A POC datasets object list

Our POC evaluation sets (*i.e.*, POC-CS, POC-IDD and POC-ACDC) are obtained by adding objects to different self-driving datasets. Following works like Lost and Found [47], we compiled a list of 25 objects that can be found on the road.

The **anomaly list** is as follows: "stroller", "trolley", "garbage bag", "wheelie bin", "suitcase", "skateboard", "chair dumped on the street", "sofa dumped on the street", "furniture dumped on the street", "matress dumped on the street", "garbage dumped on the street", "clothes dumped on the street", "cement mixer on the street", "cat", "dog", "bird flying", "horse", "skunk", "sheep", "crocodile", "alligator", "bear", "llama", "tiger" and "monkey".

Additionally, we also add a few classes from Cityscapes to make sure that anomaly segmentation models are indeed detecting *anomalies* and not merely identifying *synthetic objects*.

Cityscapes classes inluded are: "rider", "bicycle", "motorcycle", "bus", "person" and "car".

B Anomaly segmentation methods: mIoU on Cityscapes

Although fine-tuning methods with OOD samples can improve anomaly segmentation significantly, they may affect the closed-set performance. In Tab. 4 we report the mIoU on Cityscapes of all methods reported in the main paper, showing that fine-tuning with POC data does not negatively impact closed-set performance.

Method	**Mask2Anomaly**				**RPL**				**RbA**			
OOD data	No ft.	coco	POC c	POC alt.	No ft.	coco	POC c	POC alt.	No ft.	coco	POC c	POC alt.
mIoU ↑	78.29	78.34	78.33	78.49	90.94	90.94	90.94	90.94	82.25	82.15	82.17	82.16

Table 4: mIoU on Cityscapes validation set. We compute the mIoU after fine-tuning with different datasets (complementing results in Tab. 2). We observe that fine-tuning with our POC datasets does not degrade the closed-set performance.

C Adding objects with Instruct Pix2Pix

When building our pipeline, we explored different generative methods. In particular, InstructPix2Pix (IP2P) [4] has showed remarkable performance following natural language instructions (*e.g.*,"turn the sofa red"). Therefore, it would be natural to have such "general-purpose editing" methods as baselines to add objects to the images. We observed that IP2P seems to be biased towards *modifying* objects in the scene rather than *adding new ones*. In Fig. 8 we show several examples of images generated with IP2P. In our initial experiments, we observe how new objects tend to replace the logo of the ego vehicle (top images). If we remove the bottom of the image (middle section), we then observe that some

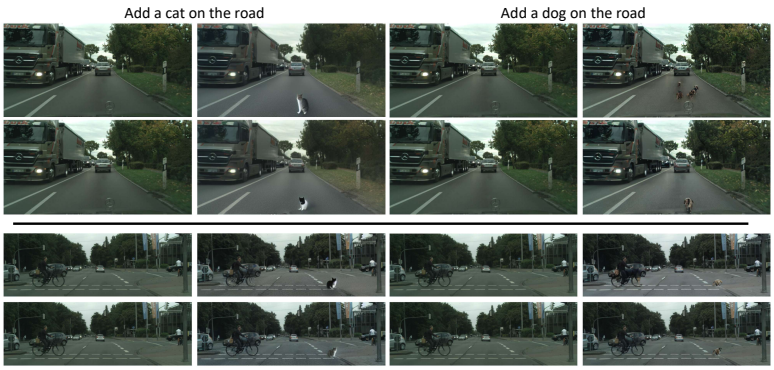

Fig. 8: Sample images from InstructPix2Pix [4]. We observe that InstructPix2Pix has a bias to replace certain objects or features in the scene. In top images it replaces the mercedes logo of the ego vehicle while in bottom images it replaces the edge of the sidewalk.

particular image features (*e.g.*, the edge of the sidewalk) tend to be replaced. Moreover, the added objects tend to lack realism and, given that the changes are not constrained to a particular region, editing via IP2P usually results in undesired modifications in other image regions.

D Ablation of guided region selection

We argued in the main text that in order to add objects into scenes realistically, it is important to properly place them. Thus, we apply GSAM to segment a valid area based on a location prompt (*e.g.*,"the road") and then select a region randomly within the valid area. Without this component in our pipeline, the objects result inpainted in clearly unrealistic positions.

To assess the realism introduced by guiding the object location *vs.* placing objects randomly, we conducted a human study where participants were shown different pairs of images, one with guided location inpainting and the other with random placement, and asked to choose the most realistic image in each pair. We observed that 39% of times the preference was unclear, 43% guided location was preferred and 18% random location was preferred. Note that a large portion of Cityscapes images is road/street, thus, a significant portion of randomly placed objects will be realistic. On the other hand, when the location is different, the generated objects also vary (even if fixing the random seed), which adds some noise to the study. All in all, we do observe a clear preference for guided location compared to random placement. In Fig. 9 we show some examples of image pairs with guided and random locations. Note how the added "dumped clothes" in the bottom right are both in realistic locations while the other objects are placed unrealistically with the random location.

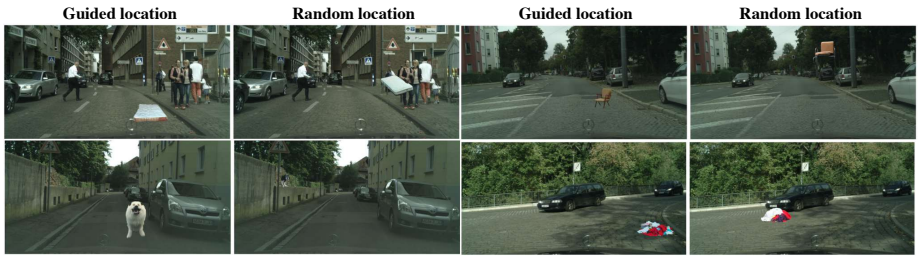

Fig. 9: Location ablation examples: Different examples of images inpainted with our guided location or random location of objects. We observe how in many cases, random location leads to unrealistic scenes.

E Ablation of image2image blending

Similar to our location ablation, we also study if applying an image2image (I2I) model after object inpainting leads to better blending. In particular, we performed a human study where participants had to choose the most realistic image between our blending ans I2I. In 71% of the cases I2I blending did not improve results significantly, in 25% it introduced artifacts that significantly reduced realism and in only 4% participants preferred I2I blending.

In particular, we noted that I2I blending adds slight artefacts that can degrade the realism of the image significantly, these become especially noticeable in text or traffic signs where small variations can change the semantics drastically. In Fig. 10 we present two examples of such images where artefacts are highlighted.

