WebSAM-Adapter: Adapting Segment Anything Model for Web Page Segmentation

Bowen Ren[1], Zefeng Qian[1], Yuchen Sun[1], Chao Gao[3], and Chongyang Zhang[1,2(✉)]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{renbowen,zefeng_qian,sunyc22,sunny_zhang}@sjtu.edu.cn
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China
[3] China Pacific Insurance (Group) Co., Ltd., Shanghai 200010, China

Abstract. With the advancement of internet technology, web page segmentation, which aims to divide web pages into semantically coherent units, has become increasingly crucial for web-related applications. Conventional purely visual web page segmentation approaches, which depend on traditional edge detection, face challenges in generalizing across complex web pages. Recently, the Segment Anything Model (SAM) represents remarkable visual understanding and segmentation abilities. This inspires us that SAM can also demonstrate great potential in Web Page Segmentation. However, due to the lack of web-specific training data, its direct adaptation to web page segmentation domain has been hindered. To address this challenge, we propose WebSAM-Adapter, an effective adaptation of SAM, featuring a three-module architecture specifically tailored for web page segmentation with minimal additional trainable parameters. First, we propose a patch embedding tune module for adjusting the frozen patch embedding features, which is crucial for modifying the distribution of the original model. Second, an edge components tune module is designed to learn significant structural features within each web page. Finally, the outputs of these specialized modules are sent into our key Adapter module, which employs a lightweight multi-layer perceptron (MLP) to amalgamate these enriched features and generate webpage-specific knowledge. To the best of our knowledge, our method is the first successful adaptation of a large visual model like SAM to web page segmentation. Empirical evaluations on the comprehensive Webis-WebSeg-20 dataset demonstrate our model's state-of-the-art performance.

Keywords: Web Page Segmentation · Segment Anything Model · Adapter

1 Introduction

Web page segmentation is the process of identifying semantically coherent units, essentially serving as the inverse operation to web page layout [1]. This critical

N. Goharian et al. (Eds.): ECIR 2024, LNCS 14608, pp. 439–454, 2024.
https://doi.org/10.1007/978-3-031-56027-9_27

analytical procedure is foundational in web-related applications and is widely applied in information retrieval [2–4], software engineering [5–7] and assistive technology for visually impaired users [8,9].

Within the domain of web page segmentation, previous methods are predominantly categorized into three distinct approaches: DOM-based [10–13], Integrated Visual-DOM [14–18], and purely visual-based methods [19,20]. The DOM-based methods [10–13] exclusively utilize the Document Object Model (DOM) tree of a web page's source code for segmenting and categorizing web elements. While these methods offer structural precision, the tree structure and the relationships between nodes don't always align with the user's visual perception of the rendered page.

Differently, Integrated Visual-DOM methods [14–18] aim for richer segmentation by integrating DOM attributes with visual cues of web pages. A prime example is the Visual Page Segmentation (VIPS) [14] algorithm. However, the dependency on both visual and structural elements can limit their applicability, especially when access to source code is restricted. Conversely, purely visual based methods, such as those by Cao et al. [19] and Cormier et al. [20], focusing only on the rendered web pages and eliminating the need for source code. Although these approaches represent advancements through traditional computer vision edge detection, they face difficulties in generalizing across complex web pages, as illustrated in Fig. 1. This underscores the compelling need for further exploration of purely visual segmentation methods that leverage only the rendered web page, ensuring enhanced robustness and comprehensive applicability.

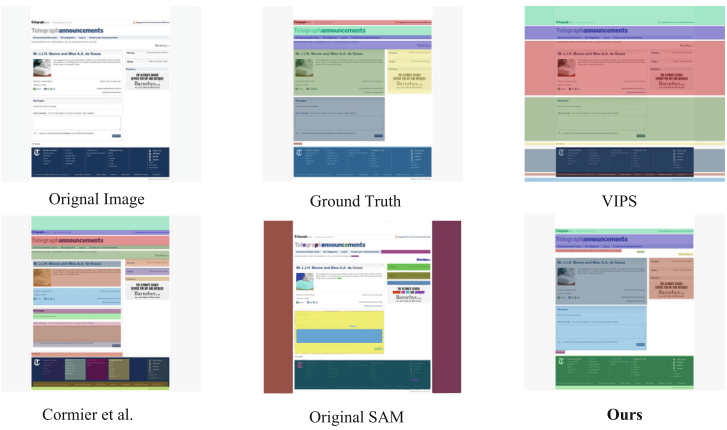

Fig. 1. A comparison of several segmentation methods. In dealing with a complex web page, VIPS [14], Cormier et al. [20], and Original SAM [21] face difficulties in performing accurate and effective segmentation. In contrast, our WebSAM-Adapter is capable of dividing a web page into more precise regions.

