapters to transfer pre-trained knowledge to zero-shot tasks in downstream applications. Recently, Li et al. [25] investigated adapters based on the Plain ViT backbone, augmenting it with additional up-sampling and down-sampling modules to tailor it for object detection tasks. Similarly, Chen et al. [9] designed an additional adapter based on the PlainViT backbone to adapt it for dense prediction tasks, yielding remarkable results.

Furthermore, numerous studies have delved into the characteristics, merits, and drawbacks of the Vision Transformer (ViT) architecture. Yuan et al. [31] highlighted the strengths of the Transformer in establishing long-range dependencies, juxtaposed with the advantages of Convolutional Neural Networks (CNNs) in extracting low-level features and enhancing locality. Chen et al. [9] emphasized that CNNs can leverage spatial prior features to compensate for the limitations of the PlainViT structure in dense prediction tasks. In a separate study, Wu et al. [8] examined the features propagated through different layers in the ViT structure, noting that early features tend to be lower-level, while deep features generally embody more abstract semantic features.

III. METHOD

In this chapter, we first introduce the overall structure of the proposed adapter. Then, we proceed to discuss the structure of the detail extractor within the adapter, which is designed to capture spatial features of target instances and local edge features. Finally, we discuss the design of the decoder, which aims to fuse detail features with minimal parameters to enhance segmentation performance.

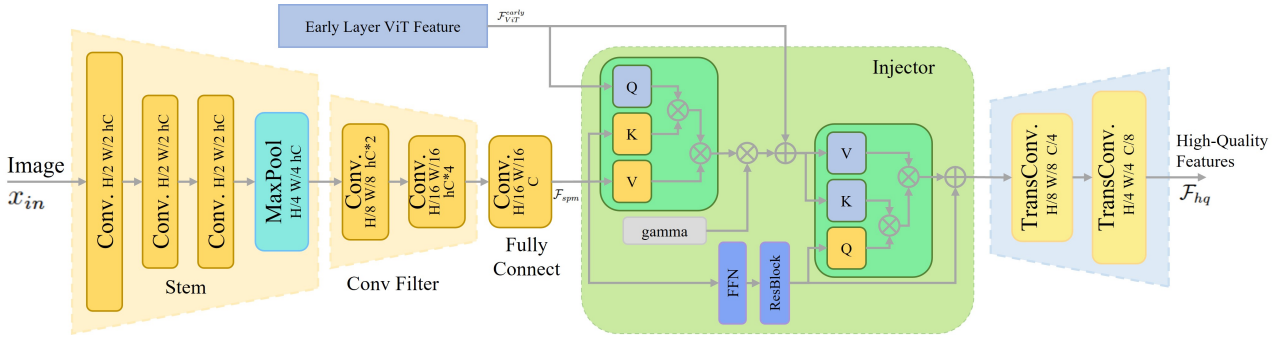


Fig. 3: Schematic diagram of the detail extractor. The detail extractor consists of three main modules: (a) a spatial prior module that extracts image spatial features using deformable convolutions, (b) an injector module that fuses shallow features from the backbone network with spatial features, and (c) a decompression module that obtains high-quality features.

A. Overall Structure

As shown in Figure 2, our model can be divided into two parts. The first part is the backbone network of Plain ViT, which consists of two patch embedding layers, N transformer encoder blocks, and a simple feature pyramid network. The second part is the proposed adapter, which includes a detail extractor module and a transformer decoder module.

Based on the analysis of computational cost and performance of various backbone networks in interactive image segmentation tasks in [4], we chose the plain non-hierarchical Plain ViT-Base [32] as the backbone of our segmentation model. The input of the network, denoted as x, consists of the input image $x_{in} \in \mathbb{R}^{B\times 3\times H\times W}$, the click map $x_c \in \mathbb{R}^{B\times 2\times H\times W}$, and the predicted result of the previous click $x_m^{t-1} \in \mathbb{R}^{B\times 1\times H\times W}$. The two channels in x_c represent the positions of positive and negative clicks, respectively. Here, B represents the batch size, and t represents the t-th click. x_p represents the position distribution of the target object, while x_m represents the final segmentation mask.

Specifically, the input image is first divided into a series of non-overlapping patches of size 16×16. Each patch's feature is linearly transformed using the patch embedding layer to map it into a feature vector of dimension C, forming a sequence of length L. The sequence is then fed into N stacked transformer blocks, where each block consists of a multi-head self-attention layer. The output of the feature vectors after the nth layer is denoted as $\mathcal{F}_{ViT}^n \in \mathbb{R}^{B\times C\times L}$. The feature vector after the last layer is denoted as $\mathcal{F}_{ViT}^N$, which contains the strongest global features of the image. We apply f^N to obtain multi-scale features. We construct a simple feature pyramid using a set of convolutions and transposed convolutions. By setting different convolution strides, we obtain feature maps of sizes $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ compared to the original image size. They are respectively denoted as $\mathcal{F}_{\frac{1}{4}}^N, \mathcal{F}_{\frac{1}{8}}^N, \mathcal{F}_{\frac{1}{16}}^N, \mathcal{F}_{\frac{1}{32}}^N$.

B. Detail Extractor

To obtain more accurate segmentation results, rich global semantic context information and local edge detail information are needed. To achieve this, we introduce a detail extractor to obtain additional spatial prior information and detailed edge information. Figure 3 illustrates the specific structure of the proposed detail extractor.