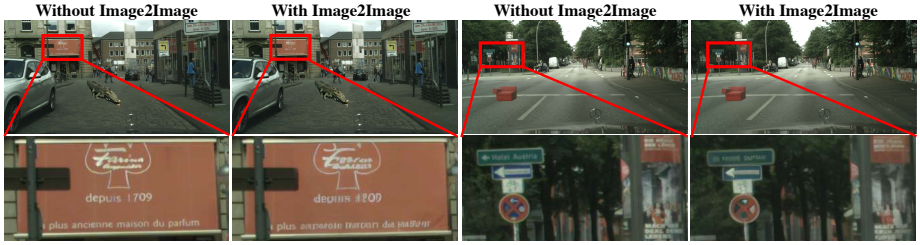

Fig. 10: Examples of object blending: On the left image pair, we observe the presence of articles in the text of an old perfume shop which is legible on the right image but becomes illegible with I2I. On the right image pair, one can observe differences on the traffic signs. For instance, the text "Hotel Austria" on the green sign on the top (legible when zoomed on the left) becomes again illegible on the right image. Also, the white squared sign with a depiction of a bike without I2I becomes uninterpretable after I2I.

F AUPRC plots

In Fig. 11 we visualize the AUPRC results for all methods, complementing the visualization in Fig. 2.

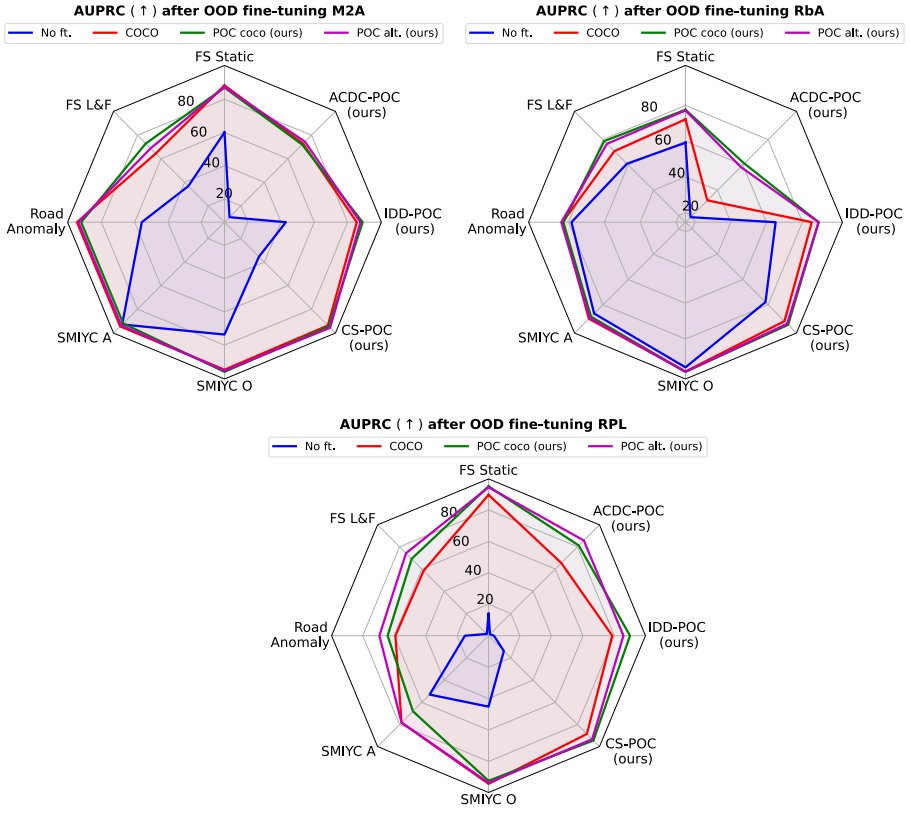

Fig. 11: AUPRC on different anomaly segmentation datasets. We compare three different anomaly segmentation methods, M2A [48], RPL [39] and RbA [44] with different fine-tuning datasets. Fine-tuning with POC-generated images tends to bring improvements or match COCO fine-tuning in most settings.

G Additional Pascal training samples

In Fig. 12 we show additional samples generated with our POC pipeline as well as T2I baselines (that use text-to-image models inspired by [28]).

Fig. 12: Training image samples. Additional training images to learn new classes, complementing Fig. 6.

H Additional anomaly score maps

In this appendix section we add more visualizations of anomaly score maps with different methods. We show results from our three POC-generated datasets as well as samples from previous anomaly datasets.

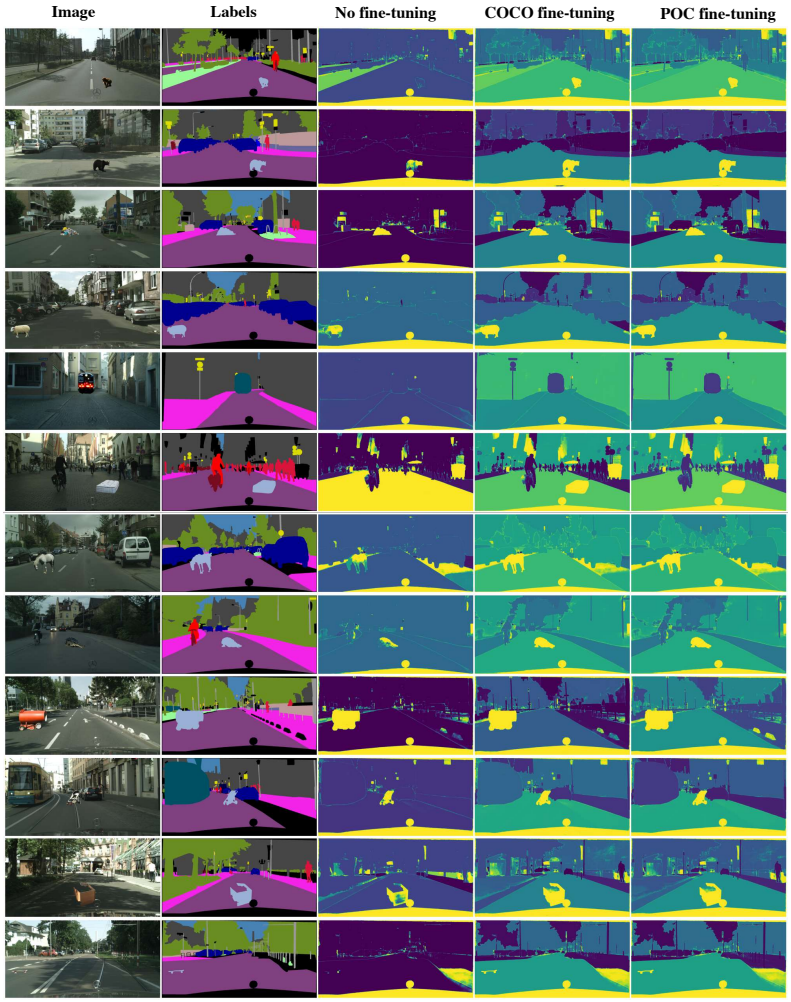

Fig. 13: M2A anomaly scores on CS-POC samples.

