This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

T-SAM: Transductive Learning for Segment Anything Model

Rangel Daroya*[1] Deepak Chandran[2] Subhransu Maji[1] Andrea Fanelli[2]
[1]University of Massachusetts, Amherst [2]Dolby Laboratories
{rdaroya, smaji}@umass.edu, {deepak.chandran, andrea.fanelli}@dolby.com

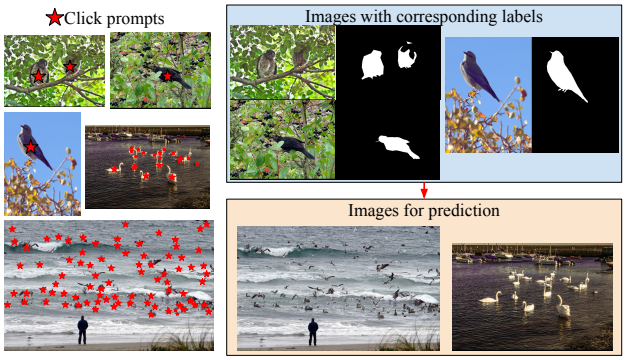
Abstract

The Segment Anything Model has been used for widespread applications by using user prompts to segment objects at test time. While this could be effective for segmenting images with a few samples, it becomes prohibitive when either labeling a lot of images or labeling images with multiple instances (e.g., segmenting a large group of cells or a migrating flock of birds). We introduce transductive learning for SAM (T-SAM) which uses a few labeled examples to learn a prompt representation that can generalize to other test images, including out-of-domain classes. T-SAM uses a cross-attention mechanism to learn distinguishing features from the labeled images to properly segment similar instances of the same object in other images. We also show an alternative method that uses the average representation based on previous labeled images. While applying model fine-tuning or utilizing memory banks have been introduced by previous works, the large number of parameters involved and storage constraints limit their applicability. We show our method performs competitively with baseline methods, and is able to scale with an increasing number of labeled images.

1. Introduction

The Segment Anything Model (SAM) [15] has been used for general segmentation tasks in various domains. It has been used to segment cells in medical images [20, 21], plants in agricultural images [11, 27], man-made structures in satellite images [26, 38] and many others [1, 5, 22]. The use of foundation models like SAM allows faster adoption for different applications. The user can simply click on the object of interest in an image, and the full object can be isolated from the rest of the image. However, since segmenting an object relies on user clicks and other similar prompts, recent works have emerged with the aim of reducing or eliminating this dependency [20, 25, 37].

Using a limited set of annotations, different methods apply SAM without having to specify a prompt for each image [3, 25, 33, 34, 37]. Works range from applying parameter efficient fine-tuning on SAM [37], adding a memory bank from which prompting information can be retrieved [25], and utilizing in-context learning to define the task to be applied to test images [33]. A common factor present in these works is that they use a few labeled samples without needing human interaction for every image at test time.



(a) SAM with manual prompts (b) Transductive learning from labeled images

Figure 1. **The Segment Anything Model (SAM) [15] (a) requires manual prompts for segmenting objects at test time.** We propose to utilize transductive learning (b) which takes a few labeled images to predict on other images at test time, requiring less effort (i.e., point clicks) when applying the model on unseen datasets and classes.

In line with this, we introduce transductive learning for SAM (**T-SAM**), which updates a limited set of parameters using a small number of labeled examples (Fig. 1). Instead of manually annotating multiple images, providing a few high-quality annotations could lead to better quality outputs while reducing the amount of needed effort. This could be particularly useful when presented with an image that contains multiple instances of the same object. For example, for an image with more than 100 birds where the bird would need to be segmented, using SAM out of the box would require around 100 clicks to segment all the birds in the image (Fig. 1a, bottom image). In contrast, by providing a few examples to learn from, the model can segment all bird instances in the image without manual annotations (Fig. 1b).

*Work was done during an internship at Dolby Laboratories

As a contribution of **T-SAM**, we introduce a new approach that uses a cross-attention mechanism [8] to learn patterns from labeled images, and applies these same patterns to the unlabeled test images for prediction. We also explore an alternative method that uses the average representation from labeled images to inform unlabeled image predictions. Unlike previous works that use memory banks to store a limited number of image embeddings [2, 12, 15], we learn two sets of embeddings to represent relevant image features and corresponding prompt embeddings, respectively. New and unlabeled images can be queried against these learned embeddings. We show that our method scales with more than a hundred labeled examples (Fig. 4).

In summary, we make the following contributions: (1) we introduce a transductive learning method on SAM (T-SAM) that uses either a cross-attention mechanism or an average representation to predict on unlabeled images using a few labeled examples. (2) We show T-SAM is competitive with other baseline methods that learn from labeled examples. (3) We present a thorough evaluation of our method across different datasets and analyze its performance across various classes with different amounts of labeled images.

2. Related Work

Model Fine-tuning. While a typical way of adapting an existing model to new domains involves fine-tuning the model [20, 21], this usually requires a lot of resources due to the large number of parameters in existing foundation models like SAM [23, 39]. Parameter efficient fine-tuning [10, 13, 17] is an alternative way to overcome the high computation required by fine-tuning a small subset of parameters [4, 36]. However, these methods still need a significant number of labeled data to be effective. Per-SAM [37] introduces a reference-based visual prompting mechanism, where a single annotated mask guides the one-shot segmentation of similar objects in new images enabling adaptation to novel object instances.