First, we introduce a spatial prior module. Given the normalized input image, x_{in}, we employ deformable convolutions to capture local pixel correlations and spatial contextual relationships. The spatial prior module consists of three components: 1) the backbone layer, 2) the convolutional filtering layer, and 3) the fully connected layer. For the backbone layer, we adopt a convolutional backbone structure based on the standard ResNet, which includes three deformable convolutional layers and one max-pooling layer to capture spatial features of the image. Batch normalization (BatchNorm) and ReLU layers are applied between the convolutional layers to enhance generalization capability. The output of the backbone layer, denoted as $\mathcal{F}_{stm} \in \mathbb{R}^{B\times \frac{hd}{4}\times \frac{HW}{4^2}}$, reduces the spatial scale of the image and maps the channels to the hidden dimension $\frac{hd}{4}$. Next, we use cascaded deformable convolutions with a stride of 2 and a 3×3 kernel to form the convolutional filtering layer, which doubles the number of channels and reduces the size of the feature maps. The output of the convolutional filtering layer is $\mathcal{F}_{cf} \in \mathbb{R}^{B\times hd\times \frac{HW}{16^2}}$. Finally, we employ deformable convolutions with 1×1 kernels as the fully connected layer to project the feature maps to the feature dimension C of the Plain ViT, obtaining the output of the spatial prior module, $\mathcal{F}_{spm} \in \mathbb{R}^{B\times C\times \frac{HW}{16^2}}$.

Next, we propose an injector module to integrate the output of the spatial prior module with the output of the Plain ViT backbone network. Specifically, based on cross-attention, we fuse the spatial prior features, $\mathcal{F}_{spm}$, with the features from the early layers of the Plain ViT backbone network, $\mathcal{F}_{ViT}^n$. Research on ViT [5] has shown that later blocks in the backbone network have longer attention distances, whereas earlier blocks are more localized and contain lower-level semantic detail features. Based on this characteristic, we extract the output of the early attention block in the Plain ViT backbone network as the feature. Specifically, for the Plain ViT backbone based on ViT-Base, we select the output of the third block out of a total of 12 blocks and denote it as $\mathcal{F}_{ViT}^{early}$.

We use $\mathcal{F}_{ViT}^{early}$ as the query, $\mathcal{F}_{spm}$ as the key and value, and apply cross-attention to incorporate spatial features into the early-stage ViT features, as shown in equation (1) below:

$$\hat{\mathcal{F}}_{ViT}^{early} = \mathcal{F}_{ViT}^{early} + \gamma \mathbf{A}(norm(\mathcal{F}_{ViT}^{early}), norm(\mathcal{F}_{spm})) \quad (1)$$

where $\mathbf{A}(\cdot)$ represents the cross-attention layer, the input at

position one serves as the query, and the input at position two serves as the key and value. $norm(\cdot)$ denotes the LayerNorm layer. $\gamma \in \mathbb{R}^C$ is a learnable parameter used to associate the output of the attention layer in the residual connection. It is initialized to zero, and because the output of the attention layer is non-zero, during the first iteration of gradient descent, the parameter γ will be optimized to a non-zero value. In this way, the output of the spatial prior module and the ViT backbone network can be gradually balanced in a learned manner.

After incorporating spatial features into the early-stage ViT features, we apply a module consisting of a feed-forward network (FFN) and a cross-attention layer to extract high precision features. This process is illustrated by equations (2) and (3):

$$\hat{\mathcal{F}}_{spm} = \mathcal{F}_{spm} + \mathbf{FFN}(norm(\mathcal{F}_{spm}) \tag{2}$$

$$\hat{\mathcal{F}}_{hq} = \hat{\mathcal{F}}_{spm} + \mathbf{A}(norm(\mathcal{F}_{spm}), norm(\hat{\mathcal{F}}^{early}_{ViT})) \tag{3}$$

where the output of the feed-forward network, $\hat{\mathcal{F}}_{spm}$, serves as the query for the cross-attention layer, and the output of the previous layer, $\hat{\mathcal{F}}^{early}_{ViT}$, serves as the key and value for the cross-attention layer.

Finally, we designed a feature compression module that utilizes two transpose convolutional layers to compress the output, $\hat{\mathcal{F}}hq$, of the cross-attention layer. This compression reduces the dimensionality while increasing the size of the feature maps, resulting in the final high-quality feature, $\mathcal{F}ViT^{hq}$.

C. Adapter Design

Training costs are high due to the large number of network parameters in the ViT backbone network. To train more efficiently, we propose the construction of an efficient adapter for the ViT backbone network. In this section, we introduce the specific structure of the adapter designed for Plain ViT, and its overall architecture is illustrated in Figure 4.

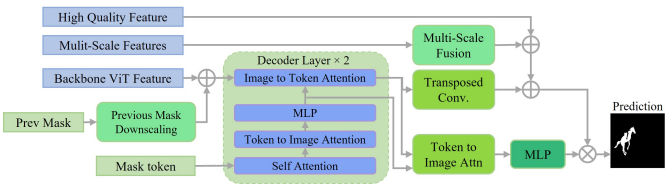


Fig. 4: Schematic diagram of the adapter module, which fuses high-quality features with the feature output of the backbone network and decodes the features to obtain the output mask.

First, we introduce an efficient token learning approach in the adapter to enhance its ability of learning high-quality prediction masks. As shown in the Figure 4, this token undergoes decoding through two decoder layers. In each decoder layer, the token is first updated through self-attention; features are then updated through a bidirectional cross-attention layer between the token and the image, and vice versa. After passing through the decoder layers, the output token is associated with the global contextual features of the image. Finally, a three-layer MLP is added, and the updated output token can predict dynamic MLP weights, generating dynamic convolutional kernels.

Second, we introduce a Dense Input Embedding module in the adapter. This module allows the previous predicted mask to be input into the adapter, enhancing the stability of the model's prediction results and preventing significant changes in the predicted mask during consecutive clicks. The module consists of four 2D convolutional layers, along with corresponding LayerNorm layers and activation functions, which reduce the size of the mask to $\frac{1}{16}$ of its original size and map it to the feature dimension C. This dense input is then concatenated with the image feature output of the Plain ViT network and fed into the transformer decoder layers for decoding.