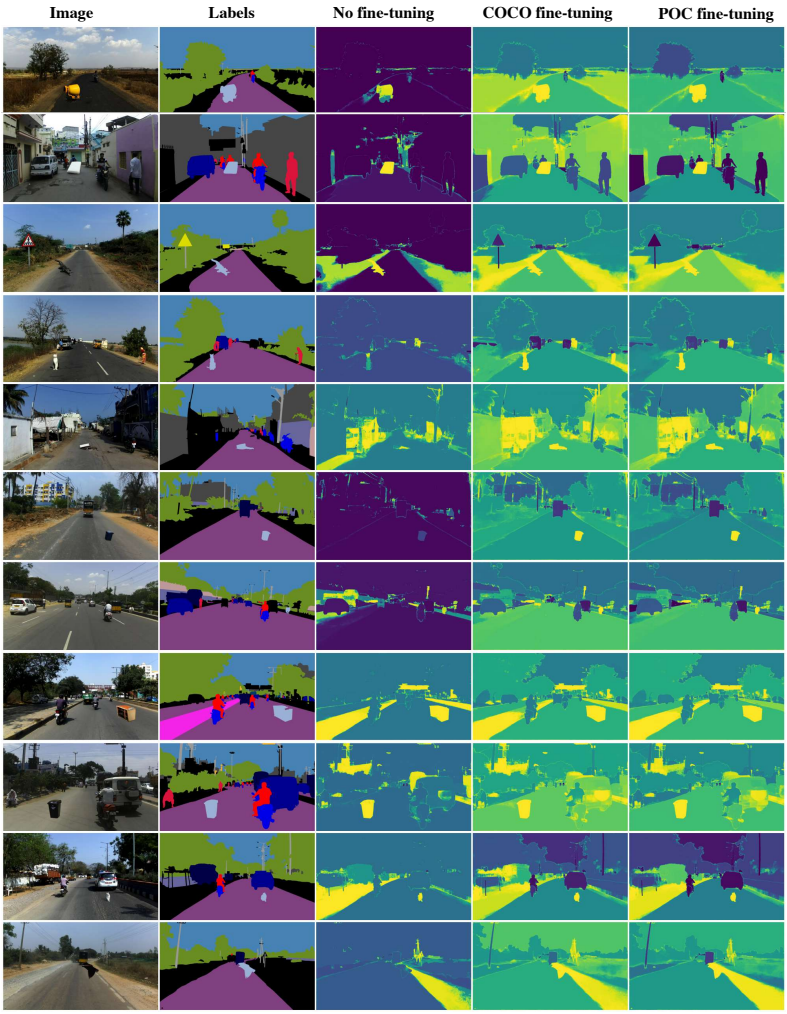

Fig. 14: M2A anomaly scores on IDD-POC samples.

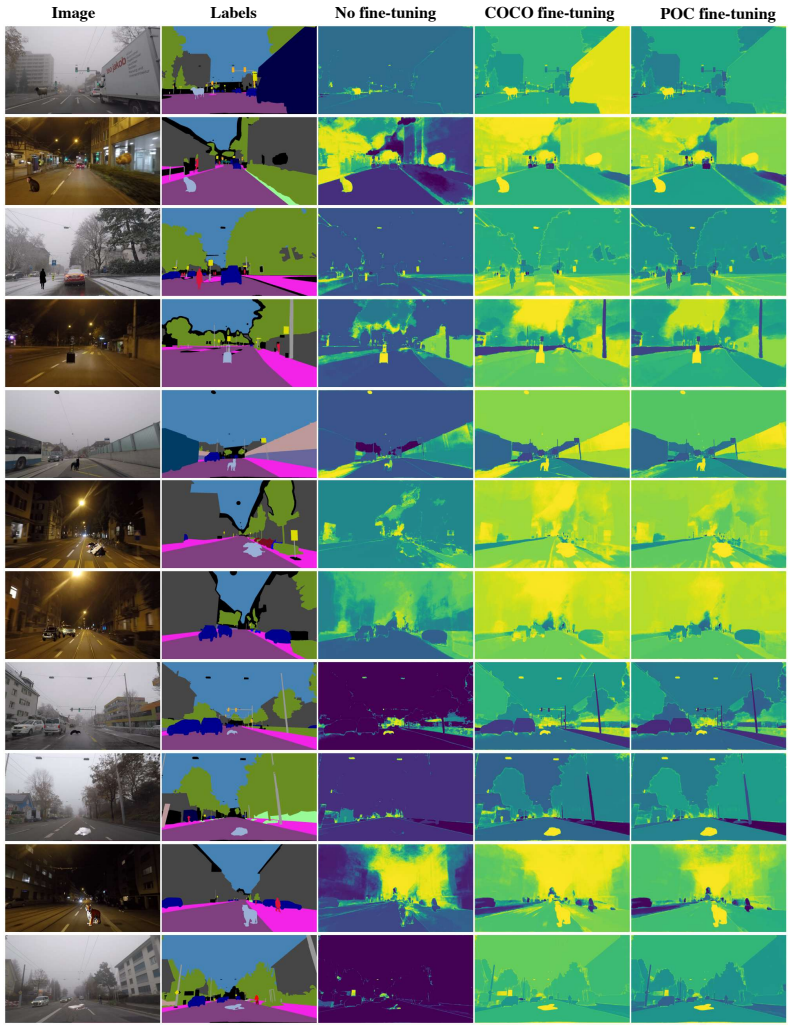

Fig. 15: M2A anomaly scores on ACDC-POC samples.

