Evaluation and improvement of Segment Anything Model for interactive histopathology image segmentation

SeungKyu Kim, Hyun-Jic Oh, Seonghui Min and Won-Ki Jeong⋆

Korea University, College of Informatics,
Department of Computer Science and Engineering
`wkjeong@korea.ac.kr`

Abstract. With the emergence of the Segment Anything Model (SAM) as a foundational model for image segmentation, its application has been extensively studied across various domains, including the medical field. However, its potential in the context of histopathology data, specifically in region segmentation, has received relatively limited attention. In this paper, we evaluate SAM's performance in zero-shot and fine-tuned scenarios on histopathology data, with a focus on interactive segmentation. Additionally, we compare SAM with other state-of-the-art interactive models to assess its practical potential and evaluate its generalization capability with domain adaptability. In the experimental results, SAM exhibits a weakness in segmentation performance compared to other models while demonstrating relative strengths in terms of inference time and generalization capability. To improve SAM's limited local refinement ability and to enhance prompt stability while preserving its core strengths, we propose a modification of SAM's decoder. The experimental results suggest that the proposed modification is effective to make SAM useful for interactive histology image segmentation. The code is available at `https://github.com/hvcl/SAM_Interactive_Histopathology`

Keywords: Segment Anything Model · Histopathology Image Anaysis · Interactive Segmentation · Foundation Models.

1 Introduction

Tumor region segmentation in whole slide images (WSIs) is a critical task in digital pathology diagnosis. Numerous segmentation methods have been developed in the computer vision community that perform well on objects with clear edges [1, 2, 9]. However, as shown in Fig. 1, tumor boundaries in histopathology images are often indistinct and ambiguous. As a result, directly applying conventional image segmentation methods to WSIs tends to yield unsatisfactory results. In recent years, deep learning advancements have shown promising outcomes in medical image segmentation [21] when sufficient training labels are available. Nevertheless, even fully-supervised models have room for improvement due to

---

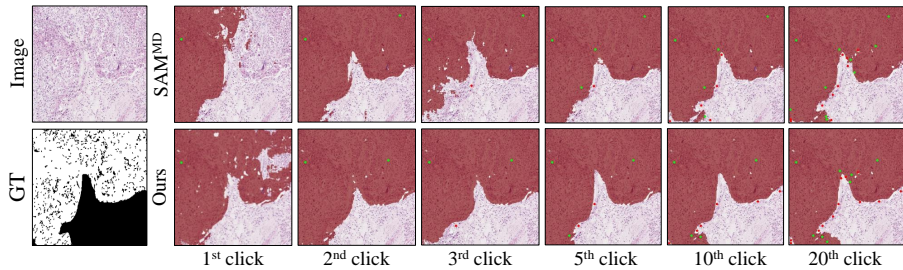⋆ Corresponding author: wkjeong@korea.ac.kr

Fig. 1: Comparison between SAM^{MD} and ours. The green and red points represent positive and negative clicks, respectively. Each click is automatically generated on the error region in the previous prediction.

potential disparities between training and inference imaging conditions, making generalization challenging.

Recently, the Segment Anything Model (SAM) [13], a large promptable foundation model that allows user interaction, has gained considerable attention as a general image segmentation model, eliminating the need for task-specific annotation, training, and modeling. Inspired by this, the primary motivation behind our work stems from the notion that a foundation model like SAM holds promise as a general-purpose segmentation model, which can also be applied to histopathology image segmentation without domain-specific training. Several early attempts have been made to employ SAM in medical image segmentation [8, 18, 22]. However, the exploration of SAM's potential for interactive segmentation has been limited, with only subjective criteria-based prompts being utilized. Another motivation of our work is that supervised training may not generalize well to various target images during inference. Hence, utilizing interactive segmentation to modify the segmentation model's results could prove to be a practical and effective strategy for improving segmentation performance. While numerous interactive segmentation methods exist in computer vision [4, 16, 24], only a few works have been proposed specifically for interactive histopathology image segmentation [5, 20] so far.