Finally, we introduce a Multi-Scale Feature Fusion module in the adapter, which takes the multi-scale features obtained from a simple feature pyramid network in the backbone network and fuses them before inputting them into the adapter. According to the research on Plain ViT [32], this simple feature pyramid structure effectively extracts visually specific inductive biases from the backbone network. In the adapter, we design a Multi-Scale Feature Fusion layer that uses deconvolution layers with different strides to compress features of different scales, thereby reducing the feature dimension to $\frac{C}{4}$ and unifying the feature size to $\frac{1}{4}$ of the input image size. Finally, the compressed multi-scale features are concatenated along the feature dimension and passed through a fully connected layer to output features with a dimension of $\frac{C}{8}$.

During the training process, we fix the model parameters of the pre-trained Plain ViT model and allow only the proposed adapter and detail extractor to be learned. Thus, the learnable parameters include the convolutional layers and cross-attention layers in the detail extractor, the output token and its associated three-layer MLP in the adapter, the downsampling convolutional layer in the Dense Input Embedding module, and the two decoder layers. During inference, we use the predictions from the adapter as high-quality prediction results. To correct the output prediction mask, we compute a logarithmic sum of the output predictions from Plain ViT's ordinary semantic segmentation head and our high-quality mask predictions from the adapter. The corrected result is then upsampled to obtain the final output.

IV. Experiments

This chapter initiates with an introduction to the foundational configuration of the proposed adapter and backbone network, accompanied by a delineation of the interactive training and validation strategy. Subsequent sections include a meticulous accuracy comparison between our proposed algorithm and existing state-of-the-art (SOTA) algorithms, utilizing established evaluation datasets. Noteworthy findings underscore that our proposed adapter markedly improves segmentation quality, concurrently preserving the expeditious and cost-effective nature of the training process, in contrast to the comprehensive network's performance across both training and inference phases. Finally, ablation experiments were conducted to rigorously validate the efficacy of each submodule of the proposed method.

TABLE I: Evaluation results of methods based on small-scale backbone networks on GrabCut [14], Berkeley [33], DAVIS [34], SBD [35], COCO MVal [36] and PascalVOC [37]. 'NoC 85/90' denotes the average Number of Clicks required to get an IoU of 85/90%. The first and the second parts display the results of all methods trained on SBD [35] and COCO [36]+LVIS [38], respectively, and the last part presents the results trained on HQSeg-44K [39]. The symbol '*' signifies direct citation of the experimental results from the corresponding papers. **Bold** data indicates the best performance on the current evaluation benchmark.

Method	GrabCut	Berkeley	SBD		DAVIS		COCO_MVal		PascalVOC	
	NoC 90	NoC 90	NoC 85	NoC 90	NoC 85	NoC 90	NoC 85	NoC 90	NoC 85	NoC 90
*f-BRS-B-hrnet32 [3]	2.16	3.69	4.31	7.08	5.54	7.62	3.82	5.44	-	-
*RITM-hrnet18s [19]	2.04	3.22	**3.39**	**5.43**	4.94	6.71	-	4.39	2.51	-
*FocalClick-segformer-B0-S2 [1]	1.90	3.14	4.34	6.51	5.02	7.06	-	-	-	-
Ours-segformer-B0	1.44	1.87	4.36	6.68	3.99	5.38	3.00	3.54	2.78	3.87
Ours-segformer-B3	**1.42**	**1.57**	4.79	5.97	**2.98**	**4.33**	**2.59**	**3.03**	**2.50**	**3.50**
*f-BRS-B-hrnet32 [3]	1.74	2.61	4.29	7.20	4.94	6.36	2.54	3.43	-	-
*RITM-hrnet18s [19]	1.68	2.60	4.04	6.48	4.70	5.98	-	3.33	2.57	-
*RITM-hrnet32 [19]	1.56	2.10	3.59	5.71	4.11	5.34	-	**2.97**	2.57	-
*EdgeFlow-hrnet18 [2]	1.72	2.40		-	4.54	5.77	-	-	**2.50**	-
FocalClick-segformer-B0-S2 [1]	1.66	2.27	4.56	6.86	4.04	5.49	3.23	4.37	3.55	4.24
FocalClick-segformer-B3-S2 [1]	1.50	1.92	**3.53**	**5.59**	3.61	4.90	3.45	3.33	2.53	2.97
Ours-segformer-B0	1.54	2.17	4.29	6.54	3.97	5.22	2.75	3.66	2.93	3.44
Ours-segformer-B3	**1.46**	**1.86**	3.73	5.85	**2.99**	**4.37**	**2.47**	3.37	2.53	**2.94**
Ours-segformer-B0	1.50	1.87	4.45	6.71	3.65	5.07	2.87	3.87	3.12	3.63
Ours-segformer-B3	1.40	1.57	3.97	6.16	2.91	4.48	2.55	3.47	2.72	3.14

TABLE II: Evaluation results of methods based on large-scale backbone networks on GrabCut [14], Berkeley [33], DAVIS [34], SBD [35], COCO MVal [36] and PascalVOC [37]. 'NoC 85/90' denotes the average Number of Clicks required to get an IoU of 85/90%. The first and the second parts display the results of all methods trained on SBD [35] and COCO [36]+LVIS [38], respectively, and the last part presents the results trained on HQSeg-44K [39]. **Bold** data indicates the best performance on the current evaluation benchmark. All methods adopt the same training strategy, that is, using the SimpleClick pre-trained backbone network and freezing the backbone network for training.

