Repurposing SAM for User-Defined Semantics Aware Segmentation

Rohit Kundu[1] , Sudipta Paul[2,*] , Arindam Dutta[1] , Amit K. Roy-Chowdhury[1]

[1] University of California, Riverside, [2] Samsung Research America

[*] *Work done while at UCR*

Correspondence: rohit.kundu@email.ucr.edu

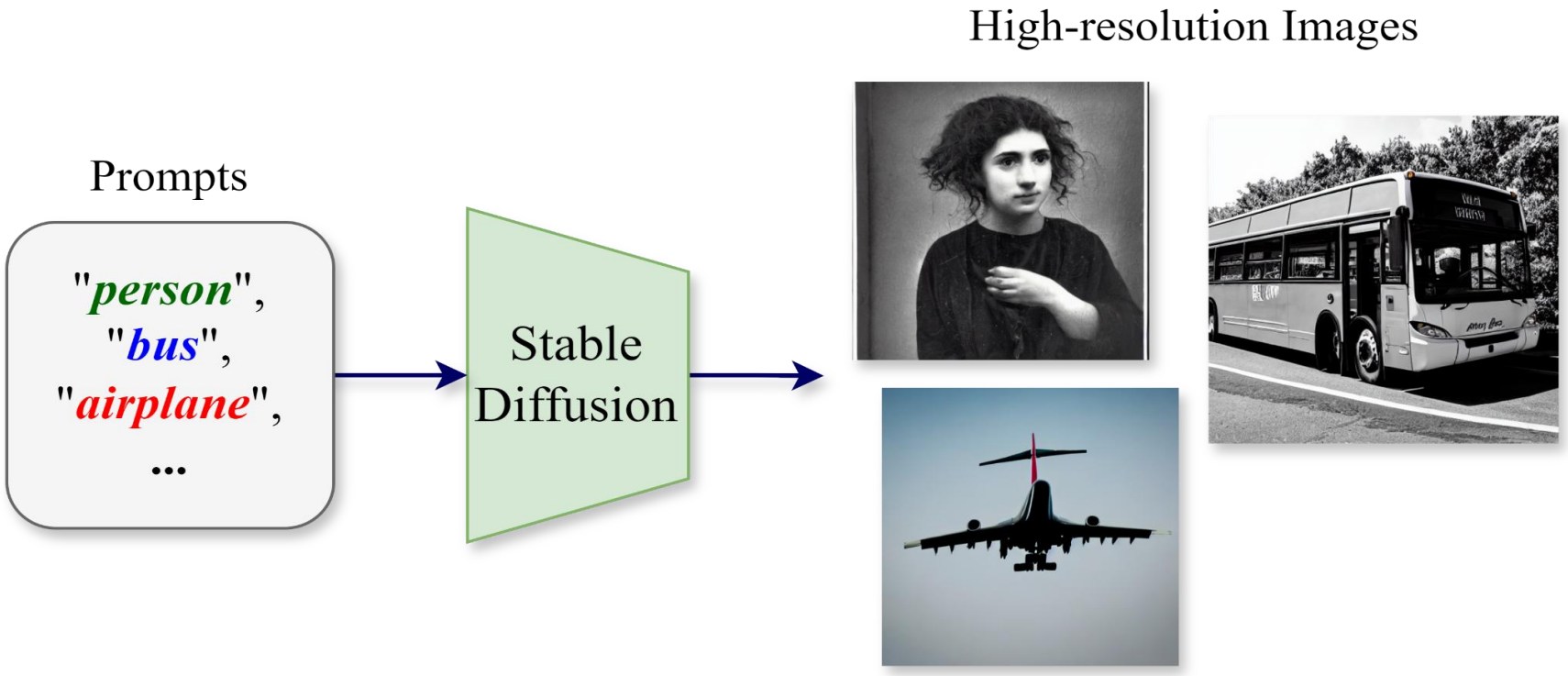
Or Visit: https://rohit-kundu.github.io/U-SAM

UCR Vision and Learning Group

Problem Statement

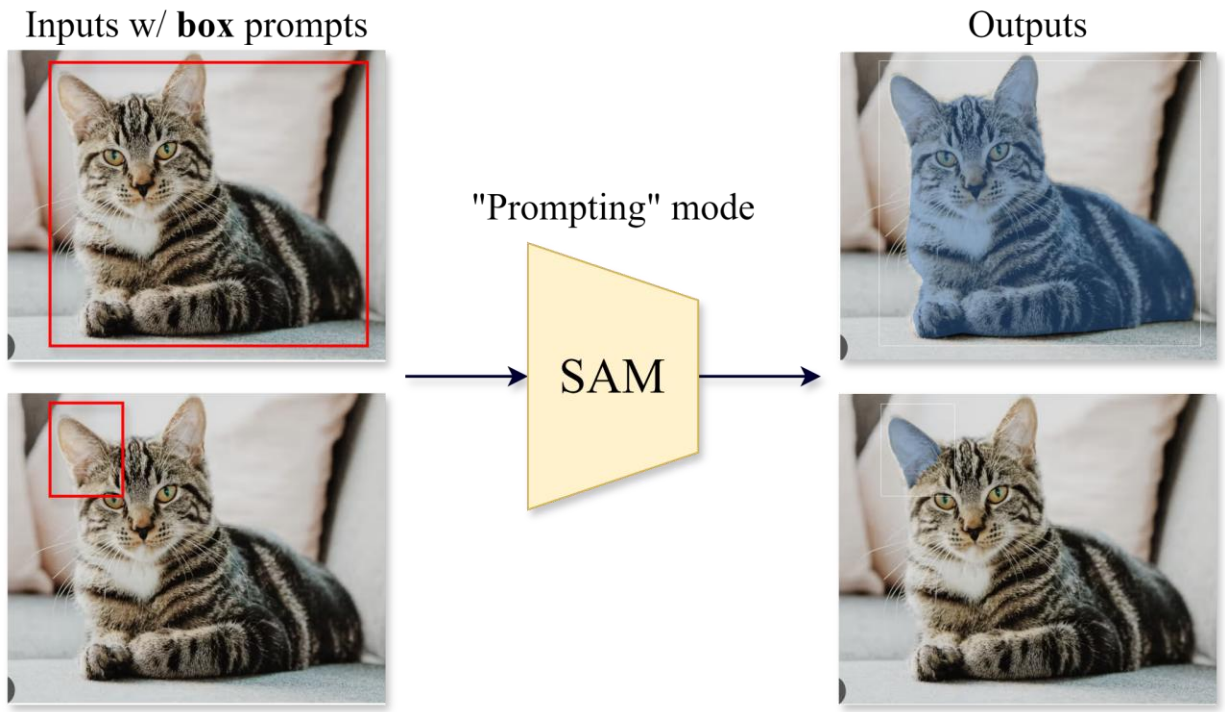
- Existing semantic segmentation methods in the literature rely on labeled/unlabeled training data.
- Such models need to be retrained/adapted each time a new test dataset is presented.
- Can we perform reliable semantic segmentation just by knowing which classes need to be segmented?
- Can we eliminate the need for retraining if the label space is identical?