To address the aforementioned challenges, particularly the intricate structures and diverse styles unique to web pages compared to natural images, we think that models endowed with enhanced visual understanding and segmentation capabilities are better equipped to tackle these issues. The recent Segment Anything Model (SAM) [21] has demonstrated remarkable visual understanding and segmentation abilities, achieving outstanding results when adapted to various vision segmentation tasks [22–26]. However, despite SAM's claim to "Segment Anything" and its impressive achievements in natural images, our empirical evaluations indicate that its direct application to web page segmentation is ineffective, as illustrated in Fig. 1. The primary reason for SAM's ineffectiveness in segmenting web pages is attributed to the lack of training data specific to web usage. To this end, this paper delves further into SAM to mitigate its limitations in web page segmentation and accordingly propose WebSAM-Adapter.

Building on this, our key insight is to leverage individual image features for webpage-specific knowledge, utilizing the comprehensive semantic understanding inherent in the foundational SAM model. The WebSAM-Adapter is crafted with a three-module architecture, specifically tailored for web page segmentation. The functions of our model can be concisely summarized as:

1. **Aligning Data Distribution through Patch Embedding Tune**: This module refines the alignment between pre-trained and target datasets by tuning the patch embedding features, thereby enhancing model generalization.
2. **Learning Structural Features through Edge Components Tune**: This module identifies and leverages significant structural features in web pages, enriching the model's representation and capturing layout information traditionally sourced from the DOM tree.
3. **Fusing Enriched Features through Adapter**: This module amalgamates the enriched outputs from the Patch Embedding Tune and Edge Components Tune modules using a lightweight MLP. It serves as an adaptable auxiliary network, infusing webpage-specific knowledge from web page samples into the foundational SAM with minimal additional trainable parameters.

To validate our approach, we conduct experiments on the Webis-WebSeg-20 dataset [1], the most comprehensive dataset currently available for this domain. Our results demonstrate that the WebSAM-Adapter not only mitigates SAM's limitations but also achieves state-of-the-art performance.

In summary, our main contributions are as follows:

1. We propose WebSAM-Adapter, an effective web page segmentation method that leverages a purely visual approach to overcome the limitations of existing techniques relying on the DOM tree or traditional computer vision edge detection methods.
2. To the best of our knowledge, we are the first to propose an adaptation strategy for web page segmentation that acquires webpage-specific knowledge through image embeddings and edge components, with minimal additional trainable parameters.
3. Our approach achieves state-of-the-art performance on the most comprehensive dataset currently available for web page segmentation tasks.

2 Related Work

2.1 Web Page Segmentation

Web page segmentation techniques can be broadly categorized into three approaches: DOM-based, Integrated Visual-DOM, and purely visual-based methods. DOM-based methods exploit the Document Object Model (DOM) to segment and categorize web page elements [10–13]. These methods necessitate the source code of the web page and specifically employ DOM attributes, tags, and subtree structures for segmentation and analysis. While offering a structured paradigm, these methods focus predominantly on the DOM's tree structure, limiting their broader applicability by failing to account for the visual content that users actually perceive on-screen.

Another category, Integrated Visual-DOM segmentation techniques [14–18], incorporates visual attributes for a more contextually rich segmentation. Notably, these methods adopt a hybrid approach that combines both DOM attributes and visual cues. For instance, Cai et al. [14] introduce the Visual Page Segmentation (VIPS) algorithm, which merges DOM structure with visual cues to form a hierarchical content model. Similarly, Bajammal et al. [16] present Cortex, which also integrates DOM attributes with visual elements to achieve superior segmentation precision.

Conversely, a few algorithms [19,20] exclusively depend on the rendered web page, eliminating the need for source code and providing a purely visual strategy for web page segmentation. Cao et al. [19] employ edge detection algorithms such as Canny in the preprocessing stage, subsequently shrinking and splitting the web page image into visually congruent sub-images. In contrast, Cormier et al. [20] utilize a Bayesian framework with Sobel operators for edge detection to achieve the most probable segmentation. However, while these approaches showcase advancements in web page segmentation through traditional computer vision edge detection methods, they face limitations in generalizing across complex web pages. Therefore, the creation of a universally applicable, visually-driven web page segmentation algorithm is essential. The emergence of large models like SAM offers a promising direction for overcoming this challenge, harnessing the power of deep learning for improved adaptability and performance.

2.2 Segment Anything Model

Recently, Meta AI Research introduced the Segment Anything Model (SAM) [21], which has received considerable academic attention. Designed as a foundational model for image segmentation, SAM frequently rivals or even outperforms fully supervised methods in zero-shot scenarios. Its robust generalization capabilities are highlighted by its training on a vast dataset SA-1B [21], encompassing over 11 million natural images and their corresponding 1 billion masks.

The model has shown an impressive range of applicability, extending beyond generic image segmentation to specialized tasks. These include medical image segmentation [22–24], remote sensing instance segmentation [25], and camouflage image segmentation [26,27].