Visual prompting. Previous works have tried to be more efficient with using a few labeled examples to predict on other inputs at test time [28–30]. Diffusion-based in-context learning approaches re-frame vision tasks as inpainting problems, where missing parts of an image—representing segmentation masks, depth maps, or keypoints—are generated based on a few-shot visual prompt [3, 33–35]. Visual Prompting via Inpainting [3] first introduced this concept, demonstrating that a pre-trained inpainting model could generalize across tasks by reconstructing missing outputs. Instead of explicitly training for segmentation or depth estimation, the model fills in missing output pixels based on examples provided in the context (i.e., the labeled images). Painter [33] and SegGPT [34] expand on this idea by redefining all vision tasks as image-to-image transformations, where outputs—such as segmentation masks or keypoints—are generated as images. This enables strong zero-shot generalization but requires iterative sampling (due to the use of diffusion models) making it slow. Prompt Diffusion [35] enhances this by integrating multi-modal conditioning, allowing the model to process both text and image prompts. This improves generalization to new tasks but further increases computational cost, limiting real-time applicability. Large Language Models (LLMs) offer a different approach by interpreting structured prompts and applying language-driven reasoning to vision tasks. Instead of retrieving neighbors or generating images, LLM-based methods [7, 31, 40] treat vision tasks as language processing problems, leveraging textual descriptions and structured prompts to predict on unlabeled images. However, these methods require pre-training on diverse visual tasks for strong generalization.

Nearest-neighbor retrieval. Nearest neighbor (NN) retrieval methods adapt to different tasks by leveraging a non-parametric memory bank to store and retrieve feature representations at inference time [2, 6, 14, 19, 32]. These methods do not require updating model weights but instead rely on retrieving similar examples from past experiences to guide predictions for new queries. Hummingbird [2] applies this retrieval-based approach to dense vision tasks such as semantic segmentation and depth estimation by encoding images into spatial feature maps using a vision transformer (ViT) [9]. SAM 2 [25] operates similarly with image embeddings stored in a memory bank. At inference time, query patches retrieve similar features from the memory, interpolating their labels to produce an output. This enables task-agnostic adaptation but introduces memory overhead and retrieval inefficiencies as the dataset scales. kNN-CLIP [12] extends Hummingbird to open-vocabulary segmentation by leveraging the CLIP [24] feature space. This allows training-free adaptation to novel categories, but it faces challenges with scalability and real-time inference speed as the dataset grows. Our proposed method T-SAM is closest to these works, but instead of relying on memory banks that could have limited capacity, we propose to learn two sets of embeddings to represent similar features and the corresponding model response or output, respectively. Doing this allows our method to easily adapt to an increasing number of examples, which could be important for cases that need more demonstrations. For example, segmenting different bird species across different images might require showing different bird images and corresponding segmentation labels for the model to generalize.

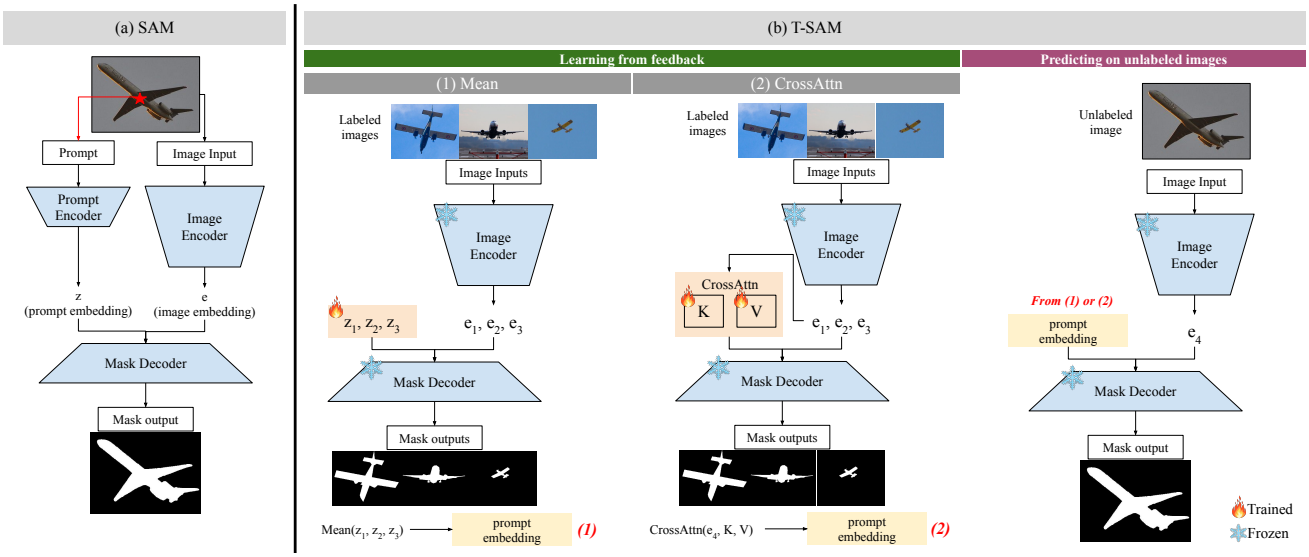


Figure 2. **Architecture of (a) SAM with manual prompting per image and (b) our proposed T-SAM that learns from feedback and reduces the need for manual prompt on all images**. (a) Using SAM on new domains and object classes requires manual prompting (e.g., point clicks) to specify the object to be segmented. This easily becomes labor-intensive and inefficient for larger sets of images. (b) We introduce **T-SAM**, a more efficient solution that minimizes the manual prompting by learning prompt embeddings from a few labeled examples. (1) The **Mean** method learns a prompt embedding for each image based on the labeled examples, such that the model outputs the correct segmentation. At inference time, Mean uses the average prompt embedding across all previously labeled images as the prompt embedding for all unlabaled images. (2) The **CrossAttn** method learns two sets of embeddings: key (K) and value (V). Using cross-attention, each image input serves as the query, and the prompt embedding is produced by combining information from the key, value, and image input itself. This results in a more adaptive way to generate prompt embeddings.

