Supplementary Material for ReMamber: Referring Image Segmentation with Mamba Twister

Yuhuan Yang*[1], Chaofan Ma*[1], Jiangchao Yao[1], Zhun Zhong[2], Ya Zhang[1], and Yanfeng Wang[1]

[1] Shanghai Jiao Tong University
[2] University of Nottingham
{yangyuhuan,chaofanma,sunarker,ya_zhang,wangyanfeng622}@sjtu.edu.cn
zhunzhong007@gmail.com

1 Speed Analysis

Here, we supplement two experiments *w.r.t.* model FPS and memory cost in Fig. 1 under different resolutions. ReMamber is consistently faster and requires fewer memory cost than LAVT, especially with large resolution (*e.g.*, 1,024).

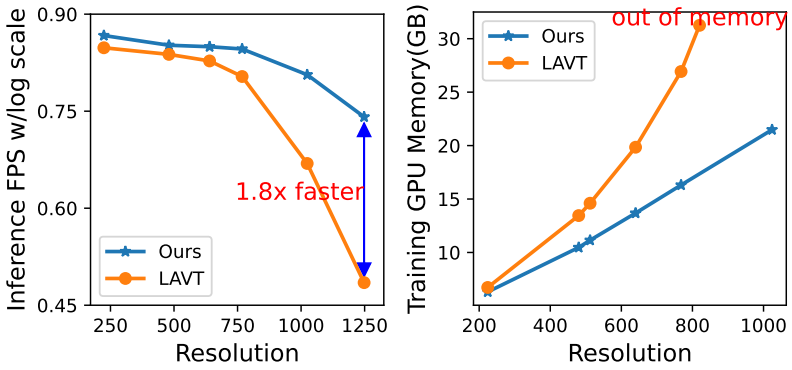

Fig. 1: Comparison of inference FPS (left) and training GPU memory (right) between LAVT and our ReMamber.

2 Implementation Details

Here we provide more details about the implementation of our method, including detailed architectural structure, training settings and other baseline implementation.

2.1 Architecture Details

The whole ReMamber architecture consists of an encoder and a decoder. The encoder is made up by a patch-embedding layer with patch-size 4 and hidden dimension 128, followed by 4 Mamba Twister blocks. Each Twister block consists of several VSS Layers and a Twisting Layer. The VSS Layer number configuration is set as 2-2-15-2, with hidden dimension 128-256-512-1024, respectively.

For the decoder part, we provide two variants in our code implementation: convolution-based decoder (`ReMamber_Conv`) and Mamba-based decoder (`ReMamber_Mamba`). `ReMamber_Conv` uses a progressive upsampling architecture with 4 residual blocks, 2 convolutional layers in each. `ReMamber_Mamba` is similar with `ReMamber_Conv`, but uses VSS layers instead of convolutional layers. This varient is slightly faster.

2.2 Implementation for Other Three Variants

Here we detail describe the implementation of the other three architecture variants in our paper, *i.e.*, In-context Conditioning, Attention-based Conditioning and Norm Adaptation.

In-context Conditioning. To enable the model to distinguish between two modalities, we add learnable position embeddings to the image and text tokens separately at each layer before the Spatial SSM.

Attention-based Conditioning. In this variant, we also incorporate learnable position embeddings. For the cross-attention block, we use image tokens as the query and text tokens as the key and value.

Norm Adaptation. Norm Adaptation learns global scale and bias. Initially, we use an MLP layer to pool a global vector from the text. This vector is then used to scale and bias the image tokens. An additional feed-forward layer is added after adjusting the scale and bias to maintain parameter size comparable to other variants.

3 Visualization Results

Fig. 2 presents the visualization results of our method compared with the baseline method LAVT. Fig. 3 presents the visualization outcomes of our method alongside three other variants. Our **ReMamber** is capable of producing segmentation results with higher accuracy. In contrast, the other three variants occasionally encounter issues with inaccurate segmentation masks or are misled towards incorrect objects.