Problem Statement



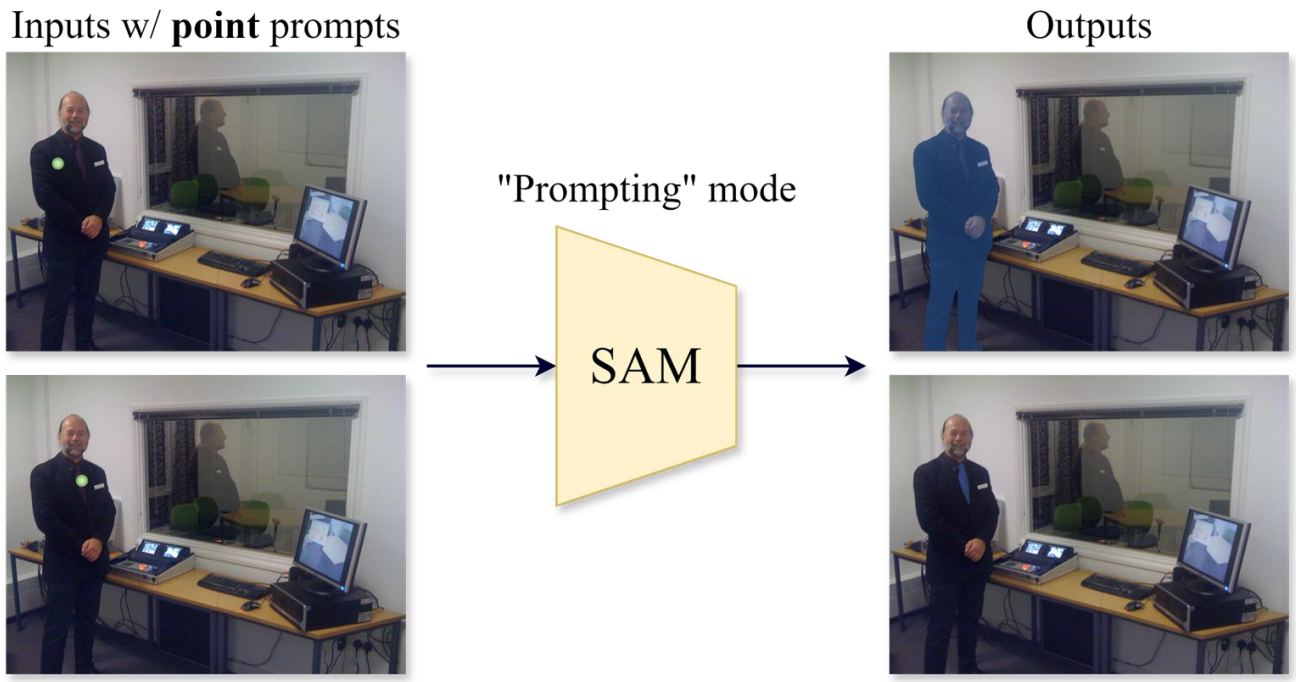
- High quality image generation models exist, but they are not used in the literature to perform downstream tasks.

Segment Anything Model



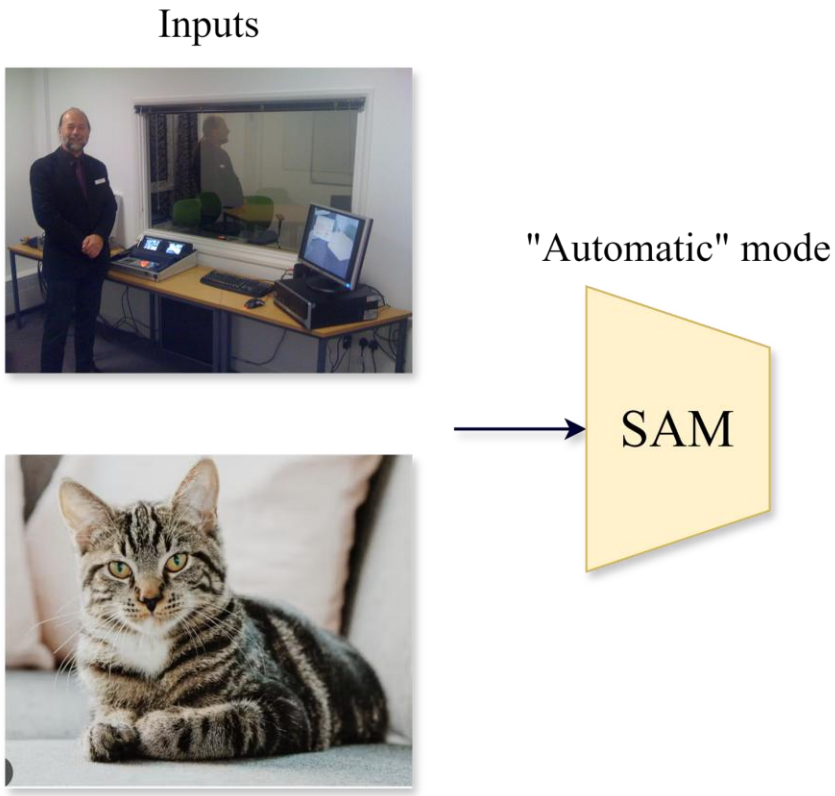
- SAM has been trained on 1B masks and segment objects accurately GIVEN a point or box prompt.
- For example, here on the same image we use two different types of box prompts to generate two different masks.
- However, collecting box annotations is still a laborious task. And implementing an object detector to collect the box prompts is not reliable because the detector will only work on the classes it has been trained on. So, if the detector has been trained on the “cat” class, the first case here will work, but it might not be able to give a box for the class “ears” like in the second case.

