A. More Experimental Details

Software We built our codebase using PyTorch [66] and the CVNets framework [57]. The evaluation code for instance segmentation relies on the publicly released codebases from Kirillov et al. [38] and Li et al. [45].

Hardware We conducted all experiments on servers equipped with 8×A100 GPUs. For training our models, we most employed multi-node training across four 8×A100 servers. The local batch size per server is one-fourth of the global batch size.

CLIP Head Structure We initialized each transformer layer of the CLIP head using parameters from the last transformer layer of SAM ViT-B, as we found this approach to expedite training compared to random initialization. Following the implementation of CLIP-ConvNeXt in Ilharco et al. [29] (the only OpenCLIP model that uses a pooling layer instead of a CLS token), we incorporated a LayerNorm layer subsequent to the pooling layer. After applying LayerNorm, we use a shallow MLP with two hidden layers to project the features into the text-embedding space, consistent with the approach in Rosenfeld et al. [75].

Hyperparameters We employ AdamW optimizers [52] with a learning rate of 8×10^{-4} (consistent with SAM training [38]) during the first training stage (head probing) for 20 epochs. This rate is reduced to 4×10^{-5} during the second stage (joint distillation) for 16 epochs. It should be noted that we apply a learning rate multiplier of 0.1 to the backbone and SAM head in the second stage to mitigate forgetting. The learning rate in the resolution adaptation stage (3 epochs) remains the same as in the first stage. The global image batch size for CLIP distillation is 2048, and for SAM distillation, it is 32 (i.e., 32 images from the SA-1B dataset [38]). In the latter case, we randomly sample 32 masks for each image.

Multi-Task Distillation Our training process consists of two stages: 1) Head probing to learn parameters of $\mathrm{Head}_{\texttt{CLIP}}$ that are initialized randomly, and 2) Joint training of the $\mathrm{Head}_{\texttt{SAM}}$, $\mathrm{Head}_{\texttt{CLIP}}$, and the ViT backbone $\mathrm{Enc}_{\texttt{SAM-CLIP}}$ using a multi-task distillation loss.

In the first stage, only the $\mathrm{Head}_{\texttt{CLIP}}$ is trainable, and it is trained using a single CLIP distillation loss (cosine distance between embeddings as in Equation (1)). At this stage, all image batches are sampled only from $\mathcal{D}_{\texttt{CLIP}}$. This stage involves training for a fixed duration of 20 epochs without early stopping. The motivation for this step is to have a warm start for the $\mathrm{Head}_{\texttt{CLIP}}$ in the next stage where we also allow modifying the backbone, similar to Kumar et al. [40].

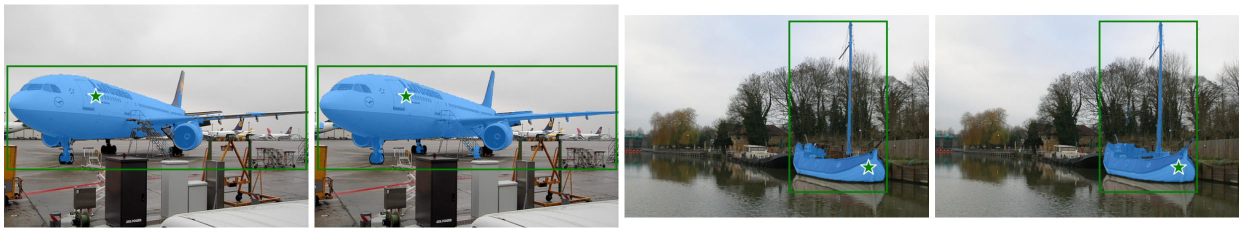
In the second stage, the $\mathrm{Head}_{\texttt{SAM}}$ and the ViT backbone $\mathrm{Enc}_{\texttt{SAM-CLIP}}$ become also trainable, and we have a multi-task objective: CLIP Distillation Equation (1) and SAM self-distillation Equation (2). The balance between the losses is determined by the coefficient λ, which we picked to optimize the trade-off between learning semantic knowledge from CLIP and forgetting SAM's segmentation knowledge. We experimented with $\lambda = 1, 10, 100$, and found that $\lambda = 10$ offers the best trade-off between mitigating the forgetting of SAM's ability and learning CLIP's ability.