Method	GrabCut	Berkeley	SBD		DAVIS		COCO_MVal		PascalVOC	
	NoC 90	NoC 90	NoC 85	NoC 90	NoC 85	NoC 90	NoC 85	NoC 90	NoC 85	NoC 90
SimpleClick-ViT-B [4]	1.54	2.46	3.28	5.24	4.10	5.48	-	-	2.38	2.81
SimpleClick-ViT-L [4]	1.46	2.33	**2.69**	**4.46**	4.12	5.39	-	-	1.95	2.30
Ours-ViT-B	1.42	1.89	3.26	5.27	4.03	5.28	2.25	3.18	2.16	**2.52**
Ours-ViT-L	**1.38**	**1.77**	2.78	4.53	**3.88**	**5.08**	**2.03**	**2.83**	**2.12**	2.75
SimpleClick-ViT-B [4]	1.48	1.97	3.43	5.62	3.66	5.06	-	-	2.06	2.38
SimpleClick-ViT-L [4]	1.40	1.89	2.95	4.89	**3.26**	4.81	-	-	1.72	1.96
CRFasRNN-ViT-L [40]	1.48	1.83	3.27	5.32	3.47	4.83	2.14	2.96	2.09	2.41
PAMR-ViT-L [41]	1.70	3.08	5.22	6.75	4.87	6.12	2.91	6.45	2.29	2.94
FocalClick-ViT-L [1]	1.60	2.04	4.06	6.64	3.72	5.39	2.52	2.97	2.69	3.67
Ours-ViT-B	1.42	1.74	3.18	5.20	3.48	4.86	2.13	2.91	2.00	2.31
Ours-ViT-L	**1.38**	**1.53**	**2.77**	**4.62**	3.27	**4.72**	**1.97**	**2.73**	**1.69**	**1.93**
SAM-H [6]	1.62	2.25	5.98	9.63	4.88	6.21	3.46	5.60	2.75	3.33
HQ-SAM-H [39]	1.84	2.00	6.23	9.66	4.15	5.58	3.81	5.94	2.50	2.93
Ours-ViT-B	1.38	1.56	3.66	5.58	3.52	4.86	**2.10**	**2.91**	**2.02**	**2.32**
Ours-ViT-L	**1.30**	**1.47**	**3.42**	**5.43**	**2.54**	**4.14**	2.48	3.19	2.17	2.44

A. Experimental Configuration

Model selection. Research findings from SimpleClick [4] reveal that PlainViT achieves superior accuracy compared to other hierarchical backbone networks in interactive image segmentation tasks. Consequently, this paper opts for the PlainViT series as the backbone network for our model and conducts comparisons with alternative methods employing similar PlainViT backbone networks. Additionally, to ensure a fair evaluation against FocalClick [1] and other methods utilizing small-scale segmentation networks (e.g., hrNet [42], Segformer [23]), our proposed method is applied to the Segformer series backbone network during training.

Training protocol. To generate training data, we randomly cropped images into a size of 448×448, considering that the precise dependencies between different click patches need to be calculated for click attention. All training was conducted in an end-to-end manner.

Regarding the simulation strategy for click points, we employed the iterative learning strategy from RITM [3] and sample positive and negative clicks using the training sample generation strategy proposed from [15]. The maximum number of click points during training was set to 24, with a decay probability of 0.8.

Data augmentation was performed during training using random flipping and random resizing. We used the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_1 = 0.999$. Each epoch consists of 30,000 training samples.

It is worth noting that our method that fixed all parameters of the backbone network during training, significantly reduced the number of learnable parameters, making the training process fast and affordable. Compared to other algorithms that use 230 epochs, our method only needed 60 epochs. The initial learning rate was set to 5×10^{-6} and was subsequently reduced by a factor of 10 at epochs 50 and 55. Our model was trained on 4 RTX2080Ti GPUs and completed the process in approximately 8 hours.

Evaluation strategy. To ensure equitable comparisons, we adhered to the performance evaluation strategy established in [3], [15]–[17], [19], [43]. During testing, each click point was sampled from the center of the region exhibiting the highest prediction error in the preceding results. This approach guarantees that the predicted results either attain the desired Intersection over Union (IoU) with the ground truth or reach the maximum number of clicks.

The performance of the algorithm was evaluated using NoC IoU (Number of Clicks), representing the average number of clicks needed to achieve the target IoU. We set the maximum number of clicks to 20, and surpassing this limit indicates a failure in the task at hand.

B. Comparison with State-of-the-Art

Performance on existing benchmarks. To facilitate a more comprehensive and equitable comparison, we conducted training on our method using both the Segformer series with smaller parameters and the PlainViT series with larger parameters. Subsequently, we compared the performance against the current SOTA algorithm employing backbone networks of varying sizes on existing benchmarks. The outcomes of these comparisons are presented in Table I and Table II, respectively. Early classical methods such as GraphCut [44] and Geodesic Star Convexity [12] were excluded from the comparison due to their performance disadvantages.

Training was conducted on three datasets. SBD [35] and COCO [36]+LVIS [38] serve as the current training datasets for most SOTA methods, ensuring a more comprehensive and equitable basis for comparison. HQSeg44K [39] is a high-precision segmentation dataset recently proposed and utilized in HQ-SAM. Its inclusion allows us to explore the upper limits of our method. Validation was conducted across six test benchmarks: GrabCut [14], Berkeley [33], DAVIS [34], SBD [35], PascalVOC [37], and COCO MVal [36]. The inclusion of these diverse datasets in our training and validating regimen is deliberate, aiming to ensure the stability and robustness of the advantages offered by our method.

All the SOTA algorithms included in Table I have openly learned all the model parameters, encompassing both the backbone network and additional modules. The comparison in Table II followed the training methodology outlined in Section III-C. In this approach, only the parameters of our adapter module or refine module, such as CRFasRNN [40], PAMR [41], FocalClick [1], were made available for learning, while the parameters of the backbone network were loaded from the pre-trained PlainViT by SimpleClick [4] and kept fixed.