Segment Anything Model



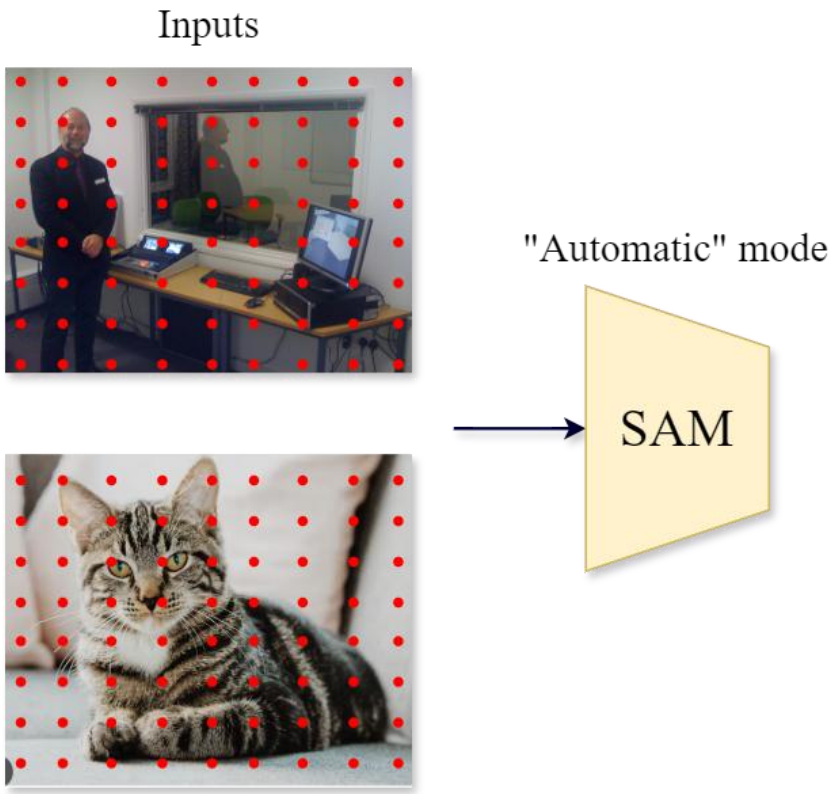
- SAM can also accept point prompts, however there are some problems with it.
- For example, in both cases here, my aim was to get the “person” segmented. However, since the point in the second case here is on the tie, the model has segmented the tie. In practice, SAM returns all the segments related to that prompt, so it will give “person”, “tie” and “suit” class segments. However, me, as the user, might only care about the person class. I don’t need the other segments.
- So, SAM is unable to provide targeted prompts based on my requirements.

Segment Anything Model



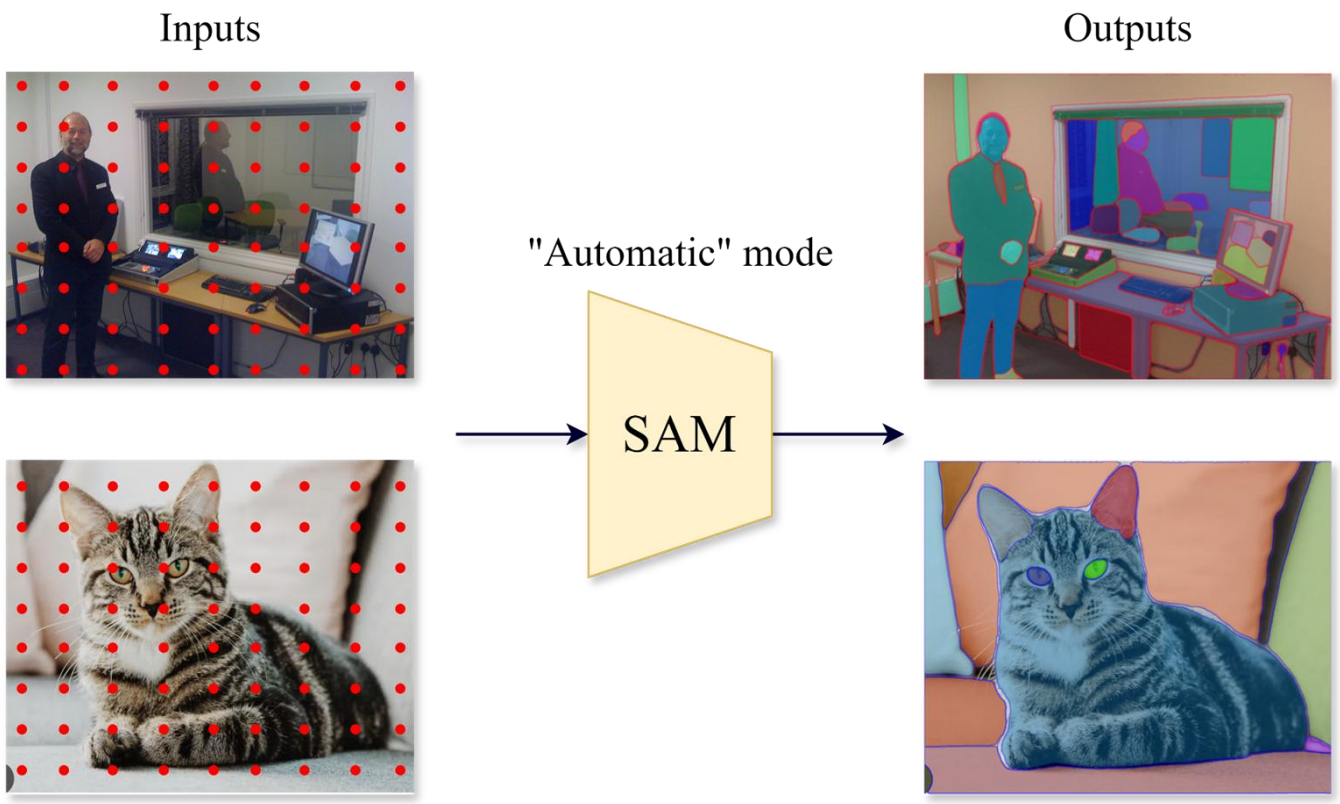
- SAM also has an “automatic” mode of operation- wherein, we don’t give it any point or box prompts, we only provide the image.