In this study, we investigate the potential of SAM for tumor region segmentation in histopathology images using a click-based interactive approach. Our primary objectives are to answer the following questions: 1) Can SAM be directly (zero-shot) applied to interactive histopathology image segmentation tasks? and 2) If not, what is the optimal approach for modifying (or fine-tuning) SAM for interactive histopathology image segmentation? To address these questions, we conducted extensive experiments on two publicly available datasets: PAIP2019 [12] and CAMELYON16 [7]. We investigated SAM mainly with point prompts, using a well-established evaluation protocol by Xu et al. [25] which mimics human click interaction. We conducted a comparison between SAM and other state-of-the-art (SOTA) interactive segmentation methods [4, 16, 24], both

with and without dataset-specific fine-tuning. We also investigated an efficient strategy to harness the potential of SAM through various fine-tuning scenarios. The main contributions of our work are several-fold as follows:

– We assessed SAM's current capability for zero-shot histopathology image segmentation in the context of interactive segmentation by comparing it against SOTA interactive segmentation algorithms.
– We provide insights into the utilization of pretrained weights of SAM by exploring various fine-tuning scenarios. We discovered that SAM requires a prediction refinement strategy for interactive histopathology image segmentation.
– We introduce a modified mask decoder for SAM which enhances performance and reduces the fine-tuning cost while retaining the original SAM's high generalization capability and inference speed. As a result, we achieved an average reduction of 5.19% in the number of clicks required to reach the target IoU.

2   Method

2.1   Overview of Segment Anything Model (SAM)

Introduced by Meta AI, SAM [13] is a promptable foundation model for image segmentation trained with the largest segmentation dataset (SA-1B) over one billion masks and 11 million images. This model demonstrates significant zero-shot performance in natural image domains. SAM consists of three main components: An Image Encoder (IE) is a Vision Transformer (ViT) [6,10]-based encoder to extract image features, a Prompt Encoder (PE) encodes various types of prompts such as points, bounding boxes, masks, and texts, and a lightweight Mask Decoder (MD) maps image embedding and prompt embeddings to segmentation results.

2.2   SAM Fine-tuning Scenarios

We set up three scenarios for fine-tuning SAM to understand the impact of each component in SAM on the performance and explore an efficient method for utilizing SAM in the interactive segmentation(see Fig. 2 (a) and (b)). First, we train only the lightweight mask decoder (SAM^{MD}) by freezing the pretrained image encoder and prompt encoder. Second, we train IE and MD (SAM^{IE_MD}) to investigate the influence of PE on the model's generalization ability. Third, we train the whole model (SAM^{Whole}) for comparison to other scenarios.

To fully utilize the pretrained weights of SAM's ViT-based image encoder, we resized input patches to a size of 1024×1024 and restored the output predictions to match the original input patch size. In the training process, we employed the click guidance scheme by Sofiiuk et al. [24] for automatic point prompts generation. The click guidance scheme uses random sampling for the first iteration and samples subsequent clicks from the error regions of the previous predictions, thereby better resembling real-world user interaction. Moreover, we use the previous prediction as a mask prompt to improve the model performance as
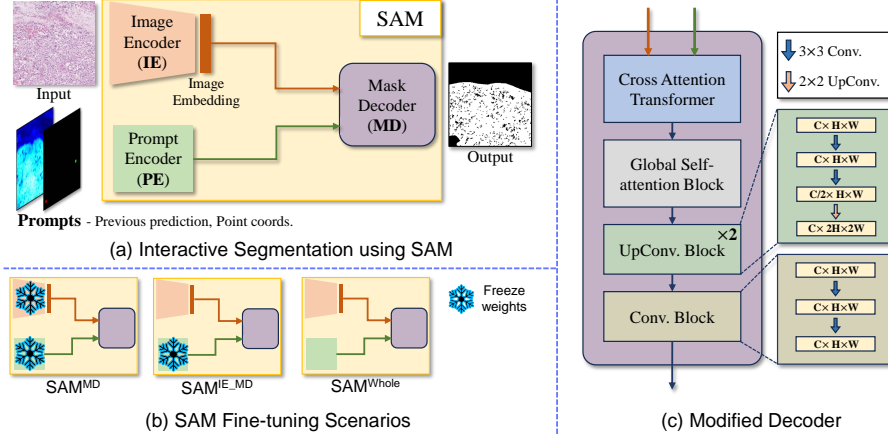
Fig. 2: (a) An illustration of interactive segmentation process using SAM. Image Encoder (IE) extracts features from input patch and Prompt Encoder (PE) encodes prompts which are previous prediction as mask prompt and point coordinates as point prompts. (b) We set up three SAM fine-tuning scenarios. SAM^{MD} freezes IE and PE, training only MD. SAM^{IE-MD} freezes PE and trains IE and MD. SAM^{Whole} trains the entire SAM. (c) Modified mask decoder architecture to improve local refinement capability. Compared to original MD, we exclude dot-product between the final output token and image embedding. Also, we use global self-attention block, and deeper decoder layers.