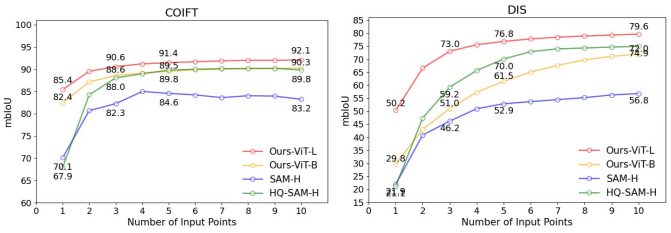
Analyzing the data presented in Table I and Table II, it becomes evident that, under the same training set, our method consistently achieved the highest or second-highest performance across all mainstream benchmark tests. On the Segformer series backbone network with fewer parameters, our method outperforms FocalClick, which also employs smaller parameters, across two distinct network settings: B0 and B3. When utilizing the larger scale of the backbone network, our method surpassed the current state-of-the-art algorithm SimpleClick by approximately 17% on the Berkeley test set, approximately 6% on the SBD and PascalVOC test sets, approximately 3% on the DAVIS dataset, and maintained comparable performance on the GrabCut dataset. Following training on the latest HQSeg-44K dataset, our algorithm demonstrated performance improvements of approximately 7% to 9% across multiple datasets.

TABLE III: Comparison of computational metrics: model parameters, FLOPs, and speed (measured by seconds per click). Each method is marked with the type of backbone network used and the input image size. *Since the original ViT-Adapter is not available for interactive segmentation, we modified the input embedding module and segmentation head to as same as ours for fair comparison.

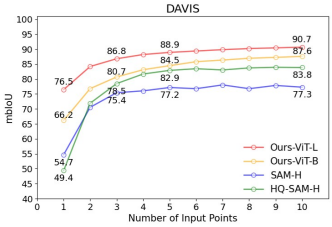
Model Type	Params(MB)	FLOPs(G)	Speed/ms
RITM-hrnet32-400	30.95	41.56	150
FocalClick-B0-S2-256	3.72	1.77	23
FocalClick-B3-S2-256	45.66	12.37	90
FocalClick-ViT-L-448	322.19	266.78	303
SimpleClick-ViT-B-448	84.89	96.46	162
SimpleClick-ViT-L-448	322.18	266.44	300
SimpleClick-ViT-H-448	659.39	700.96	585
*ViT-Adapter-B-448	142.57	151.51	-
*ViT-Adapter-L-448	405.22	397.03	-
SAM-H-1024	637.23	2830.34	665
HQ-SAM-H-1024	635.63	2802.69	658
Ours-B0-448	7.06	9.54	31
Ours-B3-448	58.77	49.95	103
Ours-ViT-B-448	128.65	148.74	195
Ours-ViT-L-448	378.30	390.19	310

Computation analysis. Table III provides a comprehensive comparative analysis of various model types, considering crucial factors such as the number of model parameters, FLOPs, and inference speed on the CPU. Given that prior research predominantly employed small-scale backbone models, we selected Segformer as a representative for comparison. In recent years, the landscape has witnessed the rapid development of large models, with many studies adopting sizes exceeding 300MB and input dimensions ranging from 400 to 600 pixels. SAM utilizes an input size of 1024, trading off some efficiency for superior segmentation results. Our method adopts PlainViT as the backbone with a default input size of 448×448, striking a balance between inference speed and memory consumption in practical annotation processes while achieving more accurate segmentation results. In comparison to SimpleClick, which also utilizes the PlainViT backbone network, our method based on the ViT-Base backbone achieves performance comparable to SimpleClick based on the ViT-Large backbone, albeit with approximately 20% parameter overhead and some compromises in inference speed.

Comparison with SAM on Point-based interactive Segmen-



(a) Results on COIFT [45] (b) Results on DIS [46]



(c) Results on DAVIS [34]

Fig. 5: Comparison of mbIoU under a varying number of positive and negative clicks with SAM and HQ-SAM. We use the evaluation strategy proposed in DIO [15] on the COIFT [45], DIS val set [46] and DAVIS [34] datasets. The abscissa represents the number of positive and negative clicks, and the ordinate represents the mbIoU split under the current click.

TABLE IV: Efficiency comparison with fine-tuning methods. 'B' and 'L' denote using pre-trained PlainViT-Base and PlainViT-Large as the backbone, respectively. The symbol * indicates a direct citation of the experimental results in the corresponding papers.

	Params(M)		Train Time	Memory	mBIoU
	Total	Learnable	s/iter	M	10 clicks
SAM-B-1024	358.32	358.32	-	*5100	78.1
HQSAM-B-1024	362.10	4.12	-	*5100	82.5
Finetune-B-448	96.72	96.72	1.83	5733	84.6
LoRA-B-448	114.15	0.56	1.29	4100	78.9
Ours-B-448	128.65	32.95	1.32	4367	87.6
Finetune-L-448	322.42	322.42	2.60	12039	86.1
LoRA-L-448	382.58	1.50	1.87	7516	86.3
Ours-L-448	378.30	57.12	1.62	7279	90.7

tation. To assess the segmentation performance of our method compared to the SOTA interactive annotation methods SAM [6] and HQ-SAM [39] based on click point guidance, we conducted accuracy comparisons on the test sets of COIFT [45] and DIS [46] under an equivalent number of click point inputs. To ensure a fairer comparison, we utilized their official predictors and weights with limited modifications, adjusting only the input format to align with the requirements of the SAM Predictor. For a more precise evaluation, we adopted the boundary metric mbIoU, as used in HQ-SAM [39], employing a stricter evaluation with an inflation ratio of 0.02. Notably, SAM and HQ-SAM used an image input size of 1024×1024, whereas our method operated with an input size of 448×448, placing our approach at a disadvantage in the comparison.

The test results, illustrated in Figure 5(a), Figure 5(b), and Figure 5(c), consistently demonstrate the superior performance of our method compared to SAM and HQ-SAM across various click numbers in click-based interactive segmentation tasks.