2.3 Adapter-Based Fine-Tuning

The Adapter methodology is inspired by Parameter-Efficient Fine-Tuning (PEFT) techniques prevalent in Natural Language Processing [28]. Unlike traditional full fine-tuning methods, the Adapter approach [29–32] incorporates parameter-efficient modules into the original model while keeping the majority of the parameters frozen. This facilitates efficient learning and faster updates, thereby addressing computational and memory constraints in large-scale model fine-tuning. Liu et al. [29] introduce the Explicit Visual Prompting (EVP) methodology for low-level structure segmentation, which integrates explicit visual cues into the adapter module to get task-specific information and outperforms existing tuning protocols.

Recent studies have extended Adapter methodologies to SAM, targeting specialized segmentation challenges. Wu et al. [23] introduced the Medical SAM Adapter (MSA), which augments SAM's capabilities in medical image segmentation by incorporating domain-specific knowledge through Adapter modules. Similarly, Chen et al. [26] developed the SAM-Adapter, enhancing SAM's performance in specialized tasks like shadow and camouflaged object detection through domain-specific visual prompts. In this paper, we introduce WebSAM-Adapter, an effective approach that capitalizes on the strengths of existing Adapter methodologies to address the unique challenges of web page segmentation.

3 Method

3.1 The Preliminary of Segment Anything Model

The objective of the WebSAM-Adapter is to capitalize on the knowledge acquired from the SAM architecture. To begin with, we provide an overview of the SAM architecture before discussing our modifications. SAM is composed of three principal components: a large-scale image encoder, a prompt encoder, and a lightweight mask decoder. This framework allows for different masks to be generated for the same image based on different prompts. The image encoder employs a pre-trained Visual Transformer (ViT) [33] to process high-resolution inputs, subsequently outputting feature maps at a scale reduced to 1/16 of the original image. The prompt encoder supports a variety of input types, including both sparse (points, boxes, text) and dense (masks). The mask decoder is a specialized Transformer decoder block, equipped with a dynamic mask prediction head.

3.2 The Architecture of WebSAM-Adapter

In this section, we introduce the WebSAM-Adapter, a specialized framework designed to address the challenge of web page segmentation. Inspired by preceding Adapter-based methodologies and tailored for the unique demands of web page segmentation, the WebSAM-Adapter represents a strategic confluence of existing methods and specific web page segmentation requirements. Built upon the foundational elements of the Segment Anything Model (SAM), the architecture incorporates three modules that are purposefully designed for web page segmentation tasks. Our key insight is to leverage individual image features from both patch embeddings and edge components for webpage-specific knowledge, a process facilitated by an Adapter module. The model is lightweight, adaptable to limited data, and serves as an auxiliary network, infusing webpage-specific knowledge from web page segmentation samples into the foundational SAM with minimal additional trainable parameters.

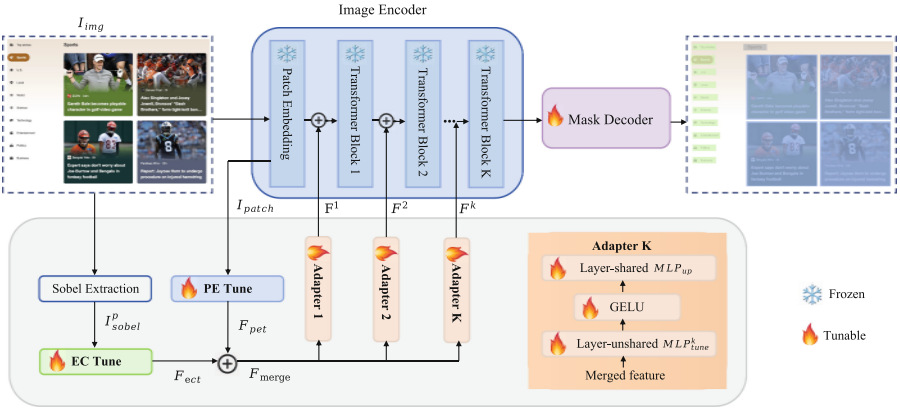

Fig. 2. The architecture of the proposed WebSAM-adapter.The Patch Embedding Tune (PE Tune) and the Edge Component Tuning (EC Tune) are utilized to refine the extracted features. The Adapter is meticulously designed to amalgamate these features, facilitating the acquisition of webpage-specific knowledge.

Framework Overview. As illustrated in Fig. 2, our image encoder, with its weights frozen from the SAM pre-trained model, incorporates three specialized modules designed for web page segmentation: Patch Embedding Tuning (PC Tune), Edge Component Tuning (EC Tune), and an Adapter module. The Patch Embedding Tune module adjusts the frozen patch embedding features, crucial for modifying the distribution of the original model. Next, the Edge Components Tune module is designed to learn significant structural features within each web page. These specialized modules then feed their outputs into our key Adapter module, employing a lightweight multi-layer perceptron (MLP) to amalgamate these enriched features and generate webpage-specific knowledge. Furthermore,

the mask decoder of SAM is employed and initialized with weights from a pre-trained SAM model, which we fine-tune during the training phase. Considering that real-world applications of web page segmentation do not necessitate interactive prompts, we have intentionally excluded the use of a prompt encoder and refrained from providing prompts to SAM's original mask decoder.