3. Method

3.1. Problem Formulation

We define the problem as follows: given a pre-trained SAM [15] f_θ, a set of unlabeled images $\mathcal{U} = \{\mathbf{I}_1^U, \ldots, \mathbf{I}_n^U\}$, a set of labeled images $\mathcal{V} = \{\mathbf{I}_1^V, \ldots, \mathbf{I}_m^V\}$, and the corresponding labels $\mathcal{Y}^V = \{\mathbf{Y}_1^V, \ldots, \mathbf{Y}_m^V\}$, the goal is to adapt f_θ such that it can predict the correct segmentation masks of $\mathcal{U}$ using $\mathcal{V}$ and $\mathcal{Y}^V$. The unlabeled images $\mathcal{U}$ and labeled images $\mathcal{V}$ come from the same domain through the test set of a dataset such that $\mathcal{U} \cap \mathcal{V} = \emptyset$. Each image is defined as $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ with corresponding segmentation masks $\mathbf{Y} \in \{0,1\}^{W \times H}$. We propose to optimize a small set of parameters ϕ to efficiently adapt to the labeled examples $\mathcal{V}$. In this way, we can adapt SAM to possibly different domains and distributions at inference time through minimal feedback and training.

3.2. Segment Anything Model

Segment Anything Model (SAM) [15] applies a prompt encoder and image encoder to a given prompt and image, respectively. Fig. 2a shows that the corresponding embeddings—$\mathbf{z}$ for prompt embedding and $\mathbf{e}$ for image embedding—are then fed to a lightweight mask decoder, which outputs a valid segmentation mask. This process relies on the presence of a prompt for each input image. However, the reliance on prompts is inefficient and not scalable when given a large set of images that need to be segmented. In the absence of prompts, the outputs of SAM can be unpredictable especially for out-of-domain classes or classes that SAM did not see during training.

3.3. T-SAM

With the limitation of SAM needing manual prompts to segment specific objects at test time, we propose an automated way of *predicting the prompt representations* given previously labeled examples $\mathcal{V}$. Given a set of unlabeled images $\mathcal{U} = \{\mathbf{I}_1^U, \ldots, \mathbf{I}_n^U\}$, we want to predict the corresponding segmentation masks $\hat{\mathcal{Y}}^U = \{\hat{\mathbf{Y}}_1^U, \ldots, \hat{\mathbf{Y}}_n^U\}$ that are close to ground truth segmentations $\mathcal{Y}^U = \{\mathbf{Y}_1^U, \ldots, \mathbf{Y}_n^U\}$ by predicting the corresponding prompt embeddings $\{\mathbf{z}_1^U, \ldots, \mathbf{z}_m^U\}$. We propose two methods of predicting these prompt embeddings: (1) Mean and (2) Cross-Attention. We discuss them below.

3.3.1. Mean

We want to use the previously labeled images $\mathcal{V}$ to inform predictions on unlabeled images $\mathcal{U}$. That is, given m labeled images, we want to learn corresponding prompt embedding for each image: $\{\mathbf{z}_1^V, \ldots, \mathbf{z}_m^V\}$ such that the predicted segmentation masks are close to the ground truth segmenta-

tions. Each $\mathbf{z}_i^V$ is a learnable embedding, and $\mathbf{e}_i^V$ is the image embedding of image i. That is, for all $i \leq m$:

$$\mathbf{z}_i^V = \underset{\mathbf{z}_i^V}{\operatorname{argmin}} \mathcal{L}(\hat{\mathbf{Y}}_i^V, \mathbf{Y}_i^V) \quad \text{s.t. } \hat{\mathbf{Y}}_i^V = \text{dec}(\mathbf{z}_i^V, \mathbf{e}_i^V) \quad (1)$$

The 'dec' refers to the mask decoder in Fig. 2. We use binary cross entropy as the loss $\mathcal{L}$ in Eq. (1). At inference time, we then use the mean of all n prompt embeddings as the prompt embedding (Fig. 2b.1). That is, for all unlabeled images $\mathbf{I}_j^U$ where $j \leq n$:

$$\mathbf{Y}_j^U = \text{dec}(\mathbf{z}_j^U, \mathbf{e}_j^U) \quad \text{s.t. } \mathbf{z}_j^U = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i^V \quad (2)$$

This method relies on the idea that the target object being segmented is similar across the unlabeled images. Given the visual similarity of objects, the way of prompting the model to segment the target object should also be similar.

3.3.2. Cross-Attention

Similar to the Mean method, we use labeled images $\mathcal{V}$ to predict on unlabeled images $\mathcal{U}$ by predicting prompt embeddings. However, unlike the previous method, we use a cross-attention mechanism [8] to predict the corresponding prompt embeddings. Based on labeled images, we want to learn how to map image embeddings to corresponding prompt embeddings.