shown in [19]. We employed the Normalized Focal Loss [24], known for faster convergence and better accuracy compared to Binary Cross-Entropy (BCE) as explained in [4, 16, 17].

2.3 Decoder Architecture Modification

The graph in Fig. 3 (a), and (b) show the zero-shot and fine-tuned performance with mean Intersection over Union (mIoU) per interaction of SAM and SOTA interactive models on PAIP2019 [12] dataset. SAM demonstrates comparable performance to other models in early iterations, but it struggles in later iterations. As shown in Fig. 4, SAM without fine-tuning shows weakness in refining predictions locally (i.e., a local modification affects a large area). Considering the architecture of the ViT-based SAM image encoder, training the entire SAM requires a longer time and a higher computational cost. Also, as shown in Table 1 and described in Sec. 4.2, we empirically found that including IE in fine-tuning is not always beneficial.

To address this issue, we modified the lightweight decoder in the original SAM as depicted in Fig. 2 (c) to improve local refinement capability and assign prompt stability. We add a global self-attention layer to the image embedding after the cross-attention transformer block to enhance the ability to capture the global context across the entire patch. Moreover, we deepen the decoder layers
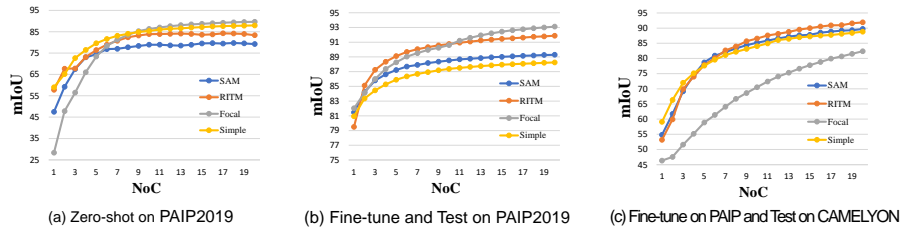
Fig. 3: (a) Zero-shot mIoU scores per click of each model on the PAIP2019 dataset. (b) mIoUs per click after fine-tuning on the PAIP dataset. (c) mIoUs per click on the CAMELYON16 ×10 dataset for models trained on the PAIP dataset.

during the upsampling process to increase the representational capacity of the decoder.

The upsampling process is described as follows: We constructed a *UpConvBlock* and *ConvBlock* module. *UpConvBlock* consists of two 3×3 convolution layers and a 2×2 up-convolution layer and *ConvBlock* consists of two 3×3 convolution layers. In the second 3×3 convolution layer of both *UpConvBlock* and *ConvBlock*, a channel reduction is performed, reducing the input channel size by half. After global self-attention, the image embedding proceeds through two *UpConvBlock*s and *ConvBlock* sequentially. We integrated instance normalization layers and *GELU* activation functions between each layer. Furthermore, to mitigate information loss, we employed shortcut connections within each Block. After passing through 3 Blocks, features are transformed into a segmentation map through the 1×1 convolution layer.