Segment Anything Model



- In this case, the SAM pipeline creates an equally spaced grid of points throughout the image, which serves as the set of point prompts for the model. Let's assume it creates a 100 points across the image.

Segment Anything Model

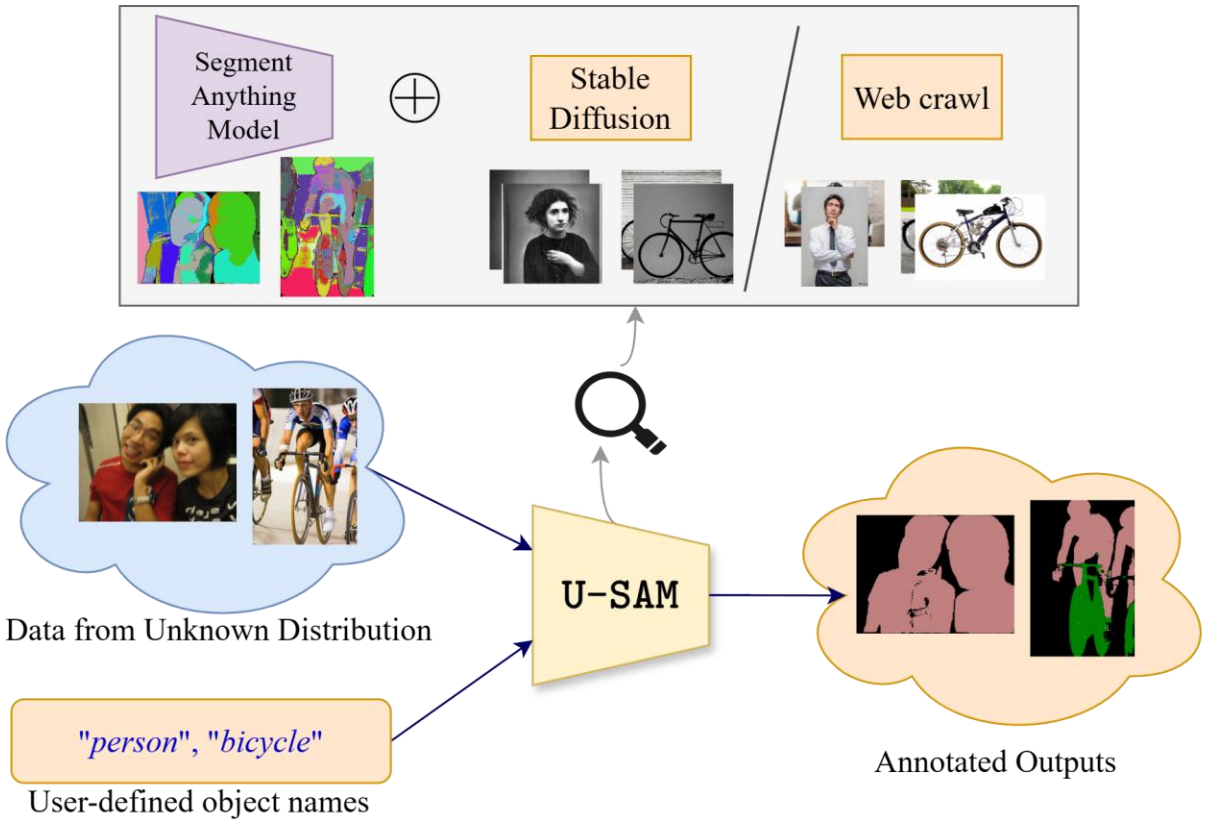


- Now, SAM generates a mask corresponding to each point, giving a hundred masks as output.
- SAM has no idea what object the masks corresponds to, since it lacks semantic awareness. Thus, in this case too targeted masks specific to the user's requirements are unavailable.

Overview

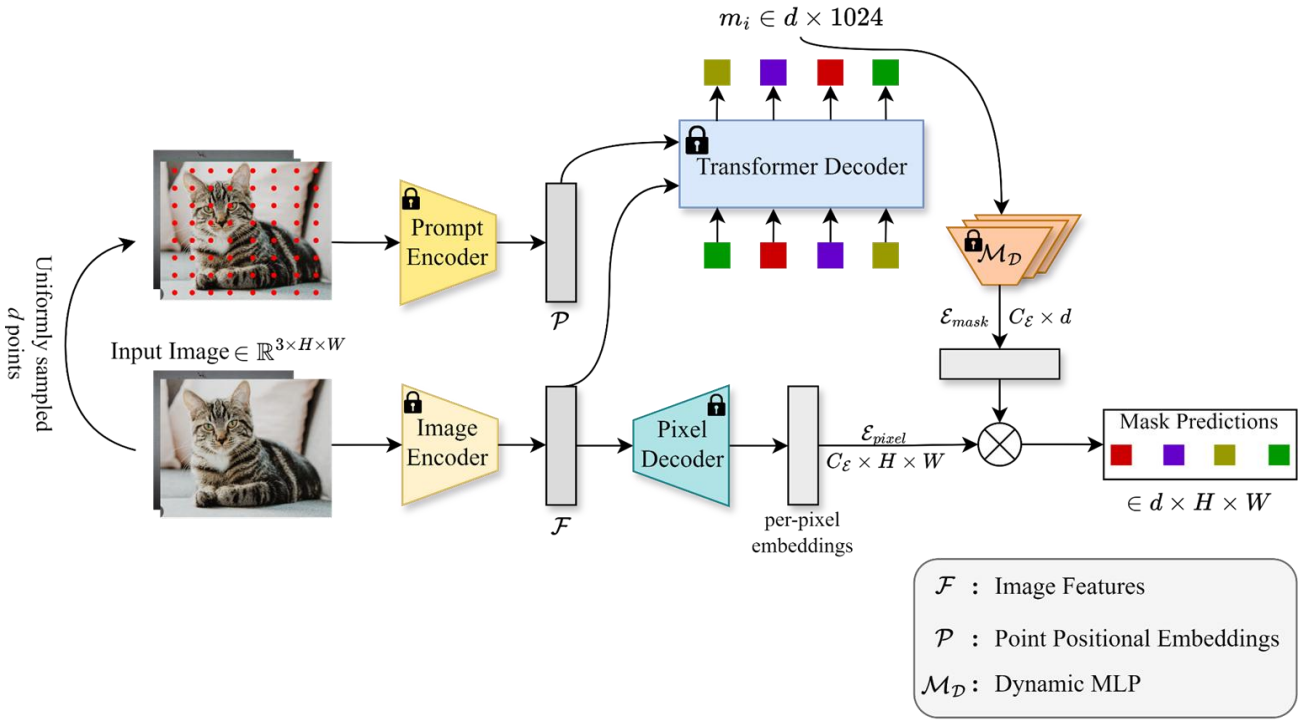
- Our goal is to segment any object the user wants without having any knowledge of the data distribution the user will provide at inference.
- To this end, we want to imbibe the semantic knowledge of objects into the existing SAM model.
- To achieve this, we implement a classifier head within the SAM architecture to provide targeted masks at inference.
- The classifier head is trained with Stable Diffusion or web-crawled images.