TABLE V: Comparison of training results on partial training set. The subsets include 500, 1000, 2000 and all samples randomly selected from the HQSeg44K [39] training set, respectively. All experimental methods are based on the pre-trained SimpleClick-B backbone network. We select two high-quality data sets, DAVIS [34] and COIFT [45], and use mbIoU NoC as the test benchmark.

Method & Samples	COIFT		DAVIS	
	bNoC85	bNoC90	bNoC85	bNoC90
Fine tune-500	4.79	12.09	8.10	12.68
LoRA-500	4.14	10.74	7.85	12.10
Ours-500	3.98	10.07	7.25	10.41
Fine tune-1000	4.22	10.78	7.67	11.39
LoRA-1000	4.06	9.10	7.42	12.21
Ours-1000	3.21	8.64	6.83	10.21
Fine tune-2000	3.98	10.22	7.50	11.19
LoRA-2000	3.45	9.19	7.60	10.92
Ours-2000	3.22	8.60	6.84	10.12
Fine tune-all	3.67	9.86	7.26	11.21
LoRA-all	3.40	9.92	7.32	11.09
Ours-all	3.10	8.50	6.66	9.99

Particularly noteworthy is the mbIoU for the initial click, which surpassed SAM by a remarkable 30.9%, 170.3%, and 54.9% on the three datasets, respectively.

Comparison with Fine-tune moethods. Given the lack of investigation on fine-tuning or adapter structures in previous studies of click-based interactive segmentation methods, we designed the following method to evaluate the improvements in training efficiency introduced by our approach. Subsequently, we conducted a comparative analysis of the training cost and performance between fine-tuning, LoRA [47] and ours.

First, we employed PlainViT-Base and PlainViT-Large pre-trained models from SimpleClick as the backbone networks, respectively. With a consistent image input size of 448×448, we conducted a comprehensive comparison of the training cost and inference speed among fine-tuning, LoRA, and our adapters, as outlined in Table IV. The training time and memory usage per iteration were measured using four RTX2080Ti GPUs, each with a batch size of 2 per GPU, and the tests were conducted using FP32. FLOPS and inference speed were gauged using one RTX2080Ti, with the input image size set to 448×448.

Subsequently, we opted for the pre-trained PlainViT-Base backbone network from SimpleClick and fine-tuned it on a portion of the new training dataset HQSeg44K to examine the performance of our methods on novel, specific data distributions. Fine-tuning was carried out on subsets comprising 500, 1000, 2000 samples, and the entire dataset. The evaluation took place on the test sets of COIFT [45] and DIS [46]. We utilized the boundary evaluation mbIoU as the accuracy metric to assess the model's performance, employing the corresponding NoC mbIoU. The results are presented in Table V.

Tables IV and V collectively illustrate the lightweight and efficient nature of our method. Notably, the training speed experienced an approximate 38% increase, accompanied by a noteworthy 24% reduction in memory usage. In terms of