Given a set of labeled images, we learn an optimal representation of keys $\mathbf{K} \in \mathbb{R}^{256\times d}$ and values $\mathbf{V} \in \mathbb{R}^{d\times 256}$. Cross-attention is applied to the image embedding $\mathbf{e}_i^V$, and the learnable embeddings $\mathbf{K}$ and $\mathbf{V}$ to produce a representation $\mathbf{p}_i^V$. The prompt embedding $\mathbf{z}_i^V$ is predicted by applying a linear layer g parametrized by ψ on $\mathbf{p}_i^V$.

$$\mathbf{z}_i^V = g_\psi(\mathbf{p}_i^V) \quad \text{s.t.} \quad \mathbf{p}_i^V = \text{CA}(\mathbf{e}_i^V, \mathbf{K}, \mathbf{V}) \quad (3)$$

$$\text{CA}(\mathbf{e}_i^V, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{e}_i^V \mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \quad (4)$$

Using the prompt embedding $\mathbf{z}_i^V$, the decoder should output a segmentation that is similar to the ground truth (Fig. 2b.2). That is, for all labeled images:

$$\psi, \mathbf{K}, \mathbf{V} = \underset{\psi,\mathbf{K},\mathbf{V}}{\operatorname{argmin}} \sum_{i=1}^{m} \mathcal{L}\left(\hat{\mathbf{Y}}_i^V, \mathbf{Y}_i^V\right) \quad \text{s.t.} \quad \hat{\mathbf{Y}}_i^V = \text{dec}(\mathbf{z}_i^V, \mathbf{e}_i^V) \quad (5)$$

We use binary cross entropy as the loss $\mathcal{L}$. At inference time, we then use the learned $\psi, \mathbf{K}, \mathbf{V}$ to predict prompt embeddings for all unlabeled images $\mathbf{I}_j^U$ which is then used to predict segmentation masks $\mathbf{Y}_j^U$ for all $j \leq n$:

$$\mathbf{Y}_j^U = \text{dec}(\mathbf{z}_j^U, \mathbf{e}_j^U) \quad \text{s.t. } \mathbf{z}_j^U = g_\psi\left(\text{CA}(\mathbf{e}_j^U, \mathbf{K}, \mathbf{V})\right) \quad (6)$$

The cross-attention method (CrossAttn) takes advantage of similar image features found between labeled and unlabeled images. They key $\mathbf{K}$ represents a source of similar image features learned from previously seen labeled images, queried by the image embedding $\mathbf{e}_j^U$. The value $\mathbf{V}$ provides the corresponding prompts associated with the retrieved image features. By identifying these similarities, the model can treat matching features as likely having similar prompt embeddings. This enables the prompt embeddings to align with these shared and distinct image features, using the same embeddings for similar parts of the image.

3.4. Implementation Details

We train and evaluate our model on an NVIDIA L40s GPU. For each additional annotation, it takes an average of less than a minute to train the learnable parameters. The CrossAttn method uses a hidden dimension $d = 64$ for the learnable keys and values. The Adam optimizer is used for training both methods with a learning rate of 0.01 and trained with 100 epochs.

4. Experiments

4.1. Baselines

The following baselines were used to compare the effectivity of our proposed method on learning from continuous feedback.

Manual refers to performance when a human labels each sample image individually. Here, each labeled image represents a human-annotated example. As more labeled segmentation masks become available, manual performance improves. However, since learning does not occur, performance is limited by the number of labeled images provided.

SAM [15] refers to the SAM model without any feedback or prompting. Here, we measure performance in the absence of any human feedback.

SAM 2 [25] is a recently released update to the Segment Anything model designed to process video frames and segment a specific object across the entire video based on prompts. These prompts can come from a single frame or multiple frames. In our case, we use the labeled segmentation masks as the prompts. These labels are embedded and stored in a memory bank that SAM 2 uses for conditioning other images or frames to make predictions. The memory bank uses an attention mechanism to retrieve relevant features at inference time.

4.2. Datasets

The following segmentation datasets were used to evaluate segmentation performance, with multiple examples per

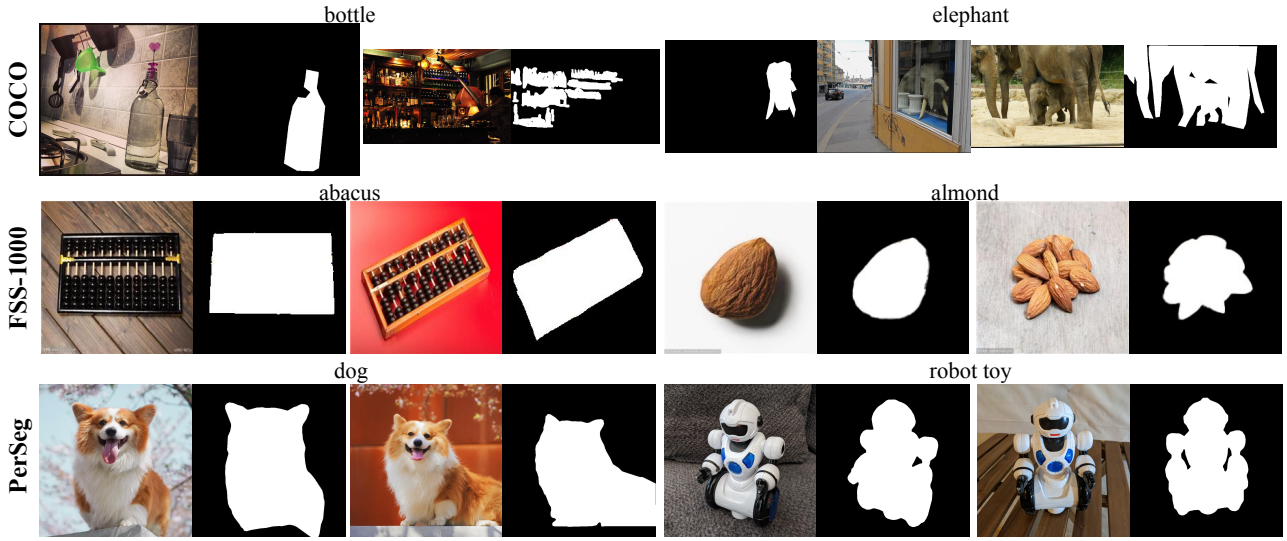


Figure 3. **Overview of datasets used for evaluation with the images and corresponding segmentation labels**. COCO and FSS-1000 cover different types of objects belonging to the same class, while PerSeg shows the same object given the same class.

class. Performance for each method is assessed as the number of examples increases. Fig. 3 shows samples from the different datasets.