Patch Embedding Tune. In the pre-trained SAM image encoder [21], the patch embedding layer partitions each 1024×1024 input image I_{img} into a 64×64 grid of non-overlapping patches. These patches are then transformed into embeddings I_{patch} with dimensions $[B, P_{\mathrm{H}}, P_{\mathrm{W}}, E]$, where B represents the batch size, P_H and P_W are the counts of patches along the height and width, respectively, and E is the embedding dimension of 768 for *vit-b*. While this approach is effective for natural images, it falls short when applied to web page images, which often exhibit unique structural and semantic complexities.

To address this limitation, we introduce the Patch Embedding Tune module, which is positioned subsequent to SAM's patch embedding layer. This module serves two critical functions. First, it amplifies SAM's semantic understanding of web page images, aligning it more closely with the task-specific requirements of web page segmentation. Second, it refines the input patch encodings to be fed into subsequent layers, including the Adapter module, which further leverages these enriched embeddings for fine-tuning.

The Patch Embedding Tune module utilizes a tunable linear layer L_{pet} to project the original image embeddings I_{patch} from an E-dimensional space into a feature space $F_{pet} \in \mathbb{R}^e$ with reduced dimensionality e. The extent of this dimensionality reduction can be adjusted by a scale factor μ, which is employed to regulate the number of learnable parameters.

$$F_{\mathrm{pet}} = L_{\mathrm{pet}}(I_{\mathrm{patch}}), \text{with } e = \frac{E}{\mu} \tag{1}$$

Edge Components Tune. The unique layout of web pages, predominantly composed of rectangular blocks and complex images, necessitates a tailored approach in segmentation algorithms. Recognizing this, our Edge Components Tune module is specifically designed to leverage these distinctive structural characteristics inherent to web pages [20]. Traditionally, such features are extracted from the DOM tree, which plays a pivotal role in enhancing the adaptability and performance of our model in web page segmentation tasks.

Initially, the input image I_{img} is preprocessed through inverse normalization and converted to grayscale I_{gray}. We then apply the Sobel operator to compute the gradient magnitude, denoted as I_{sobel}, using the following unified equation:

$$I_{\mathrm{sobel}} = \sqrt{\left(\mathrm{Sobel}_x * I_{\mathrm{gray}}\right)^2 + \left(\mathrm{Sobel}_y * I_{\mathrm{gray}}\right)^2} \tag{2}$$

Here, Sobel_x and Sobel_y are the Sobel kernels for the x and y directions, respectively, and $*$ denotes the convolution operation.

The computed Sobel features I_{sobel} are then partitioned into non-overlapping patches I^p_{sobel}, in alignment with the SAM model [21]. These patches are projected into an e-dimensional feature space $F_{ect} \in \mathbb{R}^e$ using a tunable linear layer L_{ect}.

$$F_{\text{ect}} = L_{\text{ect}}(I^p_{sobel}) \tag{3}$$

Importantly, the edge features captured by this module can be changed to emphasize key features in other domains, thereby extending its applicability beyond web page segmentation tasks.

Adapter. The Adapter module serves as an integral component within the WebSAM-Adapter framework, engineered to synthesize webpage-specific knowledge by fusing features from both the Patch Embedding Tune and Edge Components Tune modules. It fulfills two primary objectives: enhancing the model's generalization capabilities by aligning image embeddings with web page structures, and augmenting the feature space tailored for web page segmentation tasks.

To achieve these goals, the Adapter employs a lightweight multi-layer perceptron (MLP) to generate webpage-specific knowledge. For the k_{th} Adapter, the inputs F_{pet} and F_{ect} are directly added together to form F_{merge}, which is then utilized to produce webpage-specific knowledge, denoted as F^k.

$$F^k = \text{MLP}_{up}(\text{GELU}(\text{MLP}^k_{tune}(F_{merge})), with\ F_{merge} = F_{pet} + F_{ect} \tag{4}$$

In this configuration, GELU [34] serves as the the activation function. MLP^k_{tune} is a linear layer responsible for generating different webpage-specific knowledge, while MLP_{up} is an up-projection layer designed to harmonize the dimensions of the transformer features across all Adapters. Finally, F^k represents the webpage-specific knowledge output integrated into each transformer block. Notably, k corresponds to the number of transformer blocks in the SAM image encoder, with $k = 12$ for SAM's base model.

4 Experiments

4.1 Datasets

Although the task of web page segmentation has been proposed for nearly two decades, its foundational resources, particularly datasets and evaluation methods, have been fragmented and often incompatible with each other. Prior datasets were frequently tailored to specific downstream tasks, leading to a proliferation of datasets with varying standards and metrics, most of which are not publicly available or standardized.