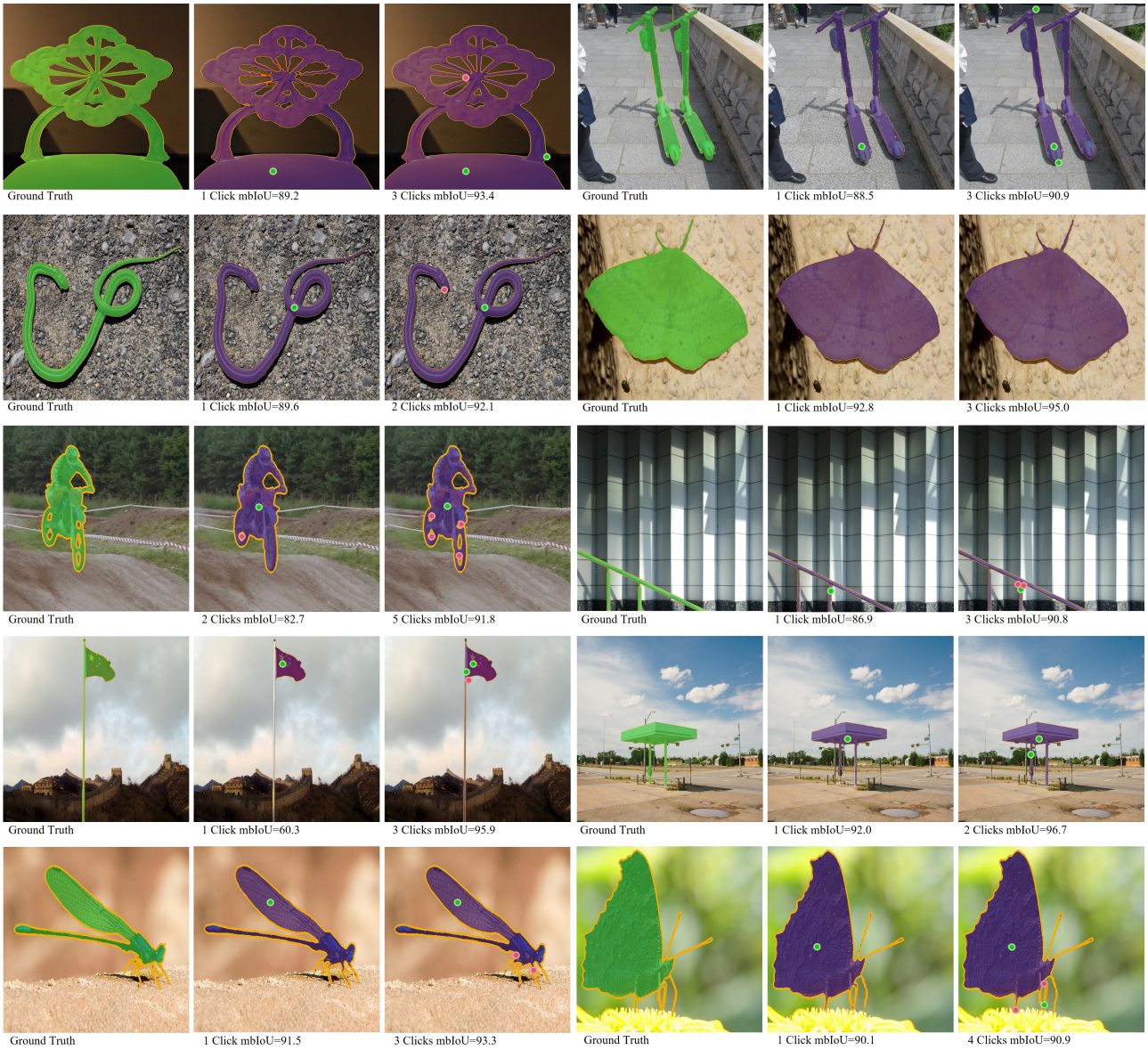


Fig. 6: Qualitative results on existing benchmarks. We evaluated the performance of our method using mbIoU. The green dots in the figure represent positive clicks, and the red dots represent negative clicks.

inference, there was a minimal increase in both memory usage and inference time. A comparative analysis of test results between fine-tuning and Adapter, using varying numbers of training samples, reveals that with an increase in the number of samples, the model's performance improves consistently, and the Adapter consistently outperforms fine-tuning and LoRA with the same backbone.

C. Ablation Study

To assess the effectiveness of each submodule in the proposed algorithm, ablation experiments were conducted. The pre-trained PlainViT-Base from SimpleClick on the COCO+LVIS training set was utilized as the backbone network. The evaluation was carried out on two datasets, DAVIS [34] and COIFT [45], with NoC IoU and RoF IoU (the ratio of failed instances) serving as the evaluation metrics. The results of the ablation experiments are presented in Table VI. We conducted three distinct sets of ablation experiments, each examining the impact of different factors: the influence of selecting ViT features from various layers as early-ViT features, the impact of early features and spatial prior features acting independently on performance, and an analysis of the feature integration process across different fusion methods.

Comparison of different selection of early-layer features Based on the experiments presented in the first section of Table VI, it is evident that the first stage of backbone networks demonstrates an advantage. However, the selection of different layers in stage one does not yield a significant impact. Therefore, as a standard practice, we decompose the ViT backbone network into four stages and opt for the output of the last layer of the first stage as the early-stage ViT features. For instance, we choose the fourth layer of PlainViT-Large or the third layer

TABLE VI: Ablation studies on the selection of early layers, impact of each submodule, and various fusion strategies. Baseline from SimpleClick with a PlainViT-Base backbone network.

	DAVIS			COIFT		
Method	NoC85	NoC90	RoF85	NoC85	NoC90	RoF85
SimpleClick-B	4.83	6.47	10.02%	2.88	3.76	8.93%
Stage0 Layer0	3.55	4.82	6.95%	1.93	3.09	8.57%
Stage0 Layer1	3.48	4.85	7.25%	1.96	3.00	3.21%
Stage0 Layer2	3.48	4.86	4.06%	1.86	2.88	2.14%
Stage1 Layer2	4.02	5.44	8.70%	2.28	3.06	8.57%
Stage2 Layer2	3.93	5.92	9.27%	2.20	3.24	8.21%
$\mathcal{F}_{ViT}^{early}$ Only	4.22	5.65	7.65%	2.17	3.49	5.71%
$\mathcal{F}_{spm}$ Only	3.81	5.11	6.56%	2.23	3.44	5.00%
Naive Add	3.57	4.85	6.37%	2.06	2.91	5.00%
Naive Cat	3.54	4.91	7.25%	2.00	3.17	8.57%
Cross Attn	3.63	4.96	7.83%	1.97	2.92	4.29%
Two-Way Attn	3.48	4.86	4.06%	1.86	2.88	2.14%

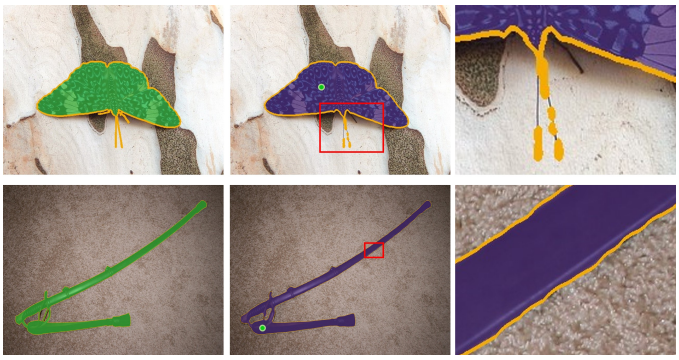


Fig. 7: Limitations of our method. The left side displays the ground truth. The middle illustrates our prediction results after a single click. The green dots represent positive clicks, while the red box marks the specific areas that have been zoomed in, and the details of these areas are presented on the right side of the figure.

of SegformerB3.

Enhancements from Spatial Prior and Early-ViT Features As depicted in Table VI, the incorporation of early-ViT features results in a significant improvement of 12.3% in overall performance. Furthermore, the integration of the proposed Spatial Prior Module leads to a notable improvement of 21.0%. Furthermore, the spatial extractor employing deformable convolution demonstrated a noteworthy improvement of 6%, surpassing the 5% performance boost achieved solely through the early-ViT features.

Comparison of different fusion method In contrast to the straightforward additive or concatenate feature fusion method, the bidirectional interactive attention layer significantly enhanced the performance of the details extractor by 4.5%.