3 Experiments

3.1 Data Description

We utilized two publicly available WSI datasets, namely PAIP2019 [12] and CAMELYON16 [7]. The PAIP2019 WSIs were scaled at a 5× magnification, while for CAMELYON16, we used WSIs scaled at both 10× and 5× magnification. Subsequently, the WSIs were cropped into patches of size 400×400 pixels. These patches were then filtered based on the proportion of tumor area, ranging from 20% to 80%, as outlined in [20]. Consequently, for the PAIP2019 dataset, we employed 5190, 576, and 654 patches for training, validation, and testing, respectively. As for CAMELYON16, we utilized 397 patches at 10× magnification and 318 patches at 5× magnification scales for testing.

3.2 Implementation Details

We compared SAM to three SOTA interactive segmentation methods: RITM [24], FocalClick [4], and SimpleClick [16]. We fine-tuned the pretrained models in PAIP2019 and validated their generalization capabilities in CAMELYON16. For a fair comparison, we used the largest pretrained model and the best-performing pretrained weights for each model. For fine-tuning the parameters, we adhered
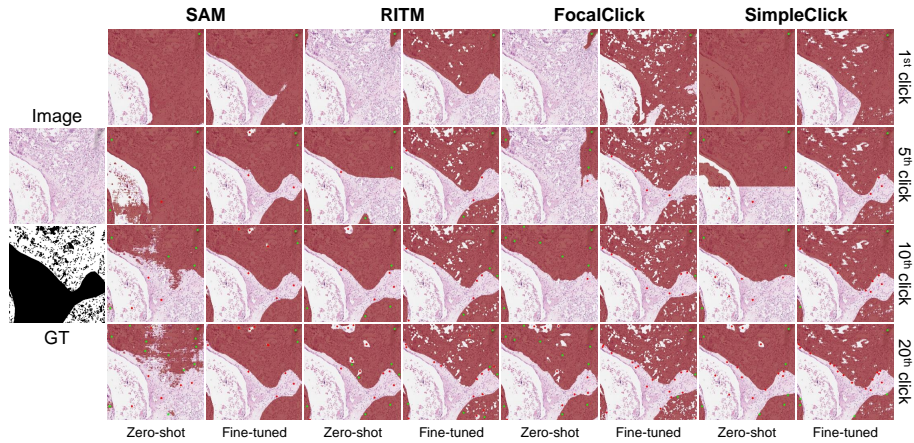
Fig. 4: Qualitative comparison between SAM$^{\text{Whole}}$ and SOTA methods. The results show the performance of each method on the PAIP2019 dataset, both in terms of zero-shot and after fine-tuning.

to the default setting of [24], wherein we set the initial learning rate to 5e-4. Additionally, we decreased the learning rate by a factor of 10 at the 20th and 25th epochs out of the total 30 epochs.

To assess the zero-shot transfer ability of each model, we conducted inference without training using WSIs. Additionally, to evaluate the generalization ability and domain adaptability, we performed inference after fine-tuning using WSIs. For a fair comparison, we employed an automatic evaluation method used in the previous work [3, 11, 14, 15, 23, 25]. Inference per patch continues until it reaches the target IoU (Intersection over Union). We set the maximum number of clicks per patch to 20. As for the evaluation metrics, we used Number of Clicks (NoC), Seconds per Click (SPC), and Number of Fails (NoF). NoC represents the average number of clicks required to reach the target IoU, while SPC measures the average time it takes for the model to perform an inference per click. NoF represents the number of images that failed to reach the target IoU despite the maximum number of clicks. For a more intuitive representation, we divide NoF by the number of testset n. All of evaluations were performed on a single NVDIA RTX A6000 GPU.

