FocSAM: Delving Deeply into Focused Objects in Segmenting Anything

Supplementary Material

The "SAM vs FocSAM.mp4" comparison video is available in the supplemental materials.

A. Implementation Details

A.1. Datasets

In this study, we conduct experiments on six datasets to assess our methods comprehensively:

- **GrabCut** [14]: Features 50 images, each with distinct foreground and background, totaling 50 instances.
- **Berkeley** [7]: Comprises 96 images (100 instances), with some overlap with GrabCut.
- **DAVIS** [13]: Focuses on 345 specific frames from 50 videos, aligning with previous studies [2, 9, 11].
- **SBD** [5]: Includes 2857 validation images with 6671 instances for evaluation purposes.
- **MVTec** [1]: Selected for its high-quality pixel-wise annotations of industrial defects, ideal for interactive segmentation's practical applications. Specific defects like cut-lead and misplaced elements in transistors are excluded due to their misalignment with image segmentation, refining the dataset to 1238 instances.
- **COD10K** [3]: Contains 2026 instances of camouflaged objects that blend into their backgrounds, providing a distinct challenge for interactive segmentation.

A.2. Training Details

In training our proposed FocSAM on COCO [10] and LVIS [4], we adopt the AdamW optimizer [12]. The initial learning rate is set to $1e-6$ for the first $1,500$ iterations, which is then raised to $1e-4$. We then apply a polynomial decay to the learning rate, setting AdamW's β_1 to 0.9 and β_2 to 0.999. Our batch size is 4 per GPU, totaling 16 samples across 4 GPUs, and images are resized and padded to 1024×1024. We attempt to jointly train FocSAM with the SAM decoder, but such a strategy results in unstable training. Therefore, we fine-tune the SAM decoder [8] alone over $320,000$ iterations at the first stage. Then, at the second stage we freeze the trained decoder and train the FocSAM's focus refiner for additional $160,000$ iterations.

A.3. Click Simulation

During training, we adopt the click simulation strategy from InterFormer [6] due to its simplicity. We set the upper limit for simulated clicks at 20. To determine the distribution of click counts, we employ a decay coefficient γ, where the probability for a given number of clicks decreases progressively. Specifically, the probability of having i clicks is γ times the probability of having $i-1$ clicks, with the constraint that $\gamma < 1$. This method ensures a higher likelihood of selecting fewer clicks, reducing computational costs. For joint training on COCO [10] and LVIS [4] datasets, InterFormer [6] sets γ at 0.6 for both. Instead, to avoid bias towards small objects in LVIS, we use different γ values for COCO ($\gamma = 0.6$) and LVIS ($\gamma = 0.9$). This adjustment allows for more effective use of LVIS's detailed annotations in the later refinement stages. In FocSAM, we first decide the number of clicks, N, and then determine the refinement step, K, using a similar sampling strategy, where we set distinct γ_r values for COCO ($\gamma_r = 0.6$) and LVIS ($\gamma_r = 0.35$) with N as the upper limit to ensure a similar refinement process. After determining the N and K (only for FocSAM), SAM and FocSAM perform click simulations on training images using GT as an oracle to specify clicks randomly within incorrectly predicted regions.

B. Ablation Study

B.1. Convergence Analysis

We perform convergence analysis experiments on the SBD [5], DAVIS [13], MVTec [1], and COD10K [3] datasets with sufficient samples. In these experiments, we compute the average IoU for all samples at each click, comparing our FocSAM with previous methods [2, 6, 11]. As depicted in Figure 1, the results showcase FocSAM's fast convergence across these datasets. FocSAM notably achieves high IoU values with only a few clicks. Such rapid convergence is particularly pronounced in the challenging MVTec [1] and COD10K [3] datasets, where FocSAM outperforms other methods, including the previous state-of-the-art SimpleClick-ViT-H [11]. In SBD [5] and DAVIS [13] datasets, FocSAM demonstrates a convergence rate on par with SimpleClick-ViT-H [11], underscoring its efficiency in various interactive segmentation scenarios.