D. Qualitative Results

Figure 6 shows the results of a comprehensive visual comparison of the performance of our method using mbIoU. The samples, sourced from DIS [46] dataset, DAVIS [34] dataset, and HRSOD [48] dataset, cover images featuring various complex structures taken from different environments. Notably, it can be observed that our algorithm generated significantly more accurate boundaries with the same number of interaction click points.

Limitations of our method are reported in Figure 7. To be specific, when processing low-resolution input images, our method predicted mask discontinuity during the segmentation of extremely thin structures such as insect legs and butterfly antennae. To address this problem, additional positive clicks are needed to ensure continuity in the segmented mask. Moreover, when the method is used to segment some instances with slender structures in input images with higher resolutions, the segmentation mask generated by a single click exhibited curvature at the edge of the instance, requiring additional click points for correction.

V. Conclusion

In this paper, we propose the ClickAdapter, an advanced interactive segmentation model that obtains higher-quality segmentation masks by attaching an adapter layer on top of the backbone network. The Adapter introduces a small number of network parameters that do not require pre-training. During network training, only the parameters of the adapter need to be trained, which greatly reduces the training consumption of the interactive segmentation model. By incorporating a detail extractor in the adapter, we infuse the spatial correlation and local features of the image into the network to achieve more refined segmentation results. The ClickAdapter achieved state-of-the-art interactive segmentation performance on existing benchmarks, significantly improving the quality of predicted masks through the incorporation of a detail extractor module. Additionally, we provide a detailed computational analysis of our method, highlighting its applicability as a tool for practical annotation.

VI. Acknowledgment

This work was partially supported by Shenzhen Science and Technology Program under Grant JCYJ20210324115604012, Grant JCYJ20220818103006012, and Grant ZDSYS20220606100601002); in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021B1515120008 and Grant 2023A1515011347; in part by Maxvision-AIRS-CUHK(SZ) Joint Laboratory of Inspection Robots; in part by Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[1] X. Chen, Z. Zhao, Y. Zhang, M. Duan, D. Qi, and H. Zhao, "Focalclick: Towards practical interactive image segmentation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1290–1299.

[2] Y. Hao, Y. Liu, Z. Wu, L. Han, Y. Chen, G. Chen, L. Chu, S. Tang, Z. Yu, Z. Chen, and B. Lai, "Edgeflow: Achieving practical interactive segmentation with edge-guided flow," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 1551–1560.

[3] K. Sofiiuk, I. Petrov, O. Barinova, and A. Konushin, "F-brs: Rethinking backpropagating refinement for interactive segmentation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8620–8629.

[4] Q. Liu, Z. Xu, G. Bertasius, and M. Niethammer, "Simpleclick: Interactive image segmentation with simple vision transformers," *arXiv preprint arXiv:2210.11006*, 2022.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[6] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[7] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[8] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 22–31.

[9] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," *arXiv preprint arXiv:2205.08534*, 2022.

[10] A. Ghiasi, H. Kazemi, E. Borgnia, S. Reich, M. Shu, M. Goldblum, A. G. Wilson, and T. Goldstein, "What do vision transformers learn? a visual exploration," *arXiv preprint arXiv:2212.06727*, 2022.

[11] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[12] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3129–3136.

[13] T. H. Kim, K. M. Lee, and S. U. Lee, "Nonparametric higher-order learning for interactive segmentation," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3201–3208.

[14] C. Rother, V. Kolmogorov, and A. Blake, """ grabcut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[15] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 373–381.

[16] Z. Lin, Z. Zhang, L.-Z. Chen, M.-M. Cheng, and S.-P. Lu, "Interactive image segmentation with first click attention," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 336–13 345.

[17] W.-D. Jang and C.-S. Kim, "Interactive image segmentation via backpropagating refinement scheme," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5292–5301.

[18] M. Forte, B. L. Price, S. Cohen, N. Xu, and F. Pitié, "Getting to 99% accuracy in interactive segmentation," *CoRR*, vol. abs/2003.07932, 2020. [Online]. Available: https://arxiv.org/abs/2003.07932

[19] K. Sofiiuk, I. A. Petrov, and A. Konushin, "Reviving iterative training with mask guidance for interactive segmentation," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 3141–3145.

[20] C. Zhang, C. Hu, Y. Liu, and X. He, "Intention-aware feature propagation network for interactive segmentation," *arXiv preprint arXiv:2203.05145*, 2022.

[21] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[23] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[24] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.

[25] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 280–296.

[26] A. C. Stickland and I. Murray, "Bert and pals: Projected attention layers for efficient adaptation in multi-task learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5986–5995.

[27] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[28] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 3, pp. 651–663, 2018.

[29] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," *Advances in neural information processing systems*, vol. 30, 2017.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[31] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, "Incorporating convolution designs into visual transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 579–588.

[32] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.

[33] K. McGuinness and N. E. Oconnor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[34] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 724–732.

[35] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *2011 international conference on computer vision*. IEEE, 2011, pp. 991–998.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[38] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5356–5364.

[39] L. Ke, M. Ye, M. Danelljan, Y. Liu, Y.-W. Tai, C.-K. Tang, and F. Yu, "Segment anything in high quality," 2023.

[40] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[41] N. Araslanov and S. Roth, "Single-stage semantic segmentation from image labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4253–4262.

[42] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[43] X. Chen, Z. Zhao, F. Yu, Y. Zhang, and M. Duan, "Conditional diffusion for interactive segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7325–7334.

[44] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1. IEEE, 2001, pp. 105–112.

[45] J. H. Liew, S. Cohen, B. Price, L. Mai, and J. Feng, "Deep interactive thin object selection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 305–314.

[46] X. Qin, H. Dai, X. Hu, D.-P. Fan, L. Shao, and L. Van Gool, "Highly accurate dichotomous image segmentation," in *European Conference on Computer Vision*. Springer, 2022, pp. 38–56.

[47] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[48] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu, "Towards high-resolution salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.