COCO [18] has 80 object classes with up to 200 examples per class. Each class contains a diverse set of objects from that category, presented in different environment settings. We use the validation split of the 2017 version in our evaluation due to the availability of the labeled images.

FSS-1000 [16] has 1000 object classes with 10 examples per class. The dataset was made for few-shot object segmentation where each class contains varying objects of the same category, similar to COCO. Images are all of the same size.

PerSeg [37] has 40 object classes with up to 12 examples per class. The number of examples vary from 5 to 12 per class, and each class shows the same object in different environments or backgrounds.

4.3. Metric

We use mIoU as the metric to evaluate performance of the different methods, which is the mean IoU over all classes. Eq. (7) shows how IoU is computed given two segmentation masks $\hat{\mathbf{Y}}_i$ and $\mathbf{Y}_i$. Models that perfectly segment a given object will have an IoU of 1.0.

$$IoU = \frac{\hat{\mathbf{Y}}_i \cap \mathbf{Y}_i}{\hat{\mathbf{Y}}_i \cup \mathbf{Y}_i} \tag{7}$$

5. Results and Discussion

5.1. Continuously learning from examples

T-SAM can learn from feedback. Fig. 4 shows the segmentation performance of different methods as the number of labeled examples increases. As more feedback is given to the model, the performance also steadily increases. mIoU is displayed for up to 200 labeled images, with the plot demonstrating that T-SAM can adapt to feedback with performance consistently higher than manual labeling and SAM. Due to the model's ability to match information about previously labeled images, it can inform and guide predictions on other unlabeled images.

Cross-attention performs better than learning a mean representation. For all plots in Fig. 4, we see the cross-attention based method (CrossAttn) consistently performing better than the mean-based method (Mean). Due to the ability of CrossAttn to store more diverse information about the previously seen data and how it relates to the labels, it can perform better. Through this, the model can adapt to the different types of inputs. In contrast, Mean simply tries to learn an overall representation that can help with all test images, which would work well likely only for a subset of images.

T-SAM (CrossAttn) does better with more examples. While SAM2 can do better on fewer labeled images (Fig. 4 PerSeg and FSS-1000), performance of T-SAM generally does better with more labeled examples (after 10 labeled examples in PerSeg, and after around 6 labeled examples

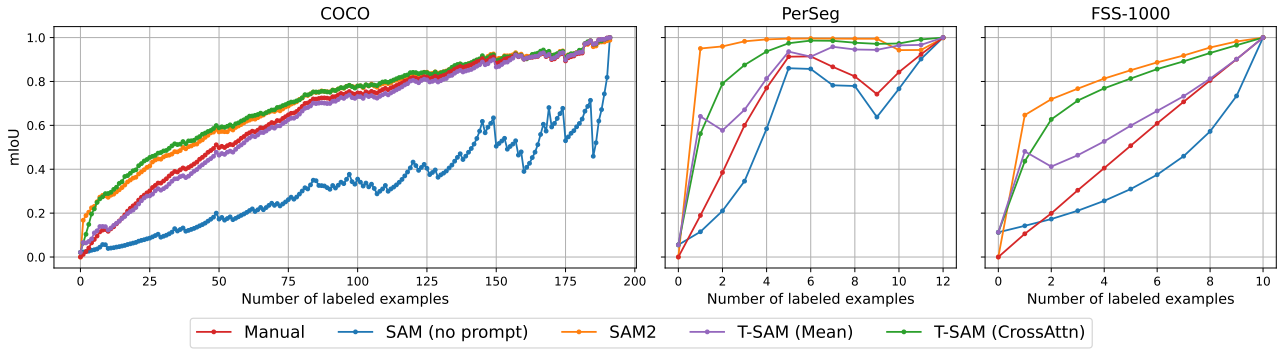


Figure 4. **Segmentation performance on the test set of the corresponding datasets as more examples are introduced**. Performance is measured across the entire test set as images get labeled one at a time. Each point on the plot shows the performance of the model taking into account that it already has the correct output for images that have been labeled so far, thus we see performance eventually reach 1.0. The right end of the x-axis represents the point when all images have been labeled. Comparing the different methods, we see T-SAM (CrossAttn) performs competitively with other baselines and significantly outperforms manual labeling. For COCO, T-SAM (CrossAttn) generally demonstrates higher performance with the same number of labeled examples compared to other methods.

	COCO [18]			FSS-1000 [16]			PerSeg [37]		
	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
SAM [15]	0.023	0.027	0.033	0.142	0.211	0.310	0.115	0.346	0.860
Manual	0.012	0.041	0.080	0.106	0.304	0.507	0.190	0.600	0.913
SAM 2 [25]	**0.168**	**0.204**	**0.231**	**0.647**	**0.767**	**0.851**	**0.950**	**0.983**	**0.995**
T-SAM (Mean)	0.066	0.071	0.110	0.481	0.464	0.599	0.641	0.671	0.936
T-SAM (Crossattn)	0.063	0.149	0.219	0.436	0.713	0.813	0.562	0.875	0.974

Table 1. **Few-shot results across all classes in datasets**. mIoU is reported for all datasets and methods. While T-SAM struggles with 1-shot performance, the gap between SAM2 and T-SAM performance becomes smaller with more examples (5-shot performance). The best performance is in bold, while the second best is underlined.

for COCO). This could be attributed to the use of memory banks in SAM2 that store a certain number of frames for context, which might be less effective for longer contexts spread out through more frames. We also find that T-SAM outperforms SAM and the Manual baseline, indicating that our model can indeed learn from examples. Surpassing these baselines suggests that our model introduces meaningful improvements that would otherwise not be achieved.