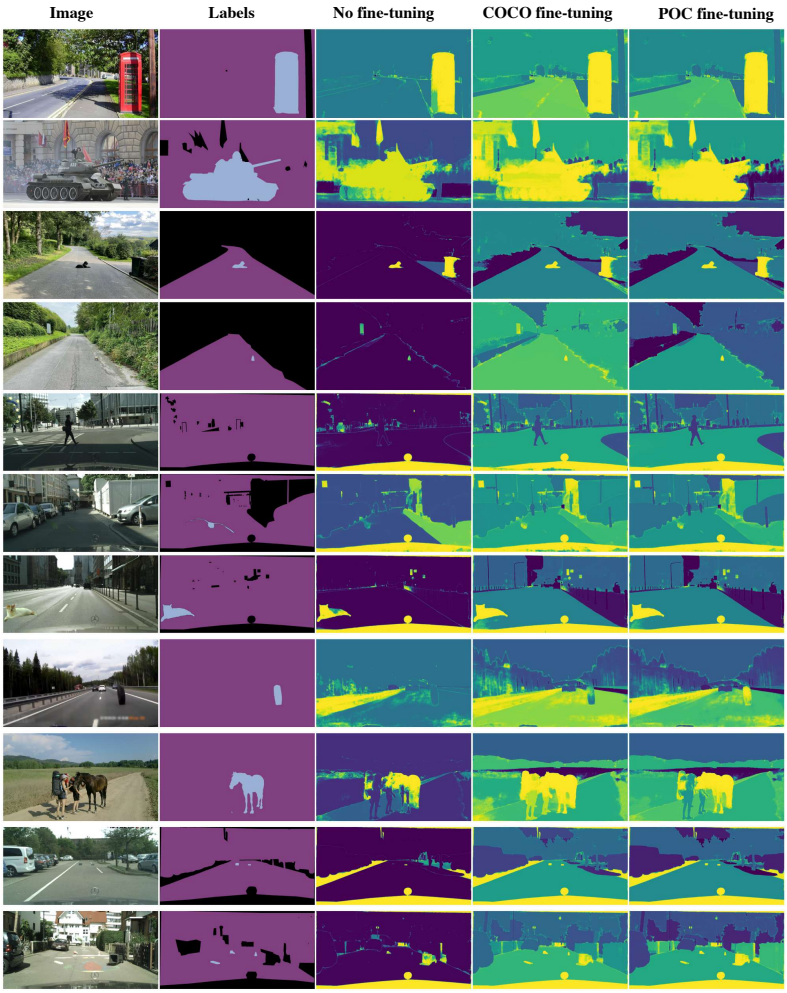

Fig. 16: M2A anomaly scores on samples from related datasets (see Fig. 1).

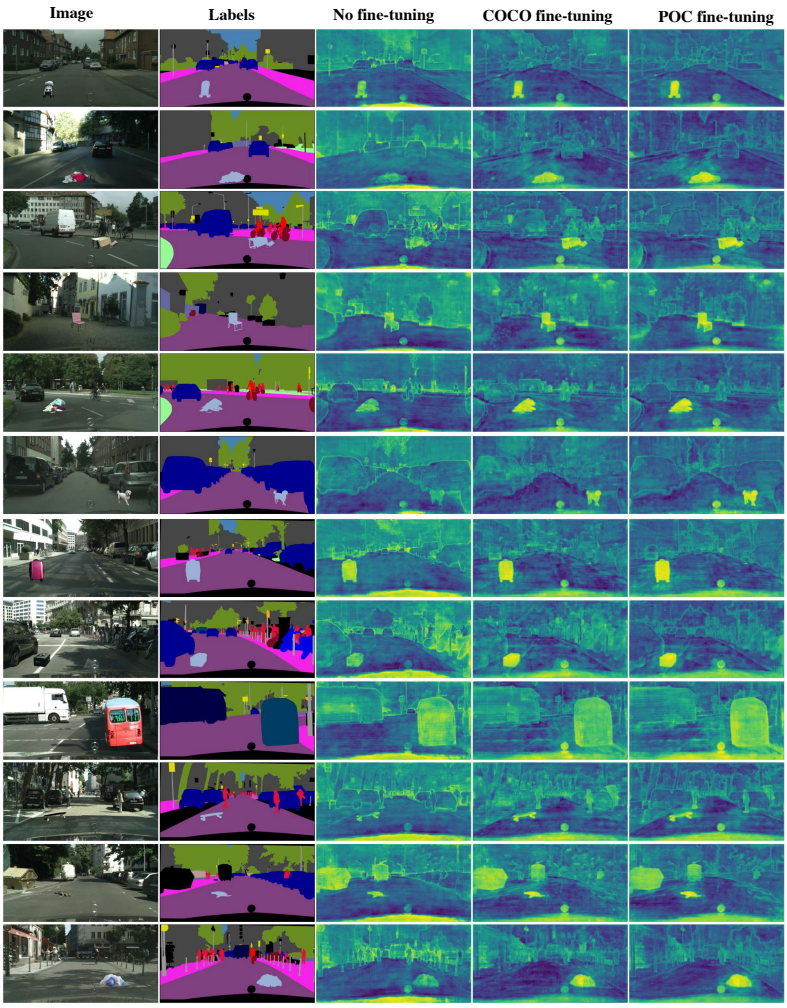

Fig. 17: RPL anomaly scores on CS-POC samples.