B.2. SAM's Bounding Box Prompt

Experimental settings. SAM [8] can simultaneously process click and bounding box prompts. Notably, in our proposed FocSAM, the Dwin-MSA module conceptually shares similarities with the processing of bounding box prompts. Therefore, we evaluate SAM with additional bounding boxes around target objects for ablation studies. Specifically, we utilize the GT to find the bounding box encompassing the target object and expand it by $1.4\times$ to include the context of the surrounding area. During the interactive segmentation of SAM, these boxes are supplied as an additional prompt. Likewise, we report the results on

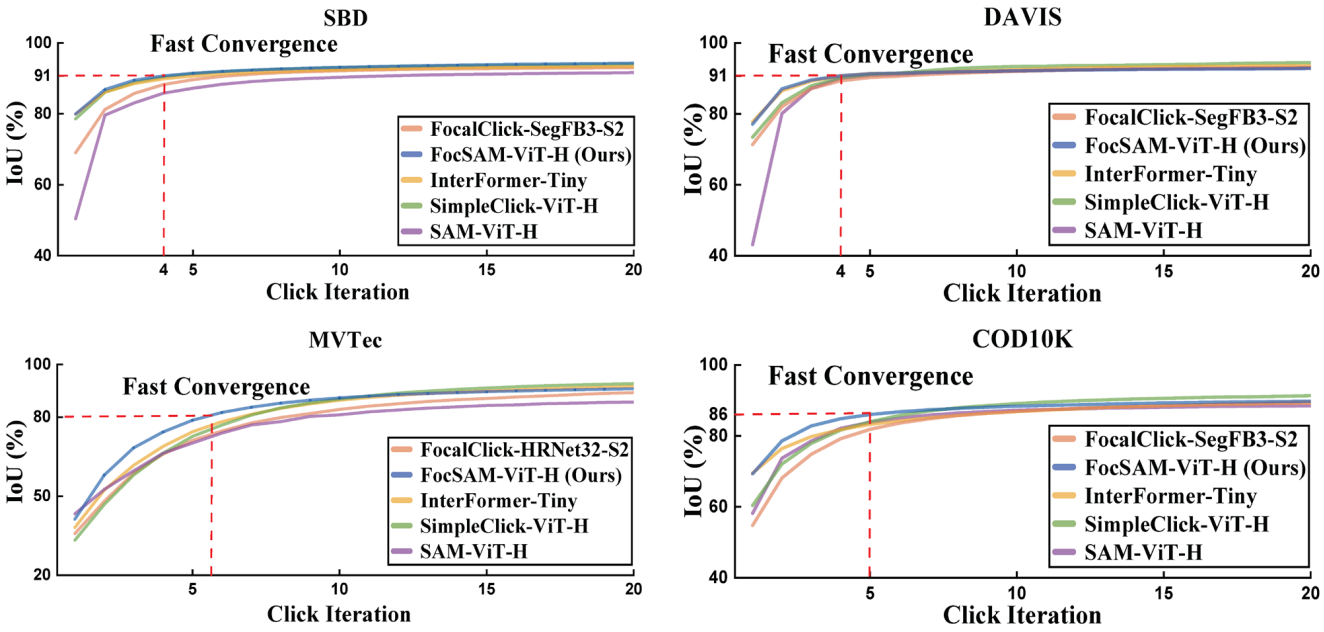

Figure 1. Convergence Analysis. Each subfigure displays the average IoU for all samples at successive clicks. These plots illustrate the rapid convergence of FocSAM, which achieves high IoU values with only a few clicks.

Method	SBD		MVTec		COD10K	
	20NoC@90	100NoC@95	20NoC@90	100NoC@95	20NoC@90	100NoC@95
SAM (w/o BBox)	7.62	63.40	13.97	81.90	10.36	76.73
SAM (w/ BBox)	7.27	63.28	13.71	82.08	10.66	76.65
FocSAM	4.69	32.96	11.14	62.82	8.91	62.61

Table 1. Ablation study on SAM with bounding boxes.