The Webis-WebSeg-20 dataset, introduced by Kiesel et al. [1], emerges as a pivotal solution and provides a more comprehensive basis for evaluation. This dataset comprises 42,450 crowdsourced segmentations across 8,490 web pages.

This includes consensus annotations from five human annotators, web archival files for historical rendering, HTML DOM for partial rendering, visual screenshots, and coordinate-mapped DOM nodes. The richness and diversity of this dataset make it an ideal choice for our research, enabling a more thorough and nuanced analysis of web page segmentation.

To more effectively evaluate our model's performance, we employed two distinct types of evaluation metrics. The primary set comprises three metrics: P_{B^3}, R_{B^3} and $F^*_{B^3}$, proposed by Kiesel et al. [1]. These metrics are adaptations of the extended BCubed measures derived from clustering theory [35]. Focusing on these metrics, P_{B^3} is calculated by dividing the elements co-segmented in both the ground-truth and algorithm-generated segmentations by the algorithm-segmented elements. R_{B^3} follows a similar approach but considers the total elements in the ground-truth segmentation as the denominator. F_{B^3} is the harmonic mean of P_{B^3} and R_{B^3} for individual pages, while $F^*_{B^3}$ is the harmonic mean of their averages across all pages. For this evaluation, we conducted experiments on each of the five types of atomic elements defined by Kiesel et al. [1]: all pixels (*pixels*), all pixels at visual edges in both coarse ($edges_C$) and fine settings ($edges_F$), all visible DOM nodes (*nodes*), and all textual characters (*chars*). The second evaluation approach utilizes F_1 score and *Pixel Accuracy* to evaluate the performance.

4.2 Implementation Details

Our method is implemented in PyTorch and runs on a single NVIDIA RTX 3090 GPU with 24 GB of memory. Due to memory constraints, we focus solely on fine-tuning the base model of SAM in this study. We employ the AdamW optimizer [36], starting with an initial learning rate of 2e-4, and train the model for a total of 20 epochs. Cosine decay is applied to adjust the learning rate during the training process. The images are resized to a resolution of 1024×1024 pixels. The loss function that supervises the mask predictions combines Binary Cross-Entropy (BCE) loss and Intersection Over Union (IOU) loss in a linear fashion.

Table 1. Comparison of six segmentation algorithms, highlighting the type of input documents they are designed for, the features they utilize, and the format of their output segmentation.

Method	Document	Features	Output
VIPS [14]	Web page	DOM tree, style, location	Rectangle tree
Cormier et al. [20]	Web page	Screenshot	Rectangle tree
HEPS [18]	Web page	DOM tree, style	Node set
MMDetection [37]	Photo	Screenshot	Pixel masks
Kiesel. [38]	Web page	DOM tree, style, location	Rectangle tree
Ours	Web page	Screenshot	Pixel masks

4.3 Main Results

In our experiments, we aim to validate the effectiveness of our method by comparing it against the benchmarks established by the empirical study conducted by Kiesel et al. [38]. This study, which evaluates five different web page segmentation algorithms across various paradigms, serves as a valuable reference for comparison in our research. To ensure a fair comparison, we strictly adhere to the testing methods outlined in Kiesel et al.'s study. We employ standard 10-fold cross-validation and utilize the 10-fold partitioning method they provided. For the sake of academic rigor, we neither modify nor re-implement the algorithms from Kiesel et al. [38], relying solely on their reported results for comparative analysis.

Table 2. The evaluation results for six algorithms on the Webis-WebSeg-20 dataset [1]: VIPS [14], Cormier et al. [20], HEPS [18], MMDetection [37], Kiesel et al. [38], and our WebSAM-Adapter. Evaluation metrics are the average F_1-score (F_{B^3}), precision (P_{B^3}), recall (R_{B^3}), and the harmonic mean of the averaged precision and recall ($F^*_{B^3}$) for each type of atomic element. The ground truth contains an average of 9.1 segments. The highest score in each row is highlighted in bold.