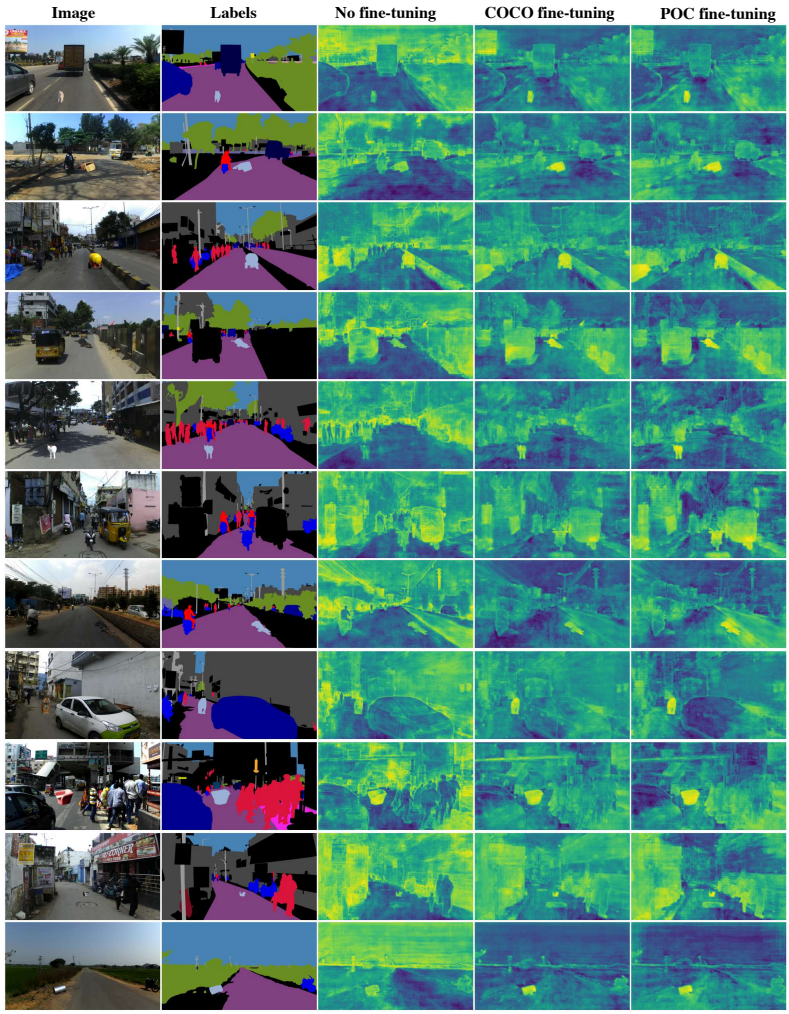

Fig. 18: RPL anomaly scores on IDD-POC samples.

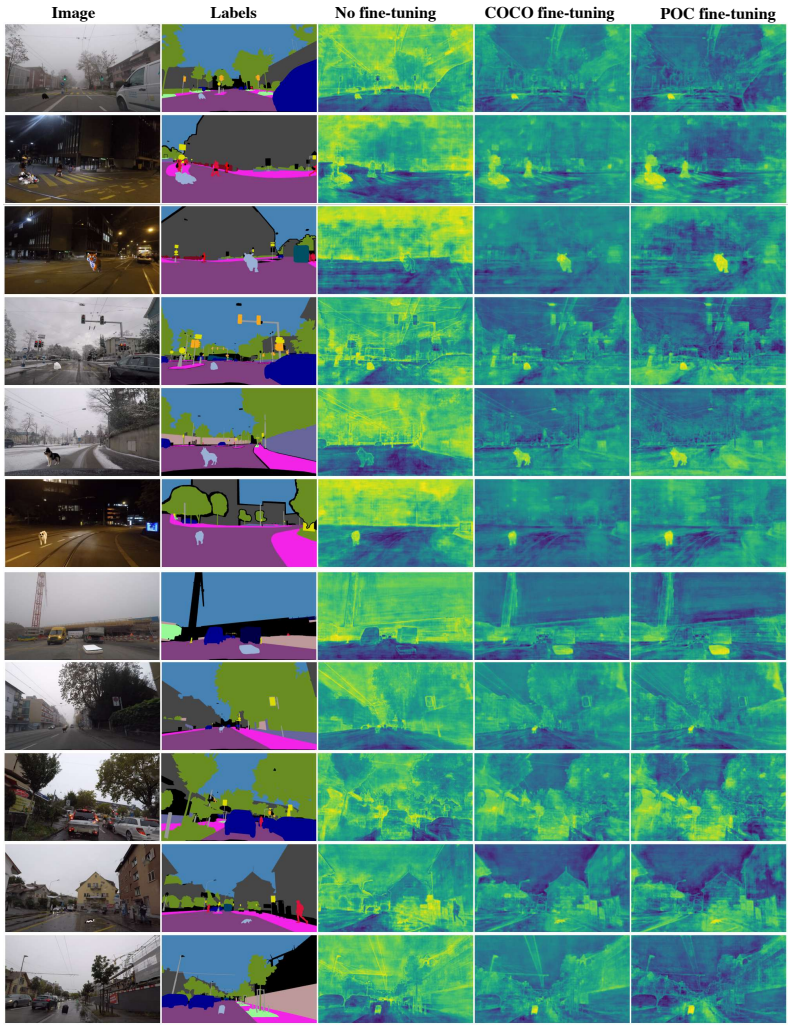

Fig. 19: RPL anomaly scores on ACDC-POC samples.

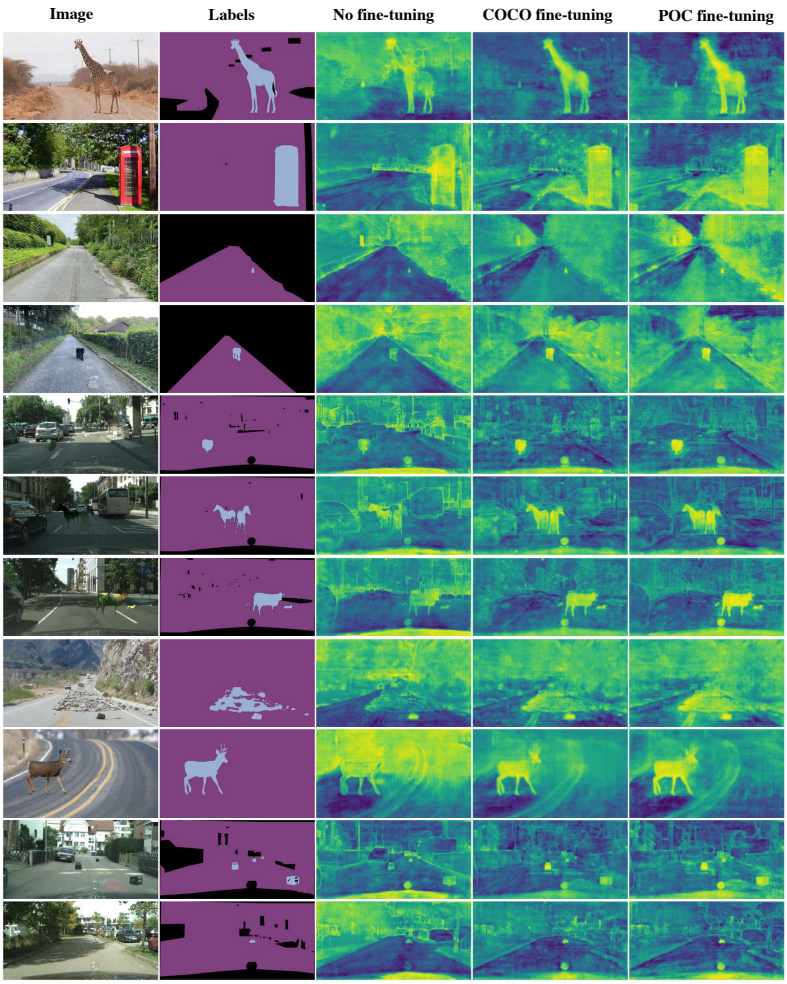

Fig. 20: RPL anomaly scores on samples from related datasets (see Fig. 1).