Each training step for the second stage is performed as follows:

- Sample a batch of 2048 images from $\mathcal{D}_{\texttt{CLIP}}$. 2048 is determined based on available total GPU memory. Run the forward pass, and compute gradients backward from $\mathcal{L}_{\texttt{CLIP}}$ (note that only parameters of the $\mathrm{Head}_{\texttt{CLIP}}$ and $\mathrm{Enc}_{\texttt{SAM-CLIP}}$ will get gradients after this step).
- Sample a batch of 32 images from $\mathcal{D}_{\texttt{SAM}}$. 32 is determined based on available total GPU memory. Run the forward pass, and compute gradients backward from $\mathcal{L}_{\texttt{SAM}}$ (note that only parameters of the $\mathrm{Head}_{\texttt{SAM}}$ and $\mathrm{Enc}_{\texttt{SAM-CLIP}}$ will get gradients after this step).
- Apply one optimization step (note that at this point, the parameters of the $\mathrm{Enc}_{\texttt{SAM-CLIP}}$ have accumulated gradients from both of the above two steps).

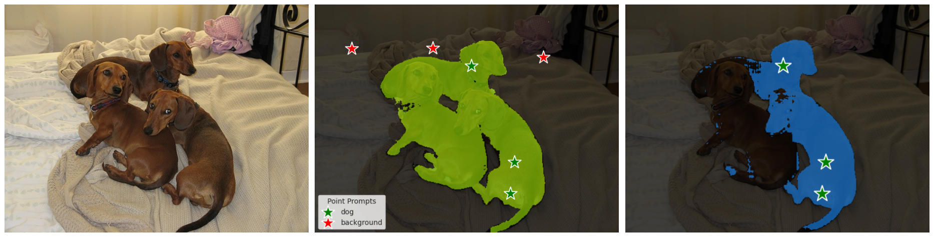
We early-stop after 16 epochs (out of a full training length of 20 epochs) as we observed more forgetting (as measured by instance segmentation performance on the COCO dataset) after the 16th epoch.

Loss Coefficients We empirically determined the loss coefficient ratio of 1:10 for the CLIP and SAM distillation losses from three options: 1:1, 1:10, and 1:100. This ratio provides the best trade-off between mitigating SAM's ability to forget and fostering the learning of CLIP's ability. Specifically, a ratio of 1:1 leads to greater forgetting of SAM's original ability (as measured by the performance drop in instance segmentation on COCO), while ratios of 1:10 and 1:100 maintain it relatively well. However, a ratio of 1:100 impedes the learning of CLIP's ability (as measured by zero-shot accuracy on ImageNet). Therefore, we ultimately selected the ratio of 1:10.

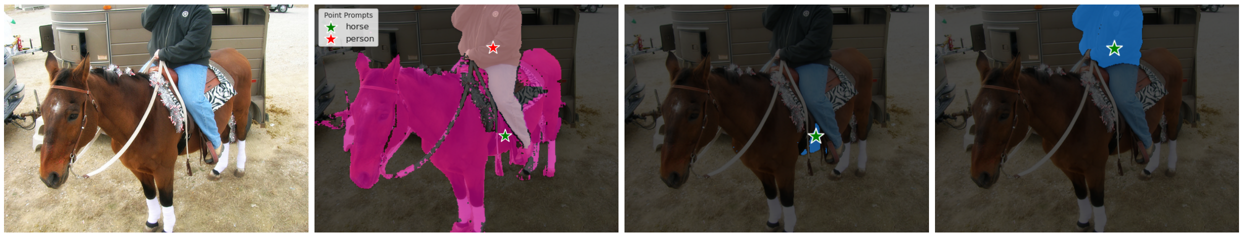


(1a) SAM Output (1b) SAM-CLIP Output (2a) SAM Output (2b) SAM-CLIP Output

Figure 5. Comparison of instance segmentation between SAM and SAM−CLIP . The same images, along with geometric prompts (bounding box and point), are provided to both SAM and SAM−CLIP , and their respective model outputs are displayed above. While the outputs of SAM and SAM−CLIP exhibit slight differences, they are overall quite similar.



(1a) Input image of three dogs (1b) SAM-CLIP Segmentation (1c) SAM Segmentation



(2a) Input image of a horse and a humen (2b) SAM-CLIP Segmentation Mask (2c) SAM Segmentation for the horse (2d) SAM Segmentation for the human

Figure 6. Comparison of SAM vs. SAM−CLIP for semantic segmentation on two images. The segmentation of SAM−CLIP is obtained by: i) using CLIP-head output (i.e., coarse-grained prediction masks) to generate point prompts automatically, and ii) passing the CLIP-head output and point prompts to the SAM-head to generate final fine-grained prediction masks. For SAM, the same point prompts for each class (“dog”, “human”, “human”) are passed to its prompt encoder to generate a segmentation mask.

Image Resolution for Zero-Shot Classification In Table 1, we report the evaluation results for both SAM−CLIP and CLIP models using the 224px image resolution. However, we found that SAM−CLIP benefits from the 336px resolution, whereas the performance of CLIP models deteriorates (they exhibit worse accuracy). The 336px results for SAM−CLIP are incorporated into the diagram in Figure 1. We provide a comparison between the 224px and 336px resolutions for SAM−CLIP in Table 6.

