Table S1: Supervised Learning parameters.

Config	CamVid	CityScapes
Learing Rate	$1e-2$	$1e-2$
Optimizer	SGD	SGD
Scheduler	PolyLR	PolyLR
Batch Size	4	4
Epoch Iterations	100	200
Epochs	100	100
Augmentations	Resize([0.75,1.25]), ColorJitter(p = 0.5), HFlip(p = 0.5)	Resize([0.5,2.0]), ColorJitter(p = 0.5), HFlip(p = 0.5)

Table S2: Active Learning with Semi-Supervised Learning parameters.

Config	CamVid	CityScapes
Learing Rate	1e-2	1e-2
Optimizer	SGD	SGD
Scheduler	PolyLR	PolyLR
Batch Size (Labeled)	4	4
Batch Size (Un-Labeled)	4	4
Epoch Iterations	50	100
Epochs	100	100
Coldstart Epochs	10	10
Final Cycle Epochs	200	200
Active Learning Cycles	2	5
Active Learning Regions	$30 \times 30 \times 4$	$43 \times 43 \times 4$
Augmentations (Labeled)	Resize([0.75,1.25]), ColorJitter(p = 0.5), HFlip(p = 0.5)	Resize([0.5,2.0]), ColorJitter(p = 0.5), HFlip(p = 0.5)
Weak Augmentations (Un-Labeled)	Resize([0.75,1.25]), HFlip(p = 0.5)	Resize([0.5,2.0]), HFlip(p = 0.5)
Strong Augmentations (Un-Labeled)	Resize([0.75,1.25]), ColorJitter(p = 0.5), HFlip(p = 0.5), ClassMix[59]	Resize([0.5,2.0]), ColorJitter(p = 0.5), HFlip(p = 0.5), ClassMix[59]
Replay Buffer Size	50	500

Supplemental Material

We will release all code towards reproducing the results in this paper post publication. We discuss the experiments and related results in the following sections.

S1. Experimental Configurations

Tables S1, S2 show the default hyper-parameters for training supervised and active learning based networks. We use the standard set of augmentations followed for supervised and semi-supervised learning, and append the latter with ClassMix [59]. For semi-supervised learning per active learning cycle, we train with only labeled data for the initial 10 epochs to provide a good initialization for the teacher. In addition, we only start adaptive ClassMix once the first active learning cycle has completed - this reduces the initial training time and also makes the intended usage with additional labeled data. We train our networks on two Nvidia Titan Xps (CamVid) and two Nvidia V100s (CityScapes).

S2. Comparison to Region-based approaches

We consider two algorithms developed for region-based semantic segmentation - RALIS [11] and EquAL [29]. RALIS uses a ResNet-50 backbone with Feature Pyramid Networks [48], enhanced with pretraining on the GTA-V dataset [65]. It also uses initial labeled sets of 30% for CamVid and 12% for CityScapes, which are higher than our initial budget of 10% on both datasets. In comparison, our approach achieved 97% of its fully-supervised performance on CamVid using while only actively sampling an additional 3.8% of total data (13.8% overall), whereas RALIS required an additional 24% of the total data (54% overall) to reach a maximum performance of 96%. In addition, our approach achieved nearly identical results to the fully-supervised performance on the CityScapes dataset using only an additional 8% of the total data (18% overall), whereas RALIS achieved 96% of fully-supervised performance using an additional 9% of total data (21% overall). It is worth mentioning that in both datasets of interest, RALIS used a pretraining boost with the GTA V dataset on ResNet-50, whereas we only use ImageNet-pretrained weights on MobileNetv2.

For EquAL, direct comparison is not possible due to the unknown labeled-unlabeled split, so instead we compare performance using the same labeled fraction of the data with ResNet-50 backbone. Under the same training paradigm (starting with 8% labeled data, a budget of 12% labeled data, and using a ResNet-50 backbone with DeepLabv3+), our approach achieved an mIOU of 65.3 $\pm$ 0.2 on CamVid, as compared to 63.4 from [29] [3]. For CityScapes, we found it realistically impossible to begin with only 1% labeled data due to sampling concerns and no recorded data splits, so we begin with 3.5% data instead which is a conventional choice for semi-supervised learning tasks [2, 46, 16, 50].