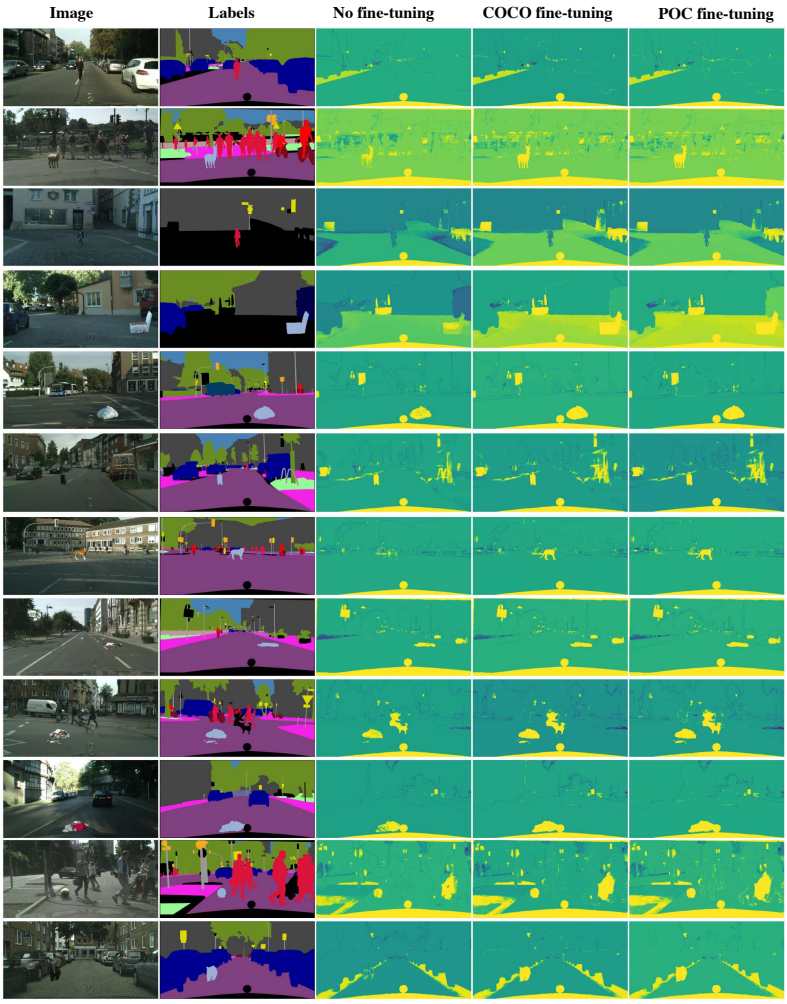

Fig. 21: RbA anomaly scores on CS-POC samples.

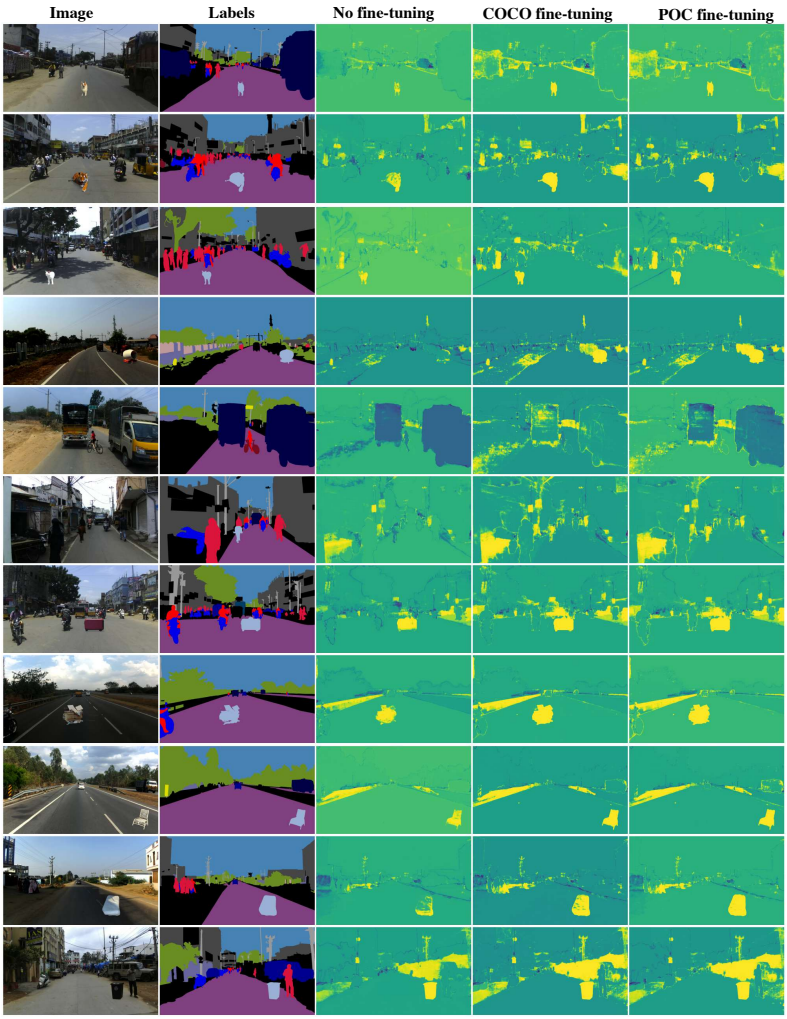

Fig. 22: RbA anomaly scores on IDD-POC samples.