Overview



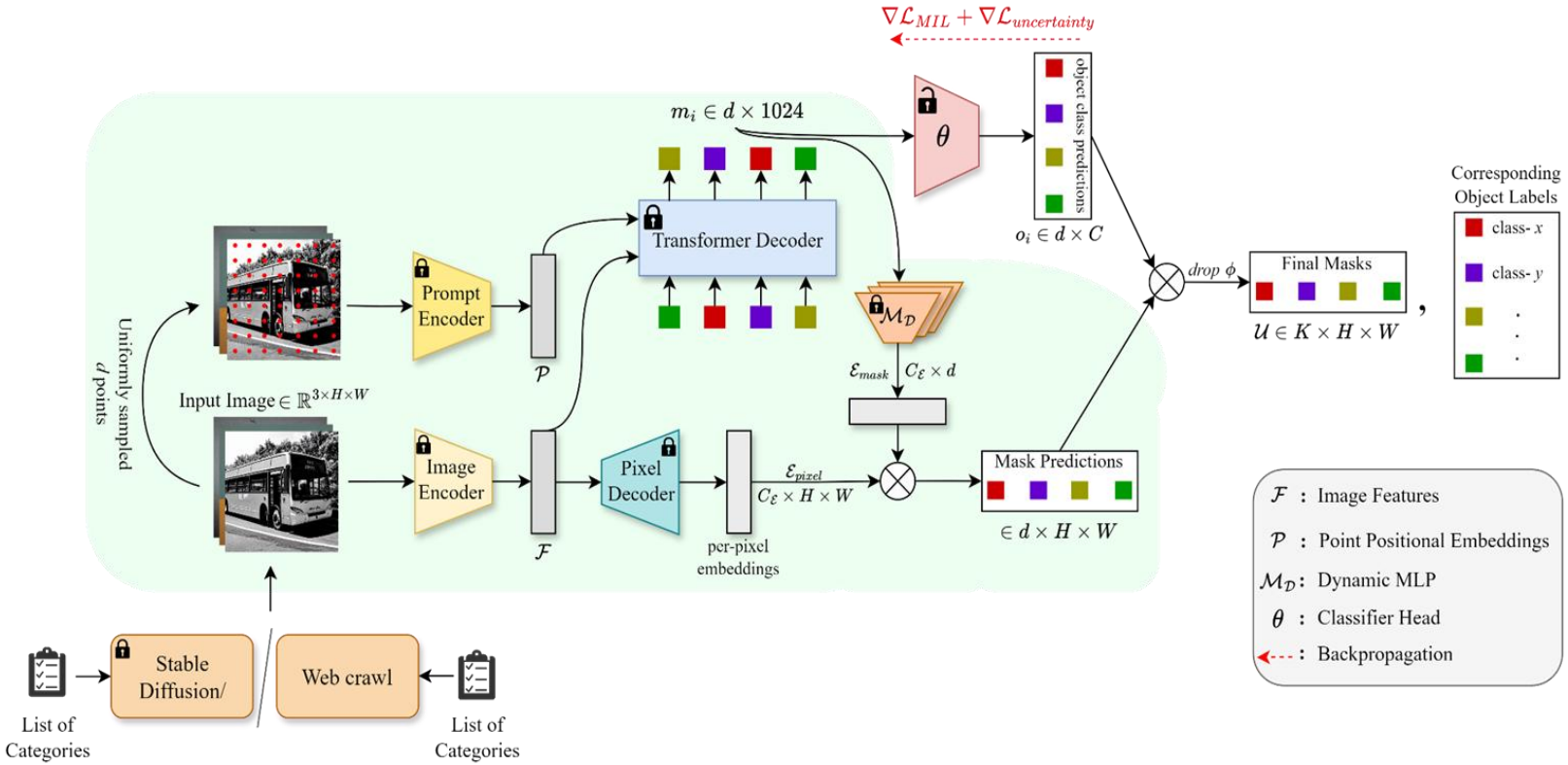
- Our model, named U-SAM, will be provided with the object class names by the user. We will generate example images from Stable Diffusion or crawl the web using these class names and train the GranSAM model.
- The trained U-SAM model will work on any test-data the user provides, given that the object class requirements are the same. It does not matter which dataset the images belong to.