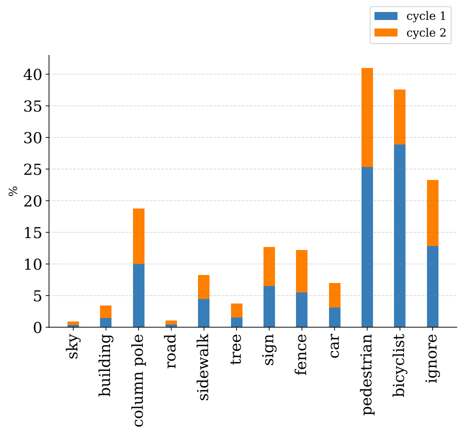
[3]as mentioned in EquAL's GitHub repository

Compared to EquAL's 12% usage with an mIoU of 67.4, we obtained 66.7 $\pm$ 1.5 using only 10% of the total labeled data. We believe the higher variance here as opposed to our other results is caused by using only 3.5% data initially labeled data (vs. 10% in the other experiments), and further research could help reduce this variance with improvements in SSL [2, 46, 16, 50].

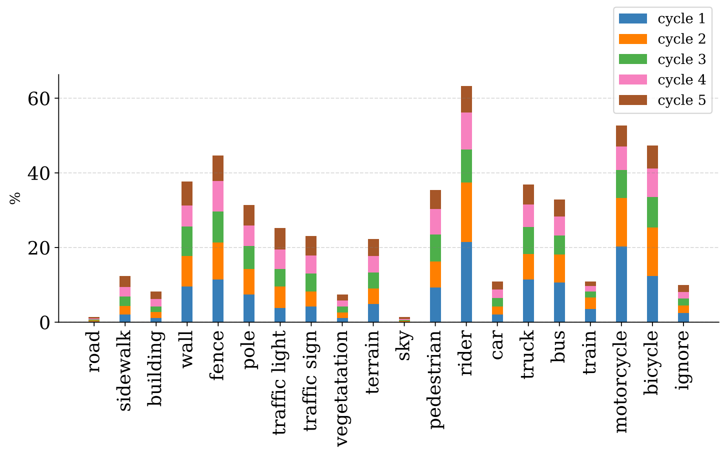
S3. Visual Results

Fig. S1a shows us the important classes belonging to the regions queried for labeling on the CamVid dataset. We can observe high priority is given to the tail distribution classes (in terms of pixel count), namely column pole, pedestrian and bicycle. Regions consisting of areas falling under the ignore category are also sampled, which also proves that the network does not easily assign a particular class from the known categories to a new-ly observed object. Concurrently, Fig. S2 shows us some examples of qualitative results on the CamVid dataset - the first three rows show the network trained with S4AL predicting near similar or better than the fully-supervised network, and the last row shows the most common failure case with respect to the 'Fence' class, which tends to get confused with 'Building' class due to their structural similarities.

Similarly, Fig. S3 shows us some examples of on the CityScapes dataset - the first three rows show the network trained with S4AL predicting near similar or better than the fully-supervised network, and the last row shows the most common failure case with respect to the 'Train' class, which tends to get confused with 'Bus' class due to their structural similarities. We observed this in Table 2b as well, thus possibly indicating that a hybrid acquisition model of image and regions, on a per image basis, would be most beneficial for complex scenes. Fig. S1b shows us that rider, motorcycle and bicycle, three very similar visual categories, were the highest queried regions in the CityScapes dataset. The network also queries a relatively low number of pixels belonging to the 'ignore index' label, effectively showing a better sense of understanding for the dataset as most regions for ignoring belong to the hood of the car that is gathering all the data.