Table 6. Different input resolutions for zero-shot image classification.

Resolution	ImageNet	ImageNet-v2	Places365
224px	71.7	63.2	43.4
336px	72.4	63.2	43.6

B. Visual Comparisons of SAM and SAM-CLIP in Segmentation Tasks

Comparison on Instance Segmentation Table 1 provides a quantitative comparison of SAM and SAM−CLIP on two instance segmentation datasets (COCO and LVIS), showing that SAM−CLIP maintains comparable performance to SAM.

To give readers a more intuitive understanding of the segmentation quality of SAM versus SAM−CLIP , we present two examples in Figure 5. These examples demonstrate that, given the same geometric prompts (bounding box and point prompt), the segmentation masks predicted by SAM and SAM−CLIP are quite similar, with slight differences. This suggests that the segmentation quality of SAM−CLIP is indeed comparable to that of SAM.

Comparison on Semantic Segmentation Figure 3 illustrates the semantic segmentation outputs of SAM−CLIP , featuring both CLIP-head segmentation predictions and SAM-head refined segmentation predictions. Specifically, the SAM-head refinement utilizes the CLIP-head output and some auto-generated point prompts from this output. The same point prompts are fed to SAM ViT-B, with its segmentation prediction shown in Figure 6. It is evident that SAM's prediction typically segments only a sub-part of the object indicated by the point prompts, instead of segmenting the entire semantic object class (e.g., "dog," "horse," "human"). This indicates that the CLIP-head of SAM−CLIP is essential for semantic segmentation, as it provides semantic understanding to the SAM-head of SAM−CLIP . In contrast, the point prompting approach used in SAM [38] is insufficient for semantic segmentation. Furthermore, point prompting requires human-provided points, making it not qualified for *zero-shot* semantic segmentation. In contrast, SAM−CLIP requires only text prompts for each object class (e.g., "dog," "horse," "human") to automatically generate semantic segmentation masks (the point prompts are auto-generated from the CLIP-head output in our pipeline).

C. Inference Experiments

CLIP and SAM Tasks The inference process for zero-shot classification is identical to that of the original CLIP [12, 68]. The evaluation of zero-shot instance segmentation also exactly follows the protocol outlined in Kirillov et al. [38]. The image resolutions for classification and instance segmentation tasks are set at 224px and 1024px, respectively.

Zero-Shot Semantic Segmentation For zero-shot semantic segmentation, we largely adhere to the practices outlined by Ranasinghe et al. [69]. We insert the class names into 80 prompt templates created by Radford et al. [68] and obtain text embeddings using the text encoder. Next, we compute the cosine similarity between each text embedding and the corresponding patch feature (the output of the CLIP head). The class with the highest cosine similarity is selected as the predicted class for each patch. We then resize the patch class predictions to match the original image dimensions and calculate mIoU scores. The evaluation resolution is maintained at 448px for fair comparison with previous methods.

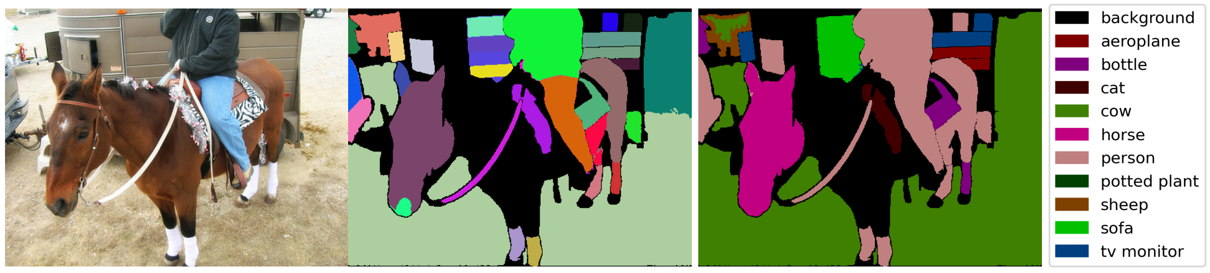
Composing CLIP and SAM Heads To combine both CLIP and SAM heads for zero-shot semantic segmentation, we first resize the image to 1024px and run the CLIP head to obtain mask predictions (i.e., logits) for each class. Subsequently, we pass the mask prediction corresponding to each class to the prompt encoder, along with 1-3 auto-generated points. These points are randomly sampled from pixels where the mask prediction logits exceed a specific threshold (for Pascal VOC, we find that a threshold of 0.5 is generally sufficient). The output from the prompt encoder is then fed to the SAM head (i.e., mask decoder) along with the patch token outputs from the ViT backbone. Finally, the mask decoder produces fine-grained mask prediction logits for each class, and we designate the class with the highest logit value as the predicted class for each pixel.