Dwin-MSA	SBD		MVTec		COD10K	
	20NoC@90	100NoC@95	20NoC@90	100NoC@95	20NoC@90	100NoC@95
Window-16	4.69	32.96	11.14	62.82	8.91	62.61
Window-8	4.75	34.08	11.31	64.69	9.21	64.17
Window-32	4.85	33.95	11.27	63.54	9.25	64.18

Table 2. Ablation study on Dwin-MSA's window sizes.

SBD [5], MVTec [1], and COD10K [3] datasets, including the metrics 20NoC@90 and 100NoC@95.

Results. Table 1 reveals that integrating interactive information from bounding boxes offers marginal improvement to SAM's performance. This demonstrates that SAM cannot fully exploit the potential of such interactive information from the additional boxes. In contrast, FocSAM effectively utilizes similar information through its Dwin-MSA module. Specifically, FocSAM enhances the performance by overlaying bounding boxes on previous predictions and feeding these into the Dwin-MSA module to select windows relevant to the object. This approach underscores FocSAM's efficiency in leveraging available information for enhanced performance.

B.3. Impact of Dwin-MSA's Window Size

In Table 2, our ablation study on Dwin-MSA's window size indicates window-16 outperforms both window-8 and window-32. The limited attention scope of window-8 constrains its performance. In contrast, while window-32 has a broader attention span, it incorporates excessive object-unrelated areas, which undermines its effectiveness.

C. Qualitative Results

In Figure 2 3, we present the interactive segmentation results of FocSAM and SAM across various scenarios. For a more comprehensive set of results, please refer to the accompanying video titled "SAM vs FocSAM.mp4."

Figure 2. Qualitative results (1). On the left, an example is depicted with an image overlaid with its GT (blue mask). To the right, two rows display interactive segmentation results at the 1st, 5th, 10th, and 20th clicks, where the most recent click is highlighted with a star, green for positive and red for negative feedback. The top row illustrates the results from SAM, and the bottom row shows those from FocSAM. These visual comparisons reveal the segmentation efficiency of FocSAM and SAM at different stages of annotator clicks.