4   Results

4.1   Zero-shot Performance

Table 1 presents the zero-shot performance of SAM and state-of-the-art (SOTA) interactive segmentation models. SAM demonstrates lower performance compared to SOTA models across all three different domains. The low NoC and high NoF in the zero-shot setting indicate that SAM lacks the ability to effectively modify the mask to accurately reflect user intent, particularly when faced

Table 1: Quantitative Result of Zero-Shot Performance

Dataset	Method	NoC@			SPC(s)	NoF/n @	
		80	85	90		85	90
PAIP2019 (×5)	RITM	7.43	10.24	13.00	0.075	0.30	0.47
	FocalClick	7.37	**9.89**	**12.42**	0.073	**0.28**	**0.43**
	SimpleClick	**7.21**	10.16	13.44	0.189	0.33	0.53
	SAM	9.13	12.10	14.73	**0.052**	0.50	0.66
CAMELYON16 (×5)	RITM	6.25	7.88	9.98	0.078	0.14	0.24
	FocalClick	**3.64**	**4.83**	**7.20**	0.085	**0.04**	**0.14**
	SimpleClick	5.26	6.94	9.23	0.187	0.15	0.25
	SAM	5.31	7.45	10.38	**0.053**	0.19	0.36
CAMELYON16 (×10)	RITM	6.91	8.19	10.07	0.077	0.15	0.22
	FocalClick	**4.73**	**5.89**	**7.87**	0.076	**0.06**	**0.14**
	SimpleClick	5.20	6.42	8.55	0.187	0.11	0.23
	SAM	6.64	8.49	11.03	**0.053**	0.24	0.37

Table 2: Quantitative Result of After Fine-tuning on PAIP2019

Dataset	Method	NoC@			SPC(s)	NoF/n @	
		80	85	90		85	90
PAIP2019 (×5)	RITM	<u>2.78</u>	4.93	9.40	0.075	0.12	0.33
	FocalClick	**2.58**	**4.54**	**8.98**	0.071	**0.06**	**0.25**
	SimpleClick	5.11	8.33	12.20	0.189	0.32	0.52
	SAM^{MD}	5.80	8.93	12.09	**0.050**	0.31	0.48
	SAM$^{IE\_MD}$	4.53	7.58	10.86	**0.050**	0.29	0.45
	SAM^{Whole}	4.53	7.50	10.95	**0.052**	0.28	0.46
	Ours	4.75	7.78	10.85	0.067	0.26	0.42
CAMELYON16 (×5)	RITM	3.74$_{(-2.51)}$	5.04$_{(-2.84)}$	7.50$_{(-2.48)}$	0.081	**0.08**	**0.17**
	FocalClick	4.15$_{(+0.51)}$	5.57$_{(+0.74)}$	8.19$_{(+0.99)}$	0.071	<u>0.10</u>	0.22
	SimpleClick	3.82$_{(-1.44)}$	5.41$_{(-1.53)}$	7.94$_{(-1.29)}$	0.187	0.15	0.25
	SAM^{MD}	4.09$_{(-1.22)}$	5.84$_{(-1.61)}$	8.31$_{(-2.07)}$	**0.050**	0.13	0.24
	SAM$^{IE\_MD}$	4.30$_{(-1.01)}$	5.64$_{(-1.81)}$	7.85$_{(-2.53)}$	**0.052**	0.15	0.24
	SAM^{Whole}	**3.28**$_{(-2.03)}$	**4.80**$_{(-2.65)}$	**7.06**$_{(-3.32)}$	**0.053**	<u>0.10</u>	<u>0.20</u>
	Ours	4.27$_{(-1.04)}$	5.59$_{(-1.86)}$	8.04$_{(-2.34)}$	<u>0.066</u>	<u>0.11</u>	<u>0.20</u>
CAMELYON16 (×10)	RITM	6.28$_{(-0.63)}$	7.65$_{(-0.54)}$	9.67$_{(-0.31)}$	0.076	<u>0.13</u>	<u>0.23</u>
	FocalClick	11.82$_{(+7.09)}$	13.01$_{(+7.12)}$	14.53$_{(+7.33)}$	0.076	0.43	0.54
	SimpleClick	6.07$_{(+0.87)}$	7.44$_{(+1.02)}$	9.94$_{(+1.39)}$	0.187	0.18	0.28
	SAM^{MD}	<u>4.88</u>$_{(-1.76)}$	6.68$_{(-1.81)}$	9.07$_{(-1.96)}$	**0.053**	<u>0.13</u>	0.25
	SAM$^{IE\_MD}$	7.63$_{(+0.99)}$	9.19$_{(+0.7)}$	11.81$_{(+0.78)}$	**0.049**	0.23	0.38
	SAM^{Whole}	6.60$_{(-0.04)}$	8.10$_{(-0.39)}$	10.66$_{(-0.37)}$	**0.053**	0.20	0.31
	Ours	**4.59**$_{(-2.05)}$	**5.92**$_{(-2.57)}$	**8.40**$_{(-2.63)}$	<u>0.064</u>	**0.11**	**0.20**