Measure		VIPS	Corm.	HEPS	MMD.	Kiesel.	ours
Segments		16.1	15.3	36.1	23.0	18.7	**8.9**
pixels	F_{B^3}	0.38	0.36	0.33	0.42	0.39	**0.62**
	$F^*_{B^3}$	0.47	0.53	0.44	0.54	0.50	**0.70**
	P_{B^3}	0.36	0.39	0.36	0.51	0.38	**0.63**
	R_{B^3}	0.67	**0.80**	0.56	0.57	0.72	0.79
$edges_F$	F_{B^3}	0.59	0.51	0.48	0.53	0.56	**0.71**
	$F^*_{B^3}$	0.68	0.65	0.58	0.61	0.66	**0.76**
	P_{B^3}	0.66	0.55	0.61	**0.73**	0.61	0.69
	R_{B^3}	0.69	0.80	0.55	0.53	0.71	**0.85**
$edges_C$	F_{B^3}	0.61	0.53	0.49	0.54	0.57	**0.72**
	$F^*_{B^3}$	0.68	0.66	0.59	0.62	0.67	**0.77**
	P_{B^3}	0.67	0.56	0.62	**0.74**	0.63	0.69
	R_{B^3}	0.70	0.80	0.56	0.53	0.72	**0.86**
nodes	F_{B^3}	0.63	0.52	0.43	0.52	0.54	**0.69**
	$F^*_{B^3}$	0.70	0.65	0.54	0.61	0.65	**0.75**
	P_{B^3}	0.69	0.53	0.63	**0.74**	0.64	**0.74**
	R_{B^3}	0.71	**0.82**	0.46	0.51	0.65	0.77
chars	F_{B^3}	0.67	0.61	0.50	0.61	0.62	**0.77**
	$F^*_{B^3}$	0.75	0.71	0.60	0.69	0.71	**0.82**
	P_{B^3}	0.77	0.61	0.73	**0.79**	0.72	0.78
	R_{B^3}	0.72	0.84	0.51	0.60	0.71	**0.87**

Table 1 provides an overview of the algorithms used in the experiments, and Table 2 displays the performance of each algorithm on the Webis-WebSeg-20 dataset [1]. Our algorithm outperforms all the algorithms in the crucial comprehensive indicators F_{B^3} and $F^*_{B^3}$ across the five types of atomic elements, demonstrating an average increase of 18.2%. However, Our P_{B^3} is marginally lower than the MMDetection [37] method in the three atomic elements $edges_F$, $edges_C$, and *chars*. This is attributed to MMDetection's propensity to detect more smaller segments, which results in a higher P_{B^3} but a significantly lower R_{B^3} compared to ours. Similarly, our R_{B^3} is marginally lower than the Cormier [20] method in the two atomic elements *pixels* and *nodes*. This algorithm tends to detect larger areas, resulting in a higher R_{B^3}, but its P_{B^3} is notably lower. Overall, our method achieves competitive results. A visual comparison of the results is represented in the Fig. 3. This visual evidence not only supports but also strengthens our claim of the method's efficacy.

Simultaneously, our semantic segmentation results achieved an F_1 score of 0.892 and a Pixel Accuracy of 0.881, underscoring the strong performance of our model.

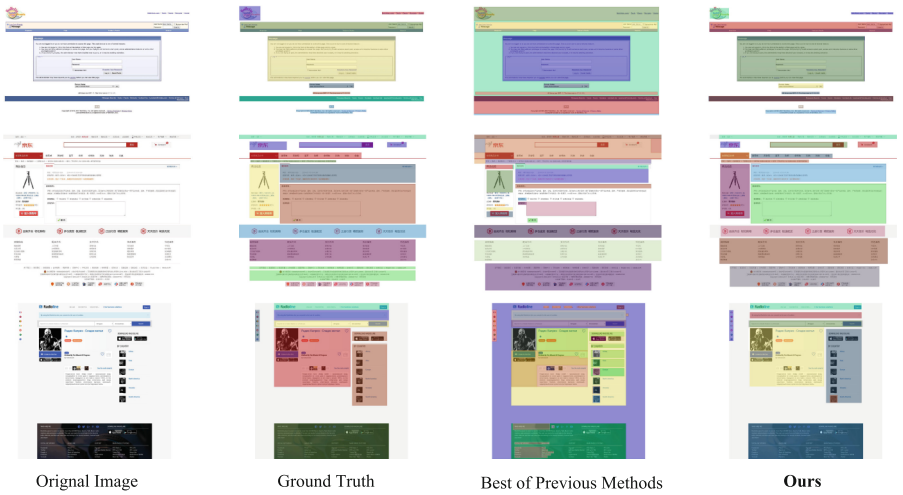

Fig. 3. Comparison of Original Image, Ground Truth, Best of Previous Methods, and Our Result, across three web page images from the dataset. The column "Best of Previous Methods" showcases the most effective result among the five algorithms previously evaluated, as referenced in Table 2. The clear differentiation in segmentation quality among these columns visually underscores the superior performance of our proposed method.

4.4 Ablation Study

We conduct an ablation study on the Webis-WebSeg-20 dataset [1] to demonstrate the effectiveness of each component. Unless specified otherwise, the experiments are performed with the scaling factor $\mu = 8$.

Table 3. Comparison with efficient tuning approaches.