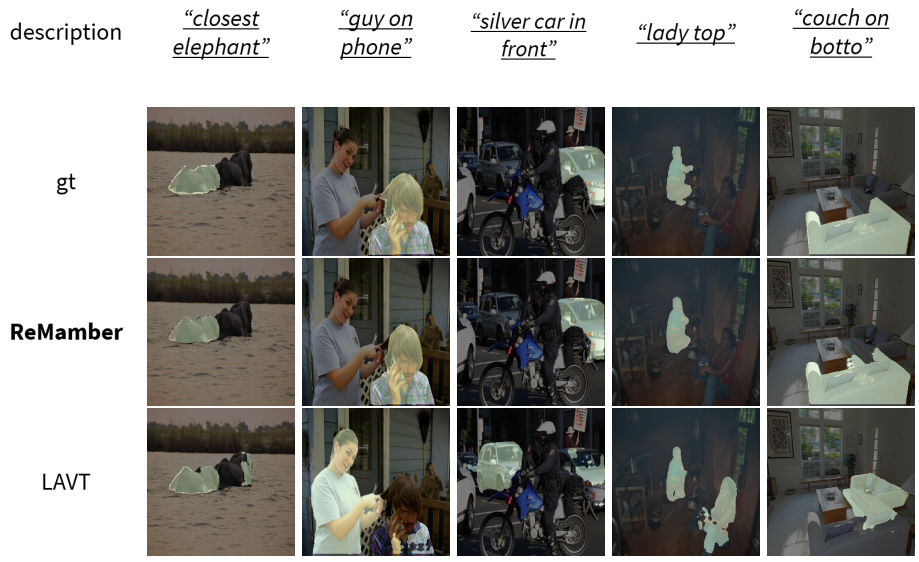

Fig. 2: Visualization results of our **ReMamber** and the baseline model LAVT. Our model is able to predict more accurate masks.

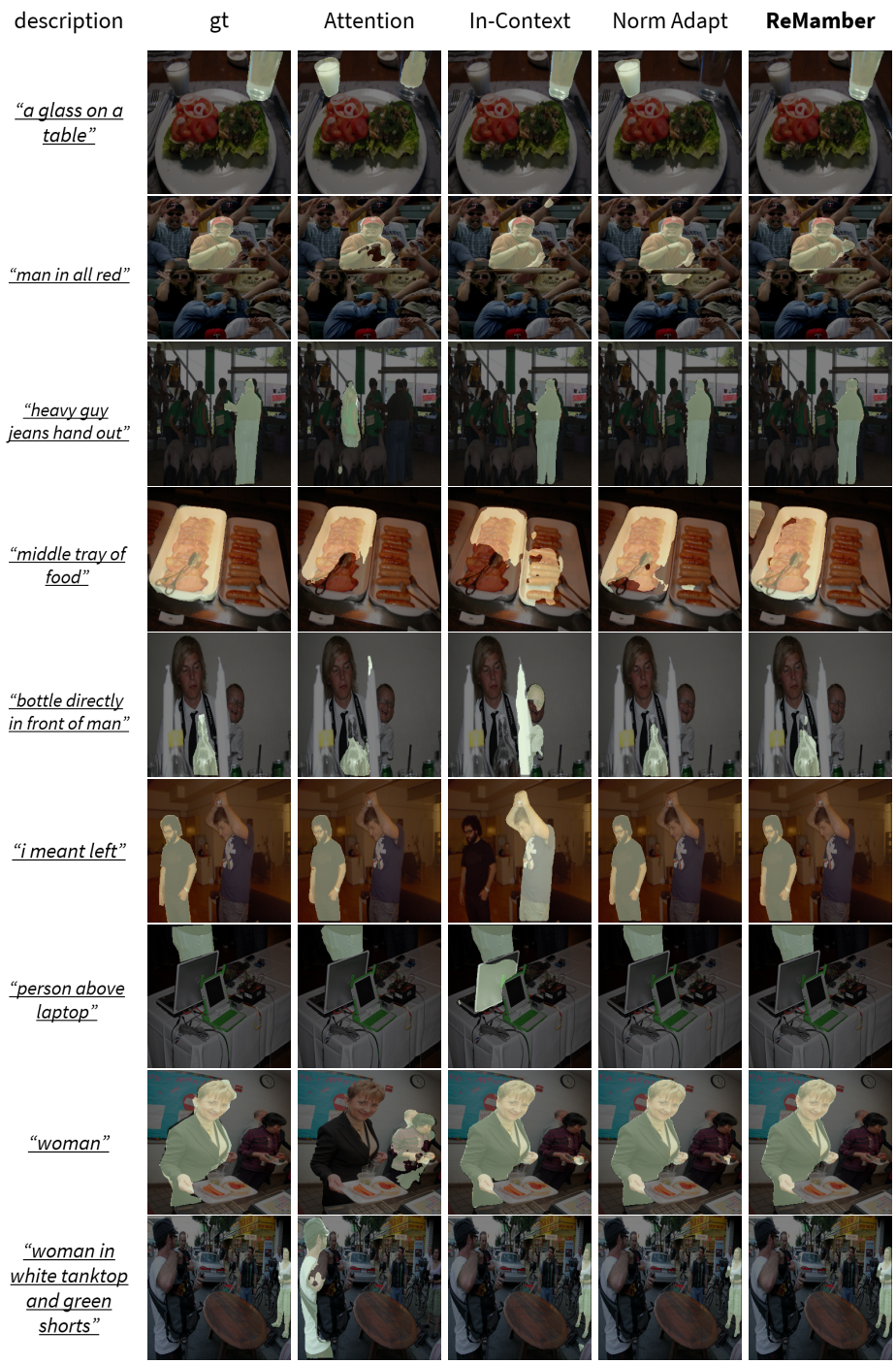

Fig. 3: Visualization results. Our **ReMamber** is capable of producing segmentation results with higher accuracy. While other three variants occasionally encounter issues with inaccurate segmentation masks or are misled towards incorrect objects.