SAM Architecture



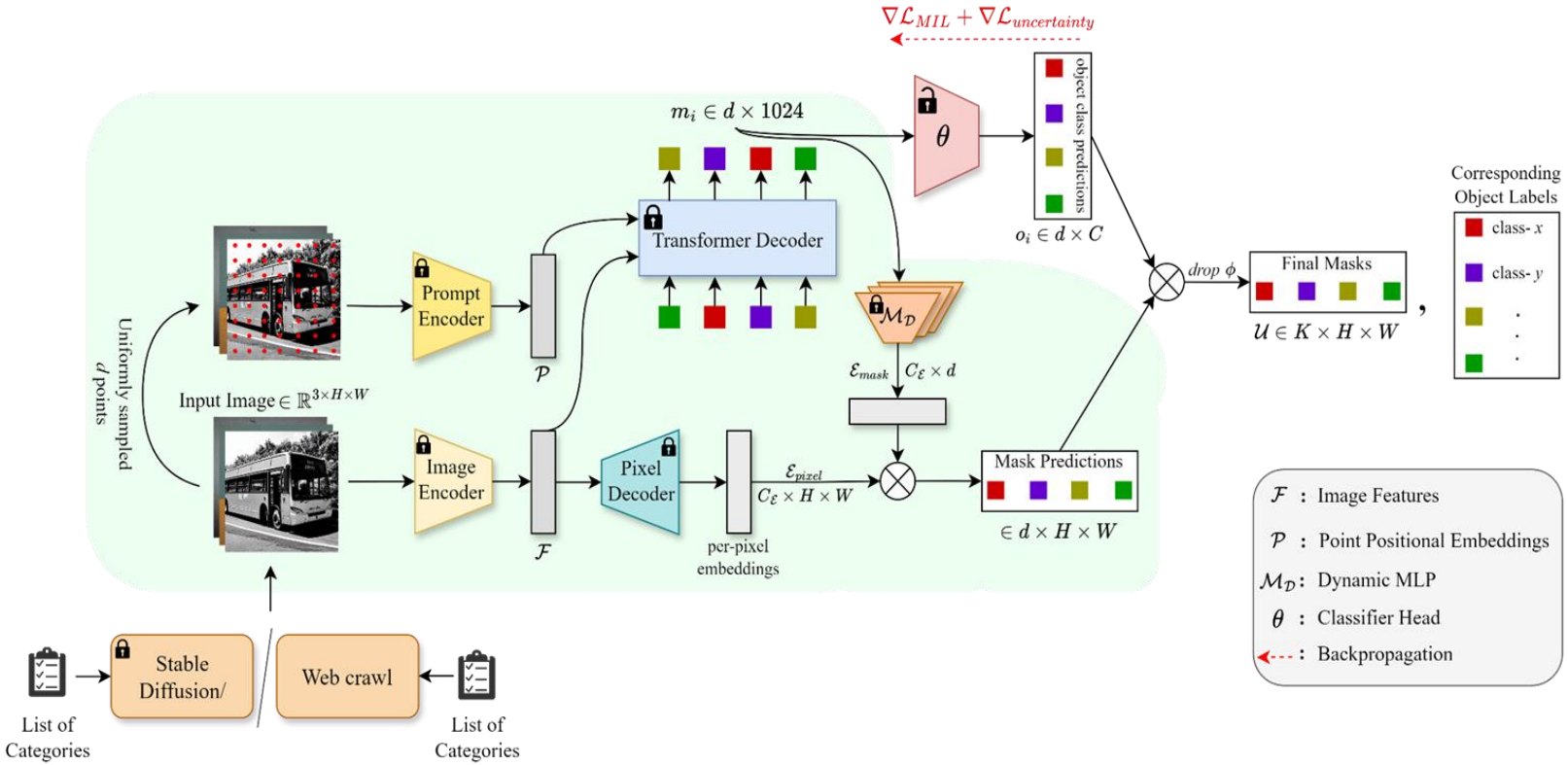
- The SAM model consists of the prompt and image encoders, and pixel and transformer decoders. As we discussed before, given an input image, SAM in its automatic mode generates "d" number of points across the image as prompts. The embeddings of the image from the image encoder, and the embeddings of these point prompts from the prompt encoder go into a transformer decoder which generates $d \times 1024$ mask embeddings.
- The mask embeddings then pass through a series of dynamic MLP to generate mask features denoted by $\mathcal{E}_{mask}$.
- $\mathcal{E}_{pixel}$ are the pixel features generated from the pixel decoder here.
- $\mathcal{E}_{mask}$ and $\mathcal{E}_{pixel}$ are jointly processed to generate the d- masks for the image.

Methodology: U-SAM



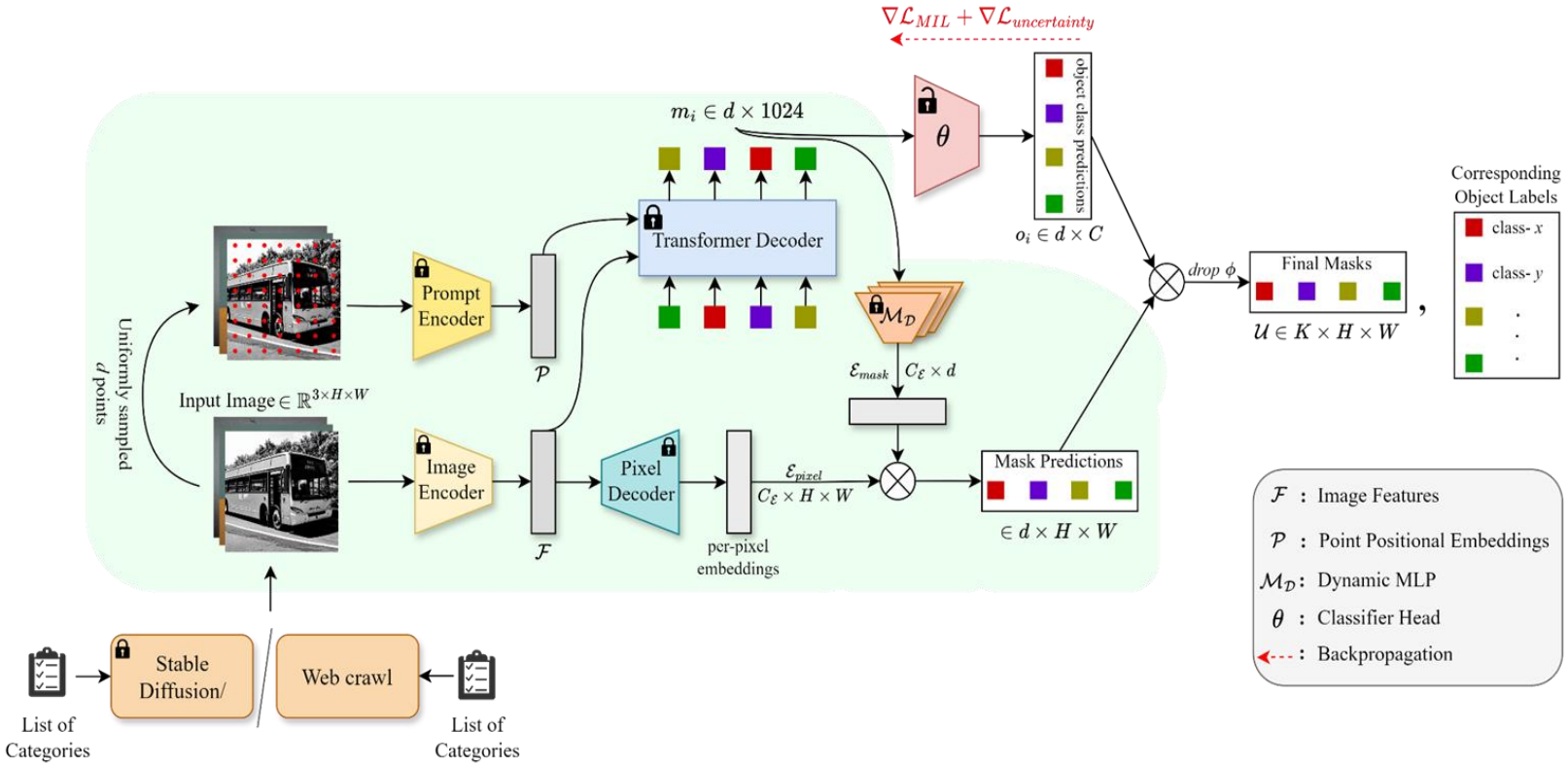
- The green highlighted portion is the original SAM from the previous slide.
- We take the mask embeddings, and design a classifier head, θ. θ's job is to predict the object class for each of the "d" mask embeddings.
- Using the predictions of θ, we select the user-required object mask and discard the others.