Method	Trainable Param.(M)	F_1 score	Pixel Accuracy
Only Decoder	4.06	0.833	0.798
MedSAM-Adapter [23]	4.67	0.883	0.865
AdaptFormer [31]	4.36	0.879	0.862
Ours	**4.34**	**0.892**	**0.881**

Comparison with Efficient Tuning Approaches. We evaluate our method by comparing it to the approach of exclusively tuning the decoder, a strategy widely used for downstream task adaptation, considering that the total parameter size of SAM's base model is 93.7M. Additionally, We also compare it with similar adaptation methods in image classification, namely AdaptFormer [31], and in medical image segmentation, specifically MedSAM-Adapter [23]. The middle dimension for AdaptFormer and hidden features for MedSAM-Adapter are set to 16 to ensure a fair comparison in terms of tunable parameters. As observed from Table 3, there is a significant drop in performance when only the decoder is tuned. The integration of additional MLPs in the Transformer block [23,31] also improves performance compared to the method of solely tuning the decoder. We set the hyper-parameter $\mu = 8$, which is used to control the number of parameters of the Adapter, as described in Eq. 1. Importantly, in our WebSAM-Adapter, we fine-tuned merely 4.34 M parameters, representing only 4.6% of the entire model, ensuring parameter comparability with other methods for a balanced evaluation. The results in Table 3 demonstrate that our method outperforms others.

Architecture Design. To validate the effectiveness of our proposed WebSAM-Adapter architecture, we modify it into different variants. As indicated in Table 4, the application of shared MLP_{tune} in various Adapters preserves only a limited number of parameters and results in a substantial decline in performance. The incorporation of different MLP_{up} in diverse Adapters does not guarantee consistent improvement and, furthermore, increases the number of parameters by 19%. In contrast, the omission of Patch Embedding Tune (PET) or Edge Components Tune (ECT) leads to a decrease in performance, underscoring their significance in visual information. The proposed WebSAM-Adapter (Decoder + PET + ECT + Adapter) exhibits enhanced efficacy.

Scale Factor μ. In Sect. 3.2 of the main paper, we introduce μ to regulate the number of learnable parameters in Eq. 1. A larger μ yields fewer parameters

Table 4. Ablation on the architecture designs described in Fig. 2.

Method	Trainable Param.(M)	F_1 score	Pixel Accuracy
Only Decoder	4.06	0.833	0.798
Ours w/o ECT	4.31	0.871	0.850
Ours w/o PET	4.27	0.872	0.852
Ours w/Shared MLP_{tune}	4.24	0.866	0.842
Ours w/Unshared MLP_{up}	5.16	0.875	0.857
Ours	**4.34**	**0.892**	**0.881**

available for tuning. As depicted in Table 5, the performance improves as μ decreases from 64 to 8. However, when μ is further reduced to 2 or 1, there is no sustained significant improvement in performance, despite the increase in model size. This observation suggests that $\mu = 8$ represents an optimal choice for balancing performance with model size.

Table 5. Ablation on the parameter scale factor μ.

scale factor μ	Trainable Param.(M)	F_1 score	Pixel Accuracy
1	12.5	**0.893**	**0.884**
2	6.52	0.892	0.884
4	4.85	0.892	0.881
8	4.34	0.892	0.881
16	4.17	0.874	0.871
32	4.11	0.867	0.863
64	4.08	0.869	0.849

5 Conclusion

In conclusion, this study presents the application of the WebSAM-Adapter, an effective implementation of SAM, to address the specific challenges of web page segmentation. Utilizing a tailored three-module architecture, it effectively acquires webpage-specific knowledge, demonstrating state-of-the-art performance on the Webis-WebSeg-20 dataset [1]. This research not only mitigates SAM's limitations in this domain but also lays the groundwork for future innovations in purely visual web page segmentation methods.

Acknowledgements. This work was financially supported by the Joint Research Fund of China Pacific Insurance (Group) Co. and SJTU-Artificial Intelligence Institute.