Trade-off on performance and labeling cost can be estimated. By using a framework such as ours that enables continuous feedback, resource-limited applications can opt to only label a subset of images until an acceptable level of performance is reached (perhaps by visually checking outputs). This is useful for applications requiring expert annotations where each image needs to be carefully annotated. It is also useful when multiple instances of the same object need to be segmented in a single image. Through experiments (Fig. 4), we show that our method can adapt to varying number of inputs with steadily increasing performance.

5.2. Few-shot performance

While T-SAM was designed for learning from continuous feedback, few-shot and one-shot performance are also promising applications.

T-SAM can segment multiple instances of the same object. Fig. 5 show sample visualizations with 10 labeled examples on the COCO dataset. While other baselines generally struggle with segmenting target objects, T-SAM (CrossAttn) shows promising results. Even for cases where there are multiple instances of the same object (4th column in Fig. 5), T-SAM (CrossAttn) can segment all instances, while other methods struggle to do the same.

T-SAM improves with more examples despite initial limitations. While T-SAM does not perform as well on a small number of examples (Tab. 1), it recovers with more examples. Comparing results in 1-shot and 5-shot, we see that T-SAM reduces the gap when presented with five examples. This could be attributed to the more diverse set

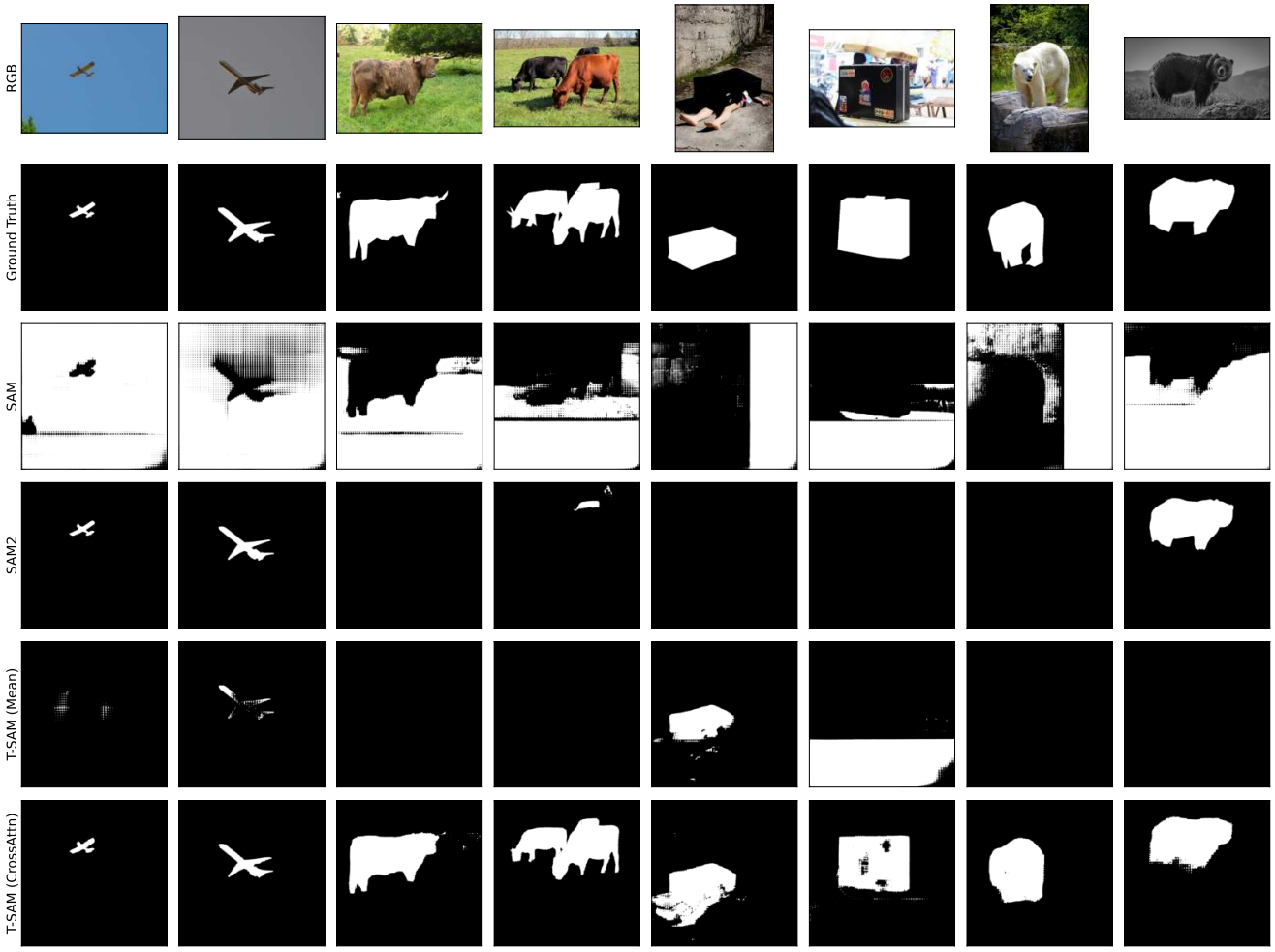


Figure 5. **Qualitative results on COCO test images given 10 labeled examples**. Note that images were padded to have the same width and height as expected by a pre-trained Segment Anything model. While SAM2 can effectively learn from labeled examples, its design for video segmentation limits its performance on static images with varying contexts. In contrast, T-SAM (CrossAttn) can adapt to different contexts.