Methodology: U-SAM



- Since θ is being trained with single object images generated by Stable Diffusion or acquired from crawling the web, the supervisory signal for training θ is the object class label.
- And since the object class label is given by the user, the model also adjusts the granularity of predictions. This means that, for the same image, θ might look for the bus segmentation mask, or for the segmentation mask of the wheels of the bus, based on what the user has requested.

Methodology: U-SAM



- The training protocol of θ is not of a simple supervised learning. The input to theta is d -dimensional, corresponding to d-masks generated by SAM, but the output supervisory signal for it is only 1D, making this a many-to-one learning problem. That is, we know what object is present in the image during training, but we do not know which out of the "d" masks contains the object mask we require. So, we devised a multiple instance learning protocol to train θ.
- $\mathcal{L}_{uncertainty}$ is an uncertainty distillation training loss used for enhancing the results of the method by suppressing uncertain predictions.

Quantitative Results

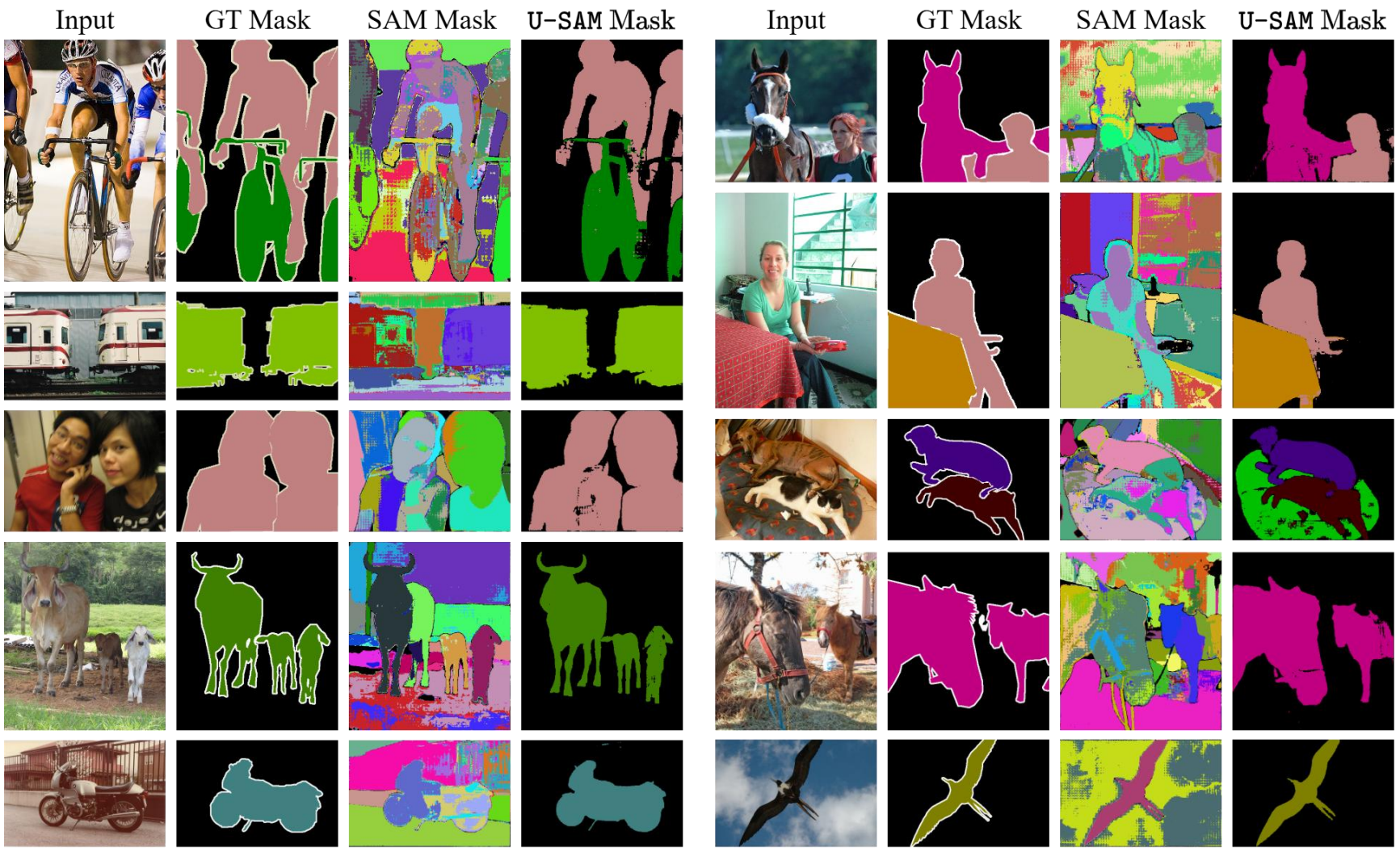
PASCAL VOC 2012