References

1. Kiesel, J., Kneist, F., Meyer, L., Komlossy, K., Stein, B., Potthast, M.: Web page segmentation revisited: evaluation framework and dataset. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 3047–3054 (2020)
2. Cai, D., He, X., Wen, J.-R., Ma, W.-Y.: Block-level link analysis. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 440–447 (2004)
3. Bing, L., Guo, R., Lam, W., Niu, Z.-Y., Wang, H.: Web page segmentation with structured prediction and its application in web page classification. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 767–776 (2014)
4. Akpinar, M.E., Yesilada, Y.: Vision based page segmentation algorithm: extended and perceived success. In: Sheng, Q.Z., Kjeldskov, J. (eds.) ICWE 2013. LNCS, vol. 8295, pp. 238–252. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-04244-2_22
5. Saar, T., Dumas, M., Kaljuve, M., Semenenko, N.: Browserbite: cross-browser testing via image processing. Softw. Pract. Exp. **46**(11), 1459–1477 (2016)
6. Mahajan, S., Abolhassani, N., McMinn, P., Halfond, W.G.: Automated repair of mobile friendly problems in web pages. In: Proceedings of the 40th International Conference on Software Engineering, pp. 140–150 (2018)
7. Geng, G.-G., Lee, X.-D., Zhang, Y.-M.: Combating phishing attacks via brand identity and authorization features. Secur. Commun. Netw. **8**(6), 888–898 (2015)
8. Cormier, M., Cohen, R., Mann, R., Rahim, K., Wang, D.: A robust vision-based framework for screen readers. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8927, pp. 555–569. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16199-0_39
9. Cormier, M., Moffatt, K., Cohen, R., Mann, R.: Purely vision-based segmentation of web pages for assistive technology. Comput. Vis. Image Underst. **148**, 46–66 (2016)
10. Sanoja, A., Gançarski, S.: Block-o-matic: a web page segmentation framework. In: 2014 International Conference on Multimedia Computing and Systems (ICMCS), pp. 595–600. IEEE (2014)
11. Vineel, G.: Web page dom node characterization and its application to page segmentation. In: 2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA), pp. 1–6. IEEE (2009)
12. Chen, Y., Ma, W.-Y., Zhang, H.-J.: Detecting web page structure for adaptive viewing on small form factor devices. In: Proceedings of the 12th International Conference on World Wide Web, pp. 225–233 (2003)
13. Rajkumar, K., Kalaivani, V.: Dynamic web page segmentation based on detecting reappearance and layout of tag patterns for small screen devices. In: 2012 International Conference on Recent Trends in Information Technology, pp. 508–513. IEEE (2012)
14. Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: Vips: a vision-based page segmentation algorithm (2003)
15. Zeleny, J., Burget, R., Zendulka, J.: Box clustering segmentation: a new method for vision-based web page preprocessing. Inf. Process. Manag. **53**(3), 735–750 (2017)
16. Bajammal, M., Mesbah, A.: Page segmentation using visual adjacency analysis. arXiv preprint arXiv:2112.11975 (2021)

17. Andrew, J., Ferrari, S., Maurel, F., Dias, G., Giguet, E.: Web page segmentation for non visual skimming. In: The 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33) (2019)
18. Manabe, T., Tajima, K.: Extracting logical hierarchical structure of html documents based on headings. In: Proceedings of the VLDB Endowment, pp. 1606–1617 (2015). http://dx.doi.org/10.14778/2824032.2824058
19. Cao, J., Mao, B., Luo, J.: A segmentation method for web page analysis using shrinking and dividing. Int. J. Parallel Emergent Distrib. Syst. **25**(2), 93–104 (2010)
20. Cormer, M., Mann, R., Moffatt, K., Cohen, R.: Towards an improved vision-based web page segmentation algorithm. In: 2017 14th Conference on Computer and Robot Vision (CRV), pp. 345–352. IEEE (2017)
21. Kirillov, A., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
22. Ma, J., Wang, B.: Segment anything in medical images. arXiv preprint arXiv:2304.12306 (2023)
23. Wu, J., et al.: Medical sam adapter: adapting segment anything model for medical image segmentation. arXiv preprint arXiv:2304.12620 (2023)
24. Shaharabany, T., Dahan, A., Giryes, R., Wolf, L.: Autosam: adapting sam to medical images by overloading the prompt encoder. arXiv preprint arXiv:2306.06370 (2023)
25. Chen, K., et al.: Rsprompter: learning to prompt for remote sensing instance segmentation based on visual foundation model. arXiv preprint arXiv:2306.16269 (2023)
26. Chen, T., et al.: Sam fails to segment anything?-sam-adapter: adapting sam in underperformed scenes: Camouflage, shadow, and more. arXiv preprint arXiv:2304.09148 (2023)
27. Tang, L., Xiao, H., Li, B.: Can sam segment anything? when sam meets camouflaged object detection. arXiv preprint arXiv:2304.04709 (2023)
28. Zaken, E.B., Ravfogel, S., Goldberg, Y.: Bitfit: simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199 (2021)
29. Liu, W., Shen, X., Pun, C.-M., Cun, X.: Explicit visual prompting for low-level structure segmentations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19 434–19 445 (2023)
30. He, X., Li, C., Zhang, P., Yang, J., Wang, X.E.: Parameter-efficient model adaptation for vision transformers. arXiv preprint arXiv:2203.16329 (2022)
31. Chen, S., et al.: Adaptformer: adapting vision transformers for scalable visual recognition. Adv. Neural Inf. Process. Syst. **35**, 16 664–16 678 (2022)
32. Chen, Z., et al.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022)
33. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
34. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). Cornell University - arXiv (2016)
35. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Inf. Retr. 461–486 (2009). https://doi.org/10.1007/s10791-008-9066-8
36. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

37. Chen, K., et al.: Mmdetection: open mmlab detection toolbox and benchmark. arXiv Computer Vision and Pattern Recognition (2019)
38. Kiesel, J., Meyer, L., Kneist, F., Stein, B., Potthast, M.: An empirical comparison of web page segmentation algorithms. In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021. LNCS, vol. 12657, pp. 62–74. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72240-1_5