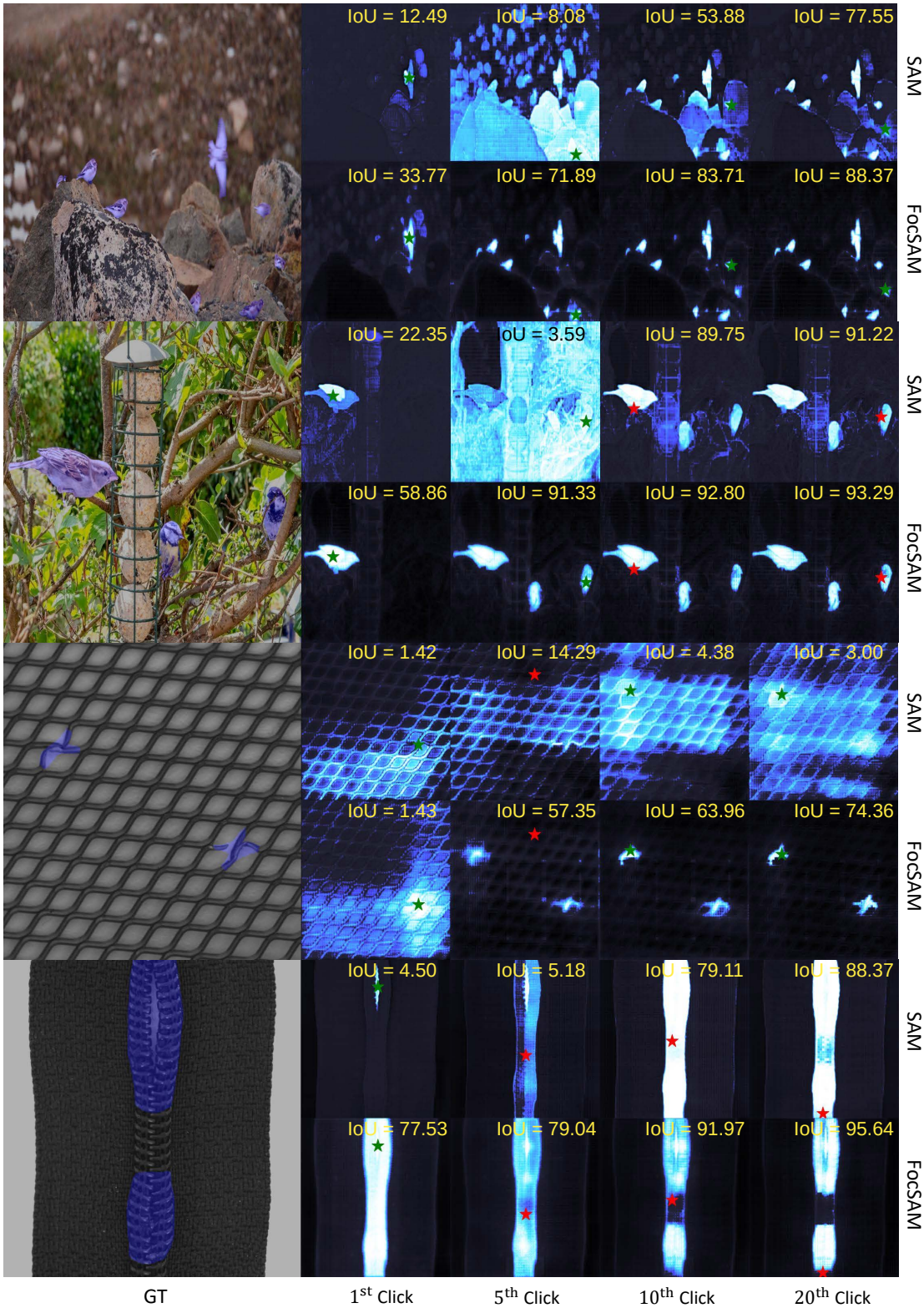

Figure 3. Qualitative results (2). On the left, an example is depicted with an image overlaid with its GT (blue mask). To the right, two rows display interactive segmentation results at the 1st, 5th, 10th, and 20th clicks, where the most recent click is highlighted with a star, green for positive and red for negative feedback. The top row illustrates the results from SAM, and the bottom row shows those from FocSAM. These visual comparisons reveal the segmentation efficiency of FocSAM and SAM at different stages of annotator clicks.

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 1, 2

[2] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: towards practical interactive image segmentation. pages 1300–1309, 2022. 1

[3] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2777–2787, 2020. 1, 2

[4] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5356–5364. Computer Vision Foundation / IEEE, 2019. 1

[5] Bharath Hariharan, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 991–998. IEEE Computer Society, 2011. 1, 2

[6] You Huang, Hao Yang, Ke Sun, Shengchuan Zhang, Liujuan Cao, Guannan Jiang, and Rongrong Ji. Interformer: Real-time interactive image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22301–22311, 2023. 1

[7] Noel E. O'Connor Kevin McGuinness. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 2010. 1

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 1

[9] Anton Konushin Konstantin Sofiiuk, Ilia A. Petrov. Reviving iterative training with mask guidance for interactive segmentation. *arXiv: Computer Vision and Pattern Recognition*, 2021. 1

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Lecture Notes in Computer Science*, 2014. 1

[11] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. *arXiv preprint arXiv:2210.11006*, 2022. 1

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[13] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 724–732. IEEE Computer Society, 2016. 1

[14] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3): 309–314, 2004. 1