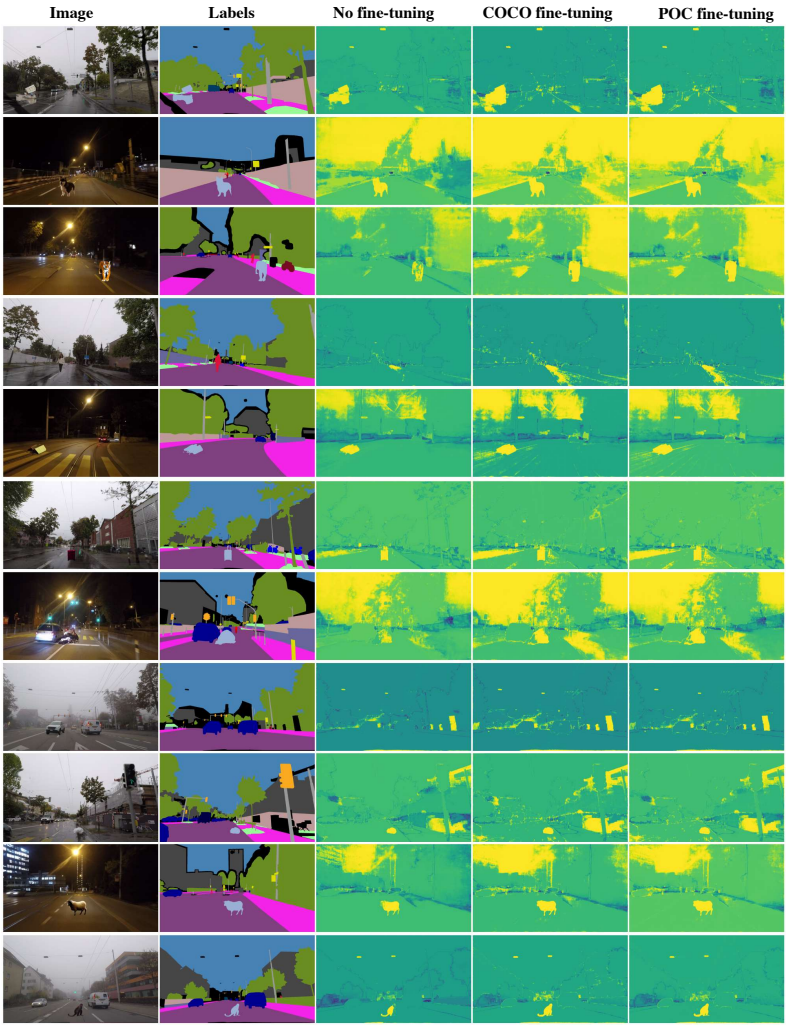

Fig. 23: RbA anomaly scores on ACDC-POC samples.

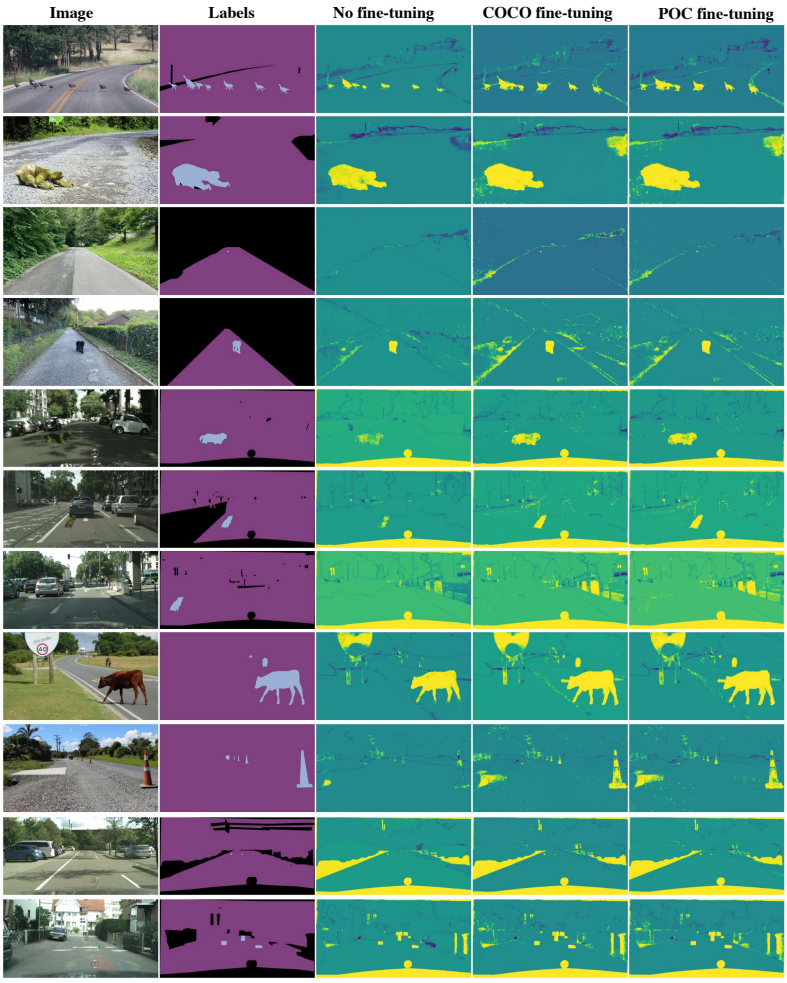

Fig. 24: RbA anomaly scores on samples from related datasets (see Fig. 1).

I Additional qualitative results

In this section we show additional qualitative results for the dataset extension experiments (*c.f.* Sec. 5). We show predictions on all evaluated datasets for DLV3+, ConvNeXt and Segmenter models.

Fig. 25: DLV3+ predictions on additional web images.

Fig. 26: ConvNeXt predictions on additional web images.

Fig. 27: Segmenter predictions on additional web images.

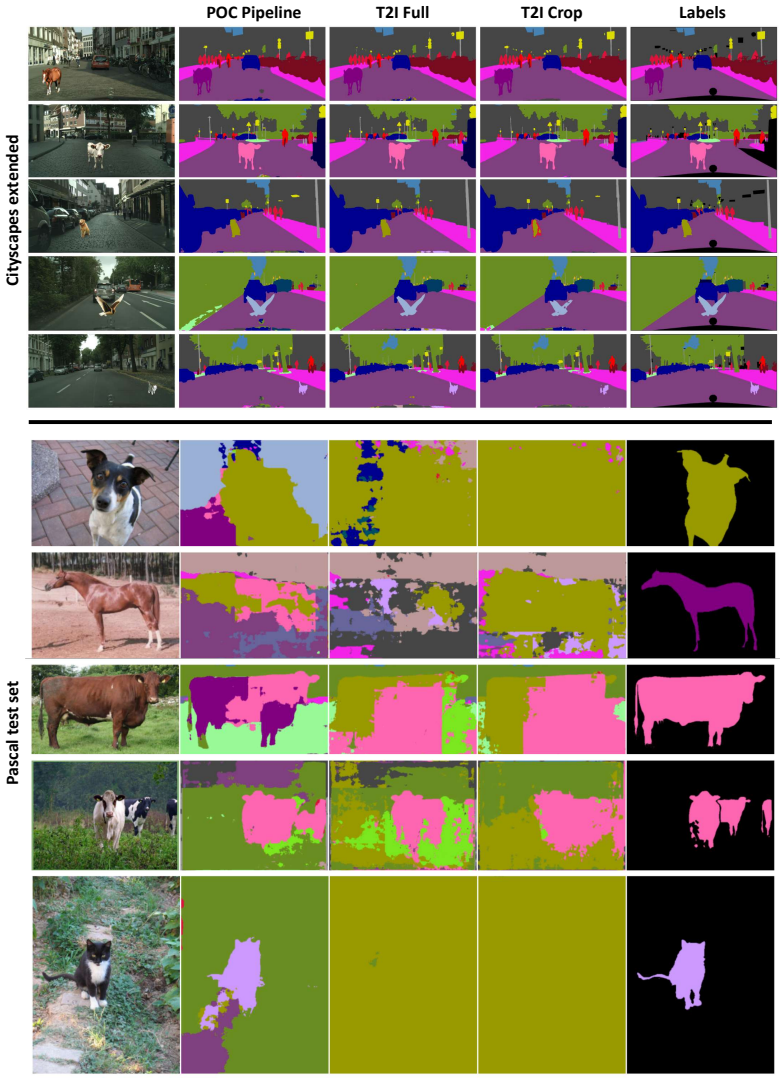

Fig. 28: DLV3+ predictions on extended Cityscapes (*POC A*) and Pascal validation sets.