C.1. SAM−CLIP vs. SAM+CLIP

One may wonder if it is possible to compose pretrained SAM and CLIP in a pipeline for zero-shot semantic segmentation, and how the results compare with SAM−CLIP . We implemented the SAM+CLIP pipeline that passes segmentation masks predicted by SAM ViT-B (in the *segment-everthing* mode) to CLIP ViT-B/16 (DataComp-1B) for class prediction. From Table 7, one can clearly observe that the results on Pascal VOC reveal the unsatisfactory performance of the SAM+CLIP pipeline, which we attribute primarily to SAM's limited semantic understanding. SAM often segments parts of objects rather than the whole, and CLIP struggles to classify these segmented parts. See visualizations in Figure 7.

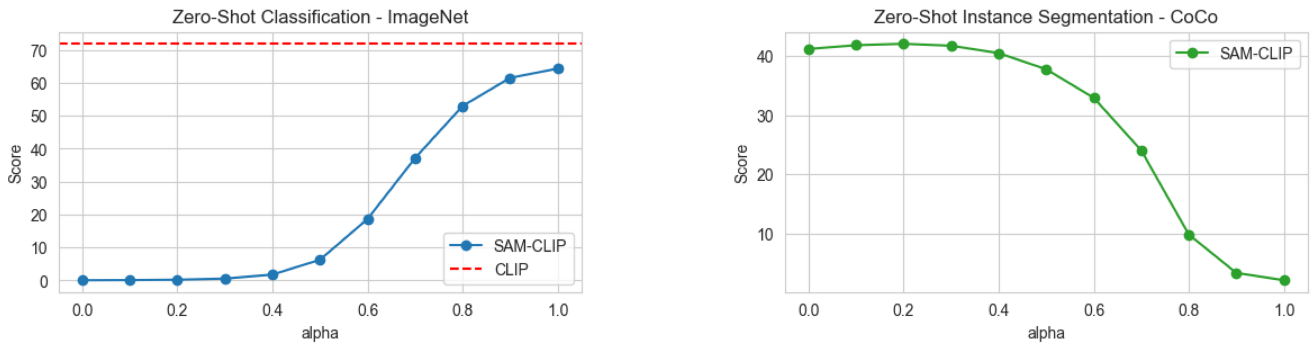
Table 7. Comparison of SAM−CLIP vs. SAM+CLIP

	SAM+CLIP	SAM−CLIP (CLIP-Head)	SAM−CLIP (Both Heads)
Pascal VOC (mIoU)	27.2	60.6	66.0



(a) Input image → (b) SAM outputs → (c) CLIP prediction

Figure 7. Visualization of the SAM+CLIP pipeline (see descriptions in Sec. C.1)



(a) Zero-Shot Accuracy (%) (b) Zero-Shot Instance Segmentation (mAP)

Figure 8. Wise-FT [86] to a CLIP-distilled SAM ViT-B model. The red dashed line marks the performance of the CLIP teacher model.

D. Weight Averaging

Weight averaging is a straightforward post-processing method proven to mitigate forgetting across a variety of fine-tuning tasks. Specifically, Wise-FT [86] proposes linearly interpolating the pretrained and fine-tuned parameters using a coefficient α. In this study, we explore the application of Wise-FT in our setup. We focus exclusively on CLIP distillation applied to SAM ViT-B (serving as the student model), with a CLIP ViT-B/16 model acting as the teacher model. The model is trained on ImageNet-21k for 20 epochs. It is evident that the fine-tuned student model ($\alpha = 1$) gains zero-shot classification capabilities at the expense of forgetting its original zero-shot instance segmentation abilities. Upon applying Wise-FT to the fine-tuned model, we observe an inherent tradeoff between learning and forgetting. Notably, no optimal point exists where both high classification accuracy ($> 60\%$ on ImageNet) and a high mAP (> 35 mAP on COCO) are achieved simultaneously.

E. Limitations

Our proposed method for merging existing foundational vision models may inherit the limitations of the original models. Specifically, our approach might carry over limitations from both the original SAM and CLIP models, including biases in data distribution. We have not assessed the robustness and fairness of our method in this work. Another potential limitation is the model size/architecture of the base VFM (SAM in this paper), which must be adopted from an existing model. However, we believe this should not be a practical limitation. The original SAM model offers several sizes/architectures (ViT-B/L/H). Moreover, follow-up works, such as MobileSAM [96], could be adopted as the base model in our proposed method to achieve a suitable final merged model. Additionally, our merged image encoder for the auxiliary model (CLIP in this case) requires an additional head (the CLIP-Head here). In this work, this increases the overall size by approximately 25% compared to a single ViT-B.