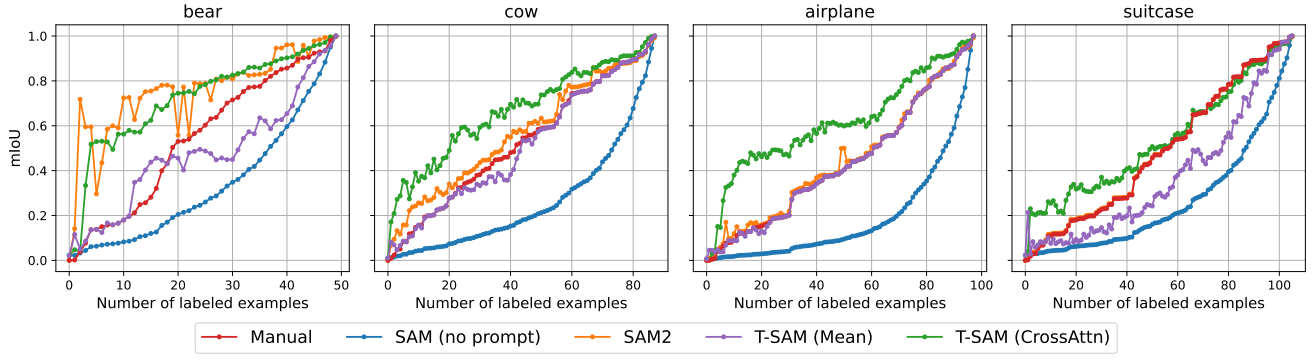


Figure 6. **Performance as the number of labeled examples increases on sample COCO classes**. While the average performance on few-shot is limited, T-SAM shows good performance on some classes shown in this plot.

	COCO [18]			FSS-1000 [16]			PerSeg [37]		
	fire hydrant	donut	motorcycle	backpack	chair	red cartoon	parthenon	prayer rug	roller skate
SAM [15]	0.005	0.067	0.010	0.111	0.160	0.069	0.437	0.250	0.095
Manual	0.013	0.036	0.004	0.125	0.263	0.240	0.031	0.131	0.116
SAM 2 [25]	0.013	0.056	0.019	0.235	0.769	**0.986**	0.106	0.131	0.166
T-SAM (Mean)	0.016	0.199	0.157	**0.729**	0.515	0.971	**0.898**	**0.910**	**0.880**
T-SAM (CrossAttn)	**0.244**	**0.270**	**0.189**	0.482	**0.787**	0.980	0.884	0.851	0.865

Table 2. **One-shot results on sample classes**. mIoU is used as the metric. T-SAM performs well on certain classes compared to baseline methods.

of labeled examples (and thus more image features) available after five examples, from which T-SAM can more effectively predict corresponding prompt embeddings. Fig. 6 also supports this, where we see that for certain classes, T-SAM (CrossAttn) can outperform SAM2 when more labeled examples are available.

5.3. One-shot performance

While we showed limited initial performance with a few examples, T-SAM performs well on some classes even with just a single labeled image. These are classes that typically cover a large area in the image (some examples are available in Fig. 6 and Tab. 2).

T-SAM (CrossAttn) performs well on domain shifts. Tab. 2 shows one-shot performance on several classes in COCO, FSS-1000, and PerSeg. In some cases, we see T-SAM (CrossAttn) outperforming all baselines. Particularly for COCO, we find that it outperforms even SAM 2 in some classes. We hypothesize that this is due to the clear distinction between the objects in the COCO images, that prevent our model from confusing the target class with other objects in the image. In addition, images of different objects in COCO usually involve different types of the same object. For example, the bear class could involve segmenting polar bears, blacks bears, and brown bears which all look slightly different and appear in different environments (see last two columns of Fig. 5). This could be particularly challenging for models like SAM 2 which are suited for processing video frames which do not necessarily have a lot of context change between frames.

T-SAM (Mean) does well on consistent image contexts. T-SAM (Mean) does particularly well on PerSeg classes (Tab. 2). In PerSeg, for a given class, the dataset shows the same object in different environments with varying backgrounds. Since the same instance of objects are usually used in different settings, similar representations of the same object can be used for all images, and is thus where the Mean method does well. Nonetheless, CrossAttn still performs reasonably well with performance close the Mean on PerSeg.

6. Limitations

While we show our method can learn from labeled examples, our method can struggle to precisely separate objects that are close to each other. For example, when segmenting the luggage in Fig. 5 (last row, 5th column), the person next to the luggage ends up getting segmented. In addition, the reliance of T-SAM on high-quality annotations can be limiting for some applications. Future work can explore and improve on these cases. Our model is also currently limited to segmentation, but a similar mechanism can be applied to other models that require prompting at test time.

7. Conclusion

We presented two transductive learning methods applied to SAM: CrossAttn and Mean. Each method was found to have different strengths—CrossAttn excels at segmenting objects that appear in different forms and context, while Mean is more effective for segmenting the same object across different backgrounds. T-SAM presents an alternative way to minimize human intervention for the Segment Anything Model at test time. This can be particularly useful for domain-specific applications that require segmenting only specific objects. By leveraging a small number of high-quality labeled images, the model can identify similar patterns in other test images, enabling SAM to scale to new classes and domains more effectively. As the number of foundation models continue to increase, many are designed to generalize across tasks while still requiring user prompts to specify the task. Our work presents a way to bypass this requirement, simplifying adoption of these types of models to different applications.