(a) CamVid



(b) CityScapes

Figure S1: What does the network want? We visualize the region-wise samples per active learning cycle on both datasets for our main split

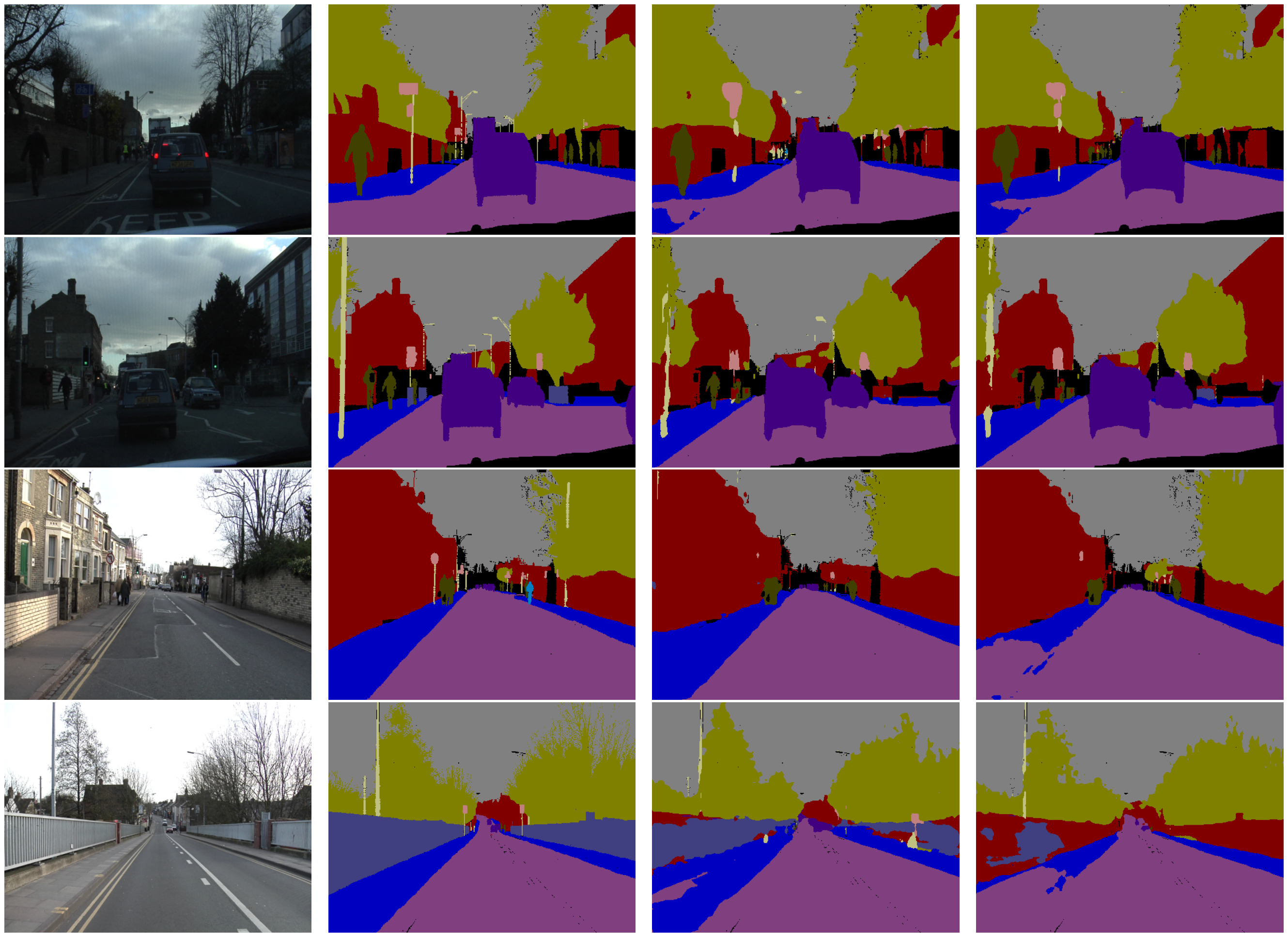


Figure S2: Examples of semantic segmentation outputs on CamVid, the columns represent the image (left), the ground truth (center-left), the predictions of a supervised network (center-right), and the predictions of a network trained via S4AL (right).