with challenging images. As depicted in Fig. 4, SAM in the zero-shot setting fails to refine local information. However, SAM exhibits minimal SPC compared to other models due to its feature extraction being performed only once on the input image, resulting in a significantly faster processing time, over three times faster than SimpleClick.

4.2   Fine-tuned SAM Performance

As introduced in Sec 2.2, we compare the three scenarios to verify the impact of each SAM component on fine-tuned performance. We fine-tuned the entire SAM (SAM^{Whole}), the IE and MD (SAM^{IE-MD}), and only the MD (SAM^{MD}) using the PAIP2019 dataset, respectively. Then, we verify the generalization ability of models at two scales of the CAMELYON16 dataset. As shown in Table 2, SAM^{MD} yield lower performance on NoC and NoF compared to IE-trained scenarios on the PAIP2019 dataset. On CAMELYON16, SAM^{MD} exhibited comparable results as IE-trained cases at scale $\times 5$ and better performance at scale $\times 10$. This result demonstrates that the pretrained IE of SAM shows the robustness of feature extraction ability on different data distributions.

4.3   Comparison between SAM and SOTA Interactive Methods

As shown in Table 2, SAM still exhibits a notable performance gap compared to RITM and FocalClick on the PAIP2019 dataset. SAM requires at least two more NoC compared to RITM and FocalClick. SimpleClick exhibited poorer performance compared to SAM^{Whole} and SAM^{IE-MD}, but similar or slightly better performance compared to SAM^{MD}. On the CAMELYON16 $\times 5$ dataset, SAM^{Whole} exhibited the best performance. The numbers in parentheses in Table 2 represent the changes in the NoC metric compared to the zero-shot performance in Table 1. All models showed a decrease in NoC but an increase in FocalClick, indicating that CAMELYON16 $\times 5$ has similar data distribution with the PAIP2019 dataset. As for the CAMELYON16 $\times 10$ dataset, SAM^{MD} performed second best after SAM with a modified decoder (ours). Furthermore, despite significant differences in data distributions between source and target domain, SAM^{MD} and SAM^{Whole} demonstrated improved performance along with RITM, indicating a higher generalized capability compared to FocalClick and SimpleClick. Note that FocalClick and SimpleClick performed poorly compared to the zero-shot approach with FocalClick's significant performance drop as shown in Fig. 3 (c). These methods utilize local segmentation which refines the prediction locally. As the data distribution changes, more iterations of local segmentation are required, resulting in performance degradation in both NoC and NoF metrics.

4.4   Modified SAM Decoder Performance

As shown in Table 2, our approach exhibits improved performance compared to SAM^{MD} in every dataset. Moreover, it shows comparable performance in PAIP2019 compared to SAM^{Whole}. Especially on the CAMELYON16 x10 dataset, our approach demonstrates the best performance among all interactive SOTA

models. It shows the highest performance in all of NoC and NoF, highlighting its remarkable generalization capability. However, in PAIP2019, it still shows lower performance compared to SOTA interactive models. Intuitively, Table 2 may not be enough to say that our approach has a clear advantage over other methods. However, when taking into account a comprehensive range of factors including performance, inferencing speed, training cost, and generalization ability, we argue that our approach is sufficiently compelling.