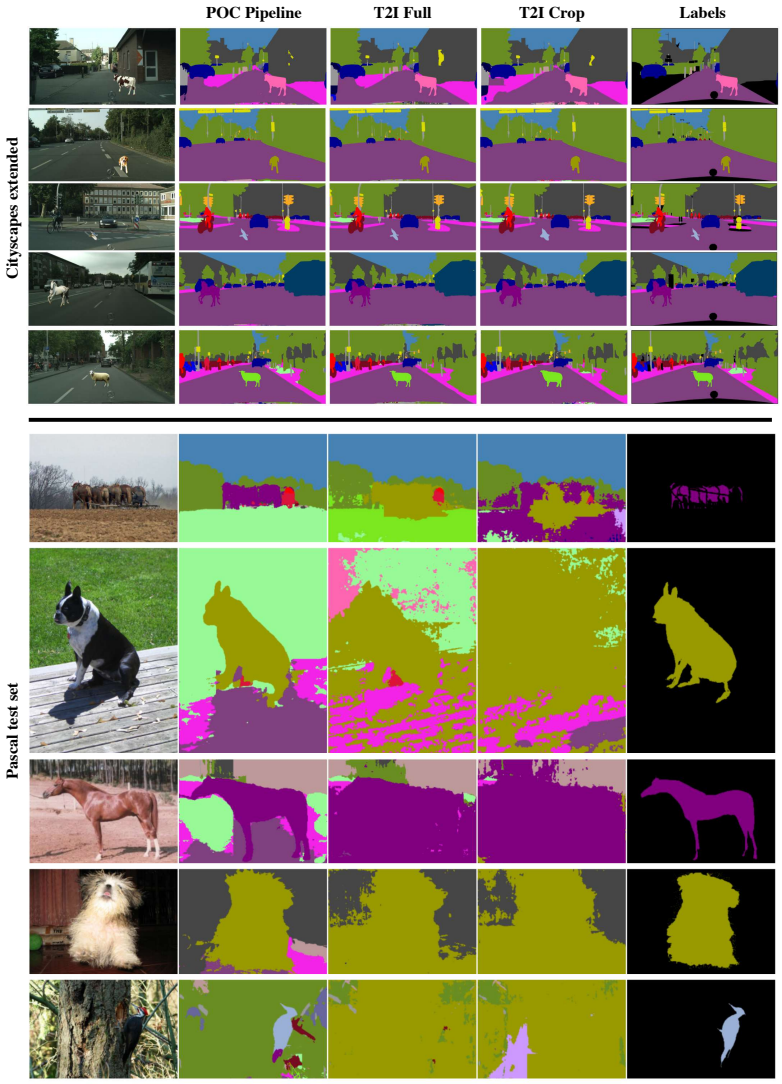

Fig. 29: ConvNeXt predictions on extended Cityscapes (*POC A*) and Pascal validation sets.

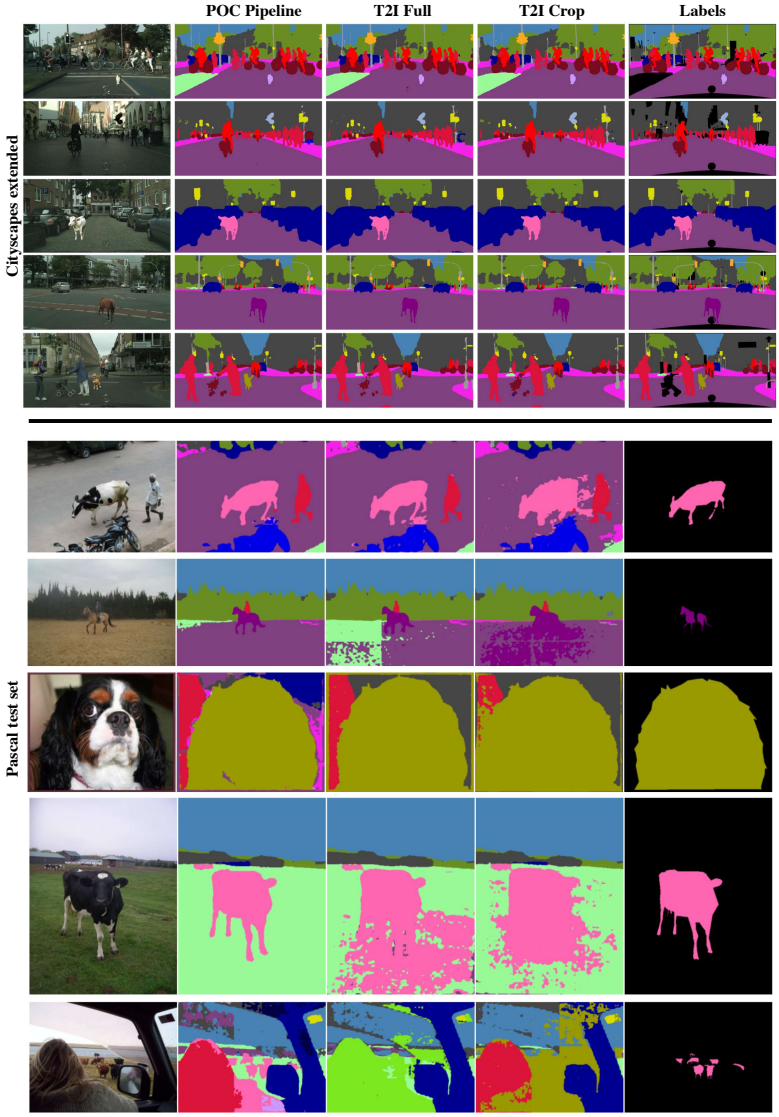

Fig. 30: Segmenter predictions on extended Cityscapes (*POC A*) and Pascal validation sets.