References

[1] Mohsen Ahmadi, Ahmad Gholizadeh Lonbar, Hajar Kazemi Naeini, Ali Tarlani Beris, Mohammadsadegh Nouri, Amir Sharifzadeh Javidi, and Abbas Sharifi. Application

of segment anything model for civil infrastructure defect assessment. *arXiv preprint arXiv:2304.12600*, 2023. 1

[2] Ivana Balazevic, David Steiner, Nikhil Parthasarathy, Relja Arandjelović, and Olivier Henaff. Towards in-context scene understanding. *Advances in Neural Information Processing Systems*, 36:63758–63778, 2023. 2

[3] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022. 1, 2

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[5] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023. 1

[6] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, 2022. 2

[7] Yi-Syuan Chen, Yun-Zhu Song, Cheng Yu Yeo, Bei Liu, Jianlong Fu, and Hong-Han Shuai. Sinc: Self-supervised in-context learning for vision-language tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15430–15442, 2023. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weis-senborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. Attention is all you need. In *International Conference on Learning Representations*, 2021. 2, 4

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[10] Jonathan Frankle, David J Schwab, and Ari S Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. In *International Conference on Learning Representations*, 2021. 2

[11] Baoling Gui, Anshuman Bhardwaj, and Lydia Sam. Evaluating the efficacy of segment anything model for delineating agriculture and urban green spaces in multiresolution aerial and spaceborne remote sensing images. *Remote sensing*, 16 (2):414, 2024. 1

[12] Zhongrui Gui, Shuyang Sun, Runjia Li, Jianhao Yuan, Zhaochong An, Karsten Roth, Ameya Prabhu, and Philip Torr. knn-clip: Retrieval enables training-free segmentation on continually expanding large vocabularies. *Transactions on Machine Learning Research*, 2024. 2

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2

[14] Ahmet Iscen, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. Retrieval-enhanced contrastive vision-text models. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 1, 2, 3, 4, 6, 8

[16] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2869–2878, 2020. 5, 6, 8

[17] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022. 2

[18] 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. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. 5, 6, 8

[19] Haotian Liu, Kilho Son, Jianwei Yang, Ce Liu, Jianfeng Gao, Yong Jae Lee, and Chunyuan Li. Learning customized visual models with retrieval-augmented knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15148–15158, 2023. 2

[20] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024. 1, 2

[21] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023. 1, 2

[22] Lucas Prado Osco, Qiusheng Wu, Eduardo Lopes De Lemos, Wesley Nunes Gonçalves, Ana Paula Marques Ramos, Jonathan Li, and José Marcato Junior. The segment anything model (sam) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation*, 124:103540, 2023. 1

[23] Zelin Peng, Zhengqin Xu, Zhilin Zeng, Lingxi Xie, Qi Tian, and Wei Shen. Parameter efficient fine-tuning via cross block orchestration for segment anything model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3743–3752, 2024. 2

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[25] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman

Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 2, 4, 6, 8

[26] Simiao Ren, Francesco Luzi, Saad Lahrichi, Kaleb Kassaw, Leslie M Collins, Kyle Bradbury, and Jordan M Malof. Segment anything, from space? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8355–8365, 2024. 1

[27] Jialin Sun, Shuai Yan, Thomas Alexandridis, Xiaochuang Yao, Han Zhou, Bingbo Gao, Jianxi Huang, Jianyu Yang, and Ying Li. Enhancing crop mapping through automated sample generation based on segment anything model with medium-resolution satellite imagery. *Remote Sensing*, 16(9): 1505, 2024. 1

[28] Yanpeng Sun, Qiang Chen, Xiangyu He, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Jian Cheng, Zechao Li, and Jingdong Wang. Singular value fine-tuning: Few-shot segmentation requires few-parameters fine-tuning. *Advances in neural information processing systems*, 35:37484–37496, 2022. 2

[29] Yanpeng Sun, Jiahui Chen, Shan Zhang, Xinyu Zhang, Qiang Chen, Gang Zhang, Errui Ding, Jingdong Wang, and Zechao Li. Vrp-sam: Sam with visual reference prompt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23565–23574, 2024.

[30] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 44(2):1050–1065, 2020. 2

[31] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021. 2

[32] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2725–2736, 2023. 2

[33] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023. 1, 2

[34] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Towards segmenting everything in context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1130–1140, 2023. 1, 2

[35] Zhendong Wang, Yifan Jiang, Yadong Lu, Pengcheng He, Weizhu Chen, Zhangyang Wang, Mingyuan Zhou, et al. In-context learning unlocked for diffusion models. *Advances in Neural Information Processing Systems*, 36:8542–8562, 2023. 2

[36] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 2

[37] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. In *ICLR*, 2024. 1, 2, 5, 6, 8

[38] Xin Zhang, Yu Liu, Yuming Lin, Qingmin Liao, and Yong Li. Uv-sam: Adapting segment anything model for urban village identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 22520–22528, 2024. 1

[39] Zihan Zhong, Zhiqiang Tang, Tong He, Haoyang Fang, and Chun Yuan. Convolution meets lora: Parameter efficient finetuning for segment anything model. *arXiv preprint arXiv:2401.17868*, 2024. 2

[40] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in neural information processing systems*, 36:19769–19782, 2023. 2