5 Conclusion

In this study, we demonstrate the potential of SAM for interactive pathology segmentation. SAM showed higher generalization capability and notably excelled in terms of inference speed per interaction compared to SOTA interactive models. Additionally, a modified decoder with pretrained encoders of SAM can achieve performance comparable to that of the entire SAM fine-tuning. In this approach, SAM could efficiently capture user intent and precisely refine predictions, which is crucial in interactive segmentation. However, SAM exhibits lower performance compared to state-of-the-art interactive models, especially in achieving high IoU scores. We plan to develope SAM as a foundational interactive model that works well for all histopathology data (different organs, tissues, etc.).

References

1. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE transactions on pattern analysis and machine intelligence **26**(9), 1124–1137 (2004)
2. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Transactions on image processing **10**(2), 266–277 (2001)
3. Chen, X., Zhao, Z., Yu, F., Zhang, Y., Duan, M.: Conditional diffusion for interactive segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7345–7354 (2021)
4. Chen, X., Zhao, Z., Zhang, Y., Duan, M., Qi, D., Zhao, H.: Focalclick: towards practical interactive image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1300–1309 (2022)
5. Cho, S., Jang, H., Tan, J.W., Jeong, W.K.: DeepScribble: interactive pathology image segmentation using deep neural networks with scribbles. In: 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI). pp. 761–765. IEEE (2021)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth

16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)

7. Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J.A.W.M., , the CAMELYON16 Consortium: Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. JAMA **318**(22), 2199–2210 (12 2017)

8. Gao, Y., Xia, W., Hu, D., Gao, X.: Desam: Decoupling segment anything model for generalizable medical image segmentation. arXiv preprint arXiv:2306.00499 (2023)

9. Grady, L.: Random walks for image segmentation. IEEE transactions on pattern analysis and machine intelligence **28**(11), 1768–1783 (2006)

10. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)

11. Jang, W.D., Kim, C.S.: Interactive image segmentation via backpropagating refinement scheme. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5297–5306 (2019)

12. Kim, Y.J., Jang, H., Lee, K., Park, S., Min, S.G., Hong, C., Park, J.H., Lee, K., Kim, J., Hong, W., et al.: Paip 2019: Liver cancer segmentation challenge. Medical image analysis **67**, 101854 (2021)

13. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)

14. Li, Z., Chen, Q., Koltun, V.: Interactive image segmentation with latent diversity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 577–585 (2018)

15. Lin, Z., Zhang, Z., Chen, L.Z., Cheng, M.M., Lu, S.P.: Interactive image segmentation with first click attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13339–13348 (2020)

16. Liu, Q., Xu, Z., Bertasius, G., Niethammer, M.: Simpleclick: Interactive image segmentation with simple vision transformers (2023)

17. Liu, Q., Zheng, M., Planche, B., Karanam, S., Chen, T., Niethammer, M., Wu, Z.: Pseudoclick: Interactive image segmentation with click imitation. In: European Conference on Computer Vision. pp. 728–745. Springer (2022)

18. Ma, J., Wang, B.: Segment anything in medical images. arXiv preprint arXiv:2304.12306 (2023)

19. Mahadevan, S., Voigtlaender, P., Leibe, B.: Iteratively trained interactive segmentation. In: Proceedings of the British Machine Vision Conference (BMVC) (2018)

20. Min, S., Jeong, W.K.: CGAM: Click-guided attention module for interactive pathology image segmentation via backpropagating refinement (2023)

21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)

22. 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)

23. Sofiiuk, K., Petrov, I., Barinova, O., Konushin, A.: f-brs: Rethinking backpropagating refinement for interactive segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8623–8632 (2020)

24. Sofiiuk, K., Petrov, I.A., Konushin, A.: Reviving iterative training with mask guidance for interactive segmentation. In: 2022 IEEE International Conference on Image Processing (ICIP). pp. 3141–3145. IEEE (2022)
25. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.S.: Deep interactive object selection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 373–381 (2016)