Method	Training Data		mIoU	mAP_{50}
	Synthetic	Web Crawl		
Leopart [59] *(US)*	✓		7.21%	-
TransFGU [51] *(US)*	✓		2.05%	-
ACR [23] *(WSS)*	✓		3.49%	-
U-SAM	✓		**25.16%** (+17.95%)	**49.06%**
Leopart [59] *(US)*		✓	6.79%	-
TransFGU [51] *(US)*		✓	2.45%	-
ACR [23] *(WSS)*		✓	13.42%	-
U-SAM		✓	**22.42%** (+9.00%)	**45.59%**

MSCOCO-80

Method	Training Data		mIoU	mAP_{50}
	Synthetic	Web Crawl		
Leopart [59] (US)	✓		3.84%	-
TransFGU [51] (US)	✓		0.95%	-
ACR [23] (WSS)	✓		0.61%	-
U-SAM	✓		**8.60%** (+4.76%)	**30.90%**
Leopart [59] (US)		✓	3.81%	-
TransFGU [51] (US)		✓	1.02%	-
ACR [23] (WSS)		✓	2.10%	-
U-SAM		✓	**9.01%** (+5.20%)	**31.80%**

Qualitative Results

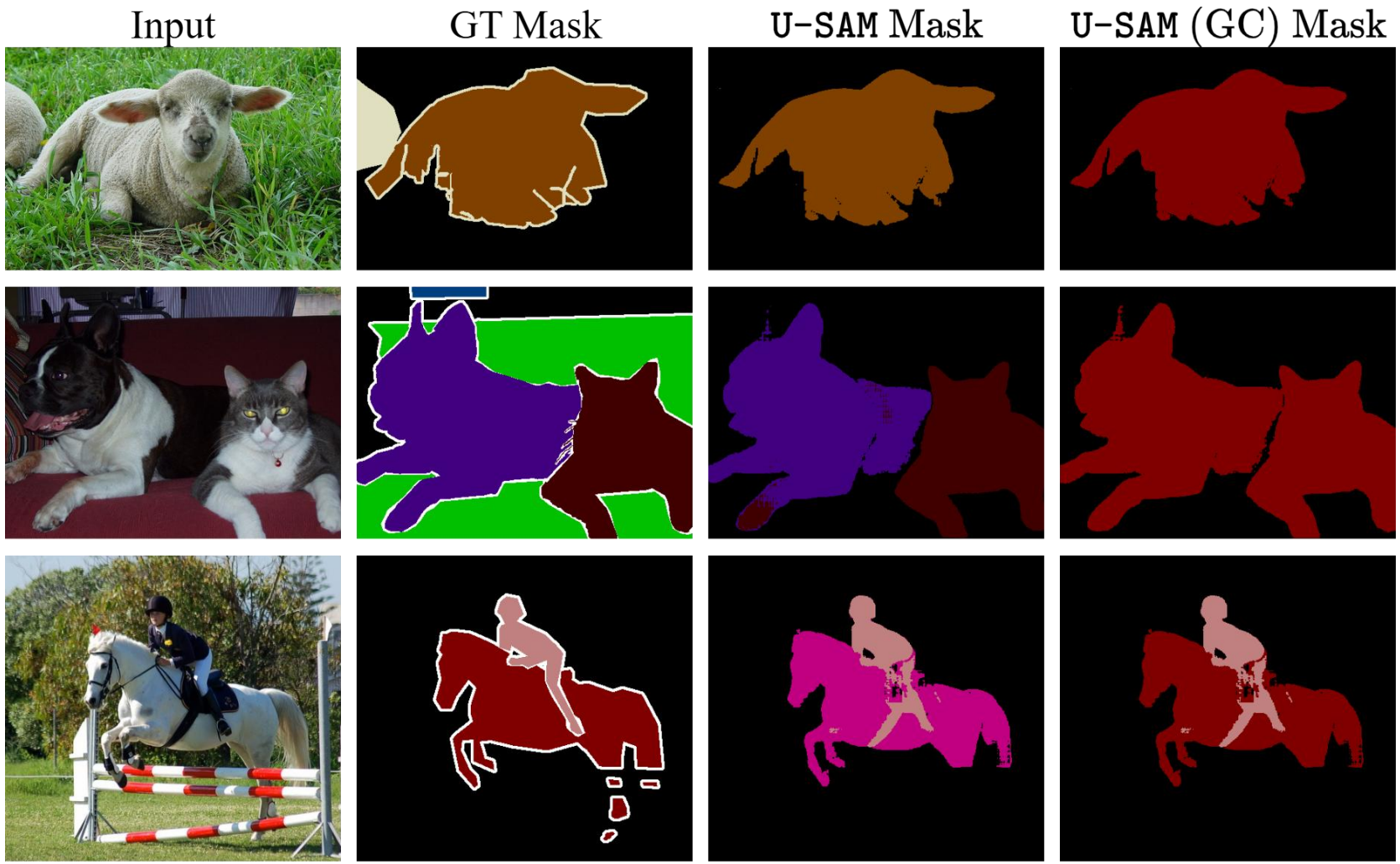


Changed Granularity

We changed the granularity of PASCAL VOC classes as follows:

1. ***animals***: “bird”, “cat”, “cow”, “dog”, “horse”, “sheep”
2. ***furniture***: “chair”, “dining table”, “sofa”
3. ***household items***: “bottle”, “potted plant”, “tv monitor”
4. ***person***: “person”
5. ***transportation***: “aeroplane”, “bicycle”, “boat”, “bus”, “car”, “motorbike”, “train”

Qualitative Results with Changed Granularity



GC: Granularity Changed

THANK YOU

UCR Vision and Learning Group