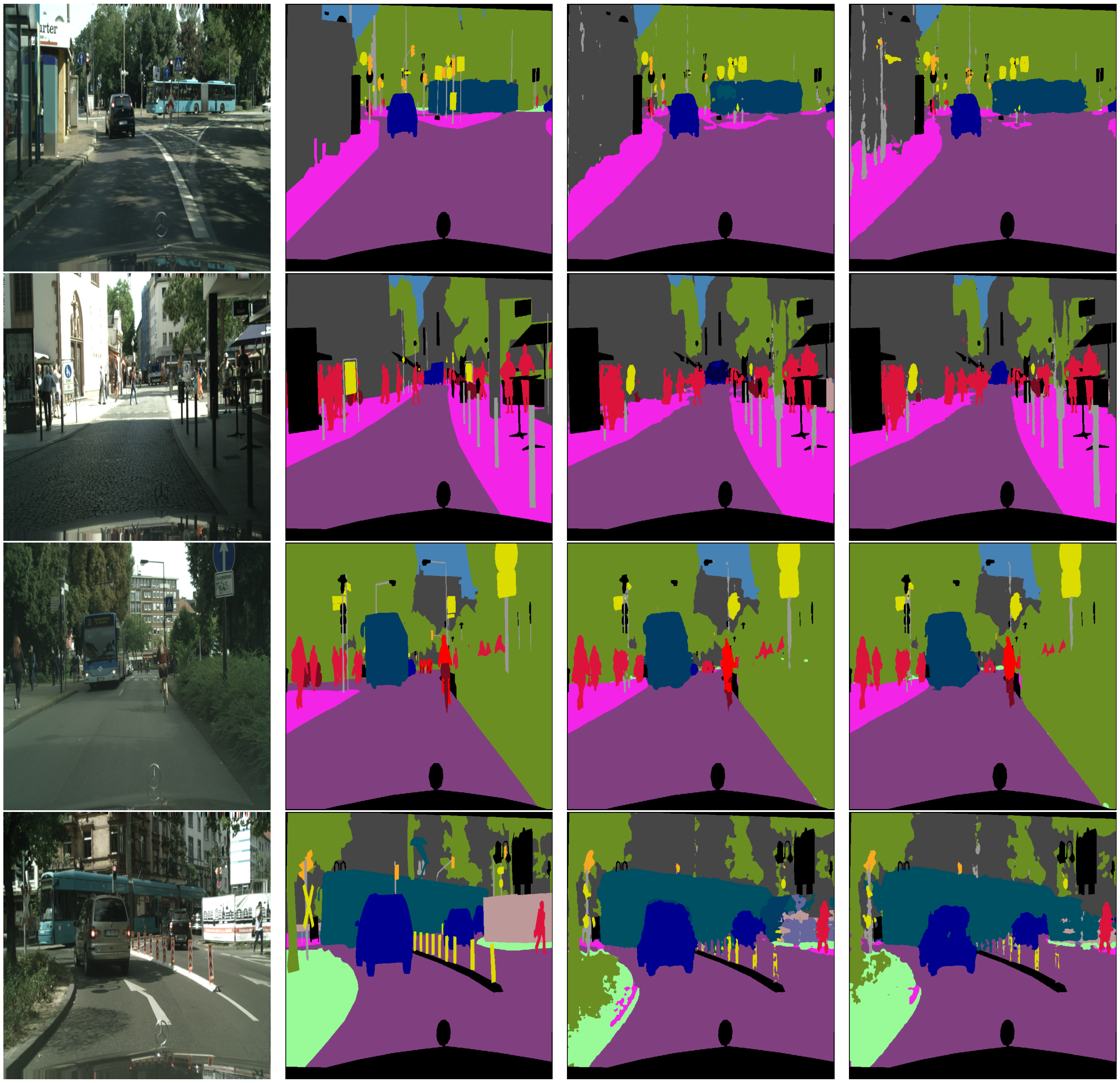


Figure S3: Examples of semantic segmentation outputs on CityScapes, the columns represent the image (left), the ground truth (center-left), the predictions of a supervised network (center-right), and the predictions of a network trained via S4AL (right).