Placing Objects in Context via Inpainting for Out-of-distribution Segmentation

Pau de Jorge Aranda, Riccardo Volpi, Puneet Dokania, Philip Torr, Grégory Rogez

Motivation

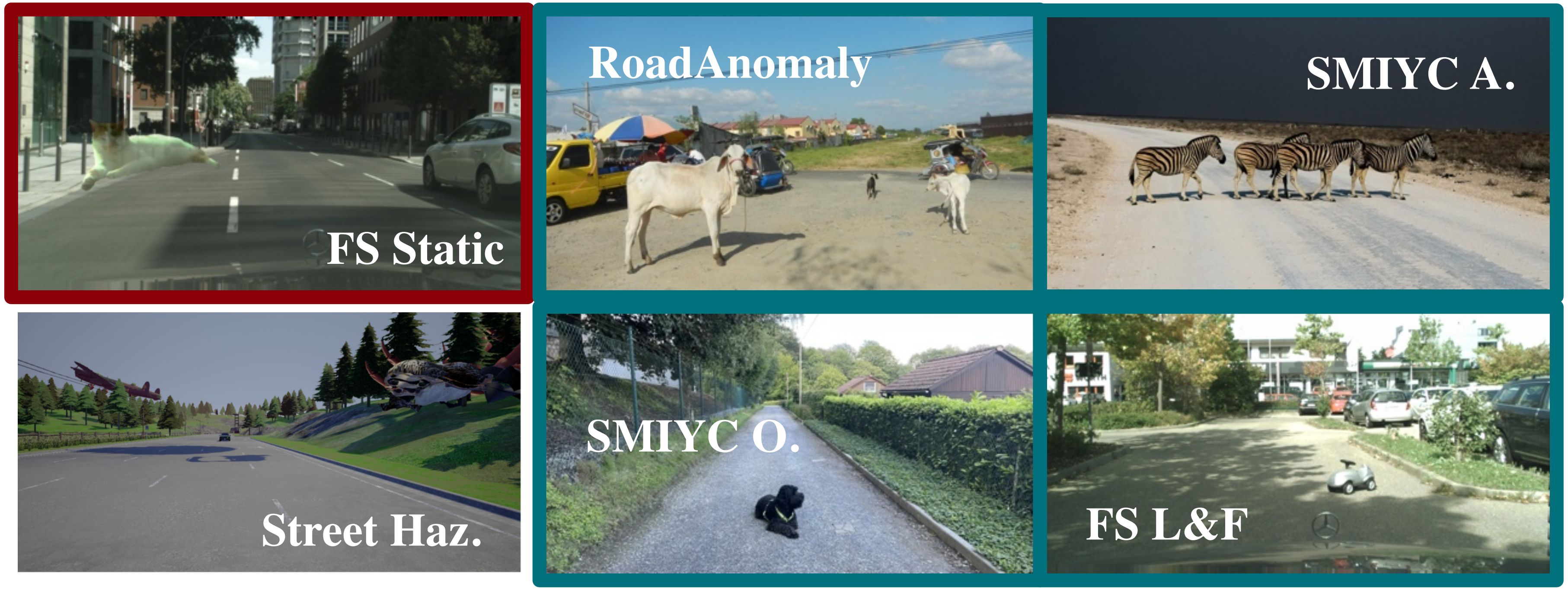
- Agents deployed “in the wild” will eventually face **novel objects**
- For a safe deployment of agents, it is crucial they are able to detect such anomalies *e.g.* **Anomaly segmentation**
- Acquiring anomaly segmentation datasets is highly **inefficient** (and even **hazardous** in some cases)

Prior work

- **Object stitching:** Efficient but unrealistic

Prior work

- **Object stitching:** Efficient but unrealistic
- **Image collection:** Realistic but costly and strong domain shift



Prior work

- **Object stitching:** Efficient but unrealistic
- **Image collection:** Realistic but costly and high domain shift
- **Simulation:** Flexible but costly and high shift

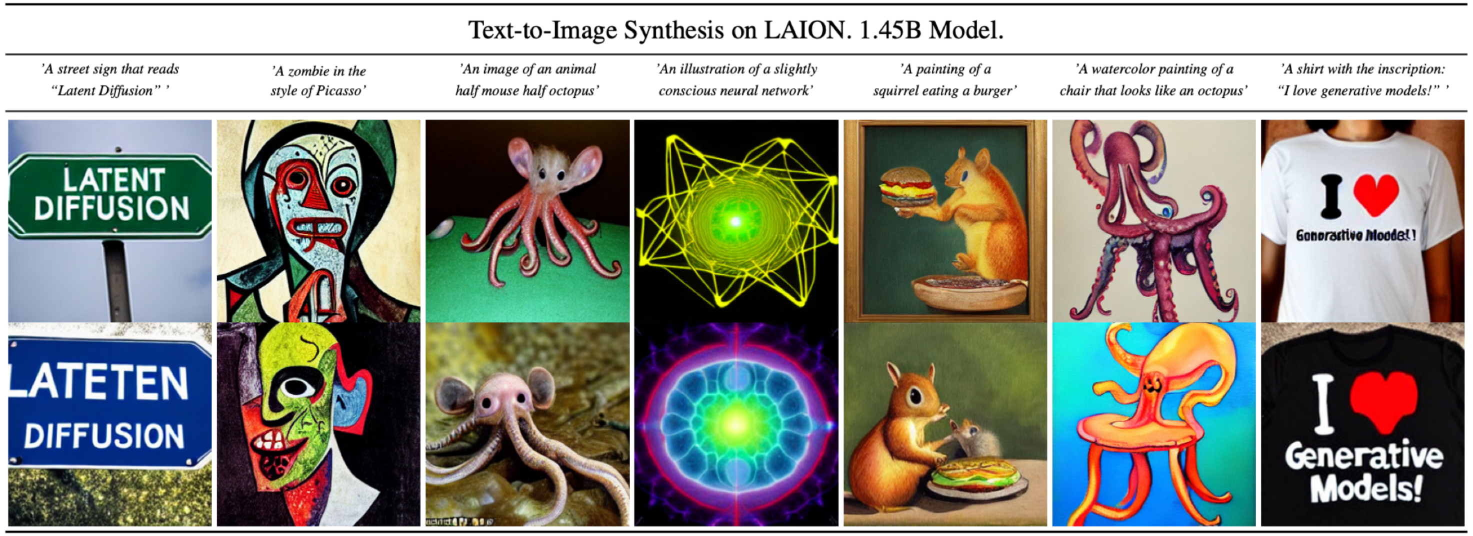
Prior work

- **Object stitching:** Efficient but unrealistic
- **Image collection:** Realistic but costly and high domain shift
- **Simulation:** Flexible but costly and high shift

Desiderata: (i) No domain shift (ii) Realistic anomalies (iii) Dynamic generation (iv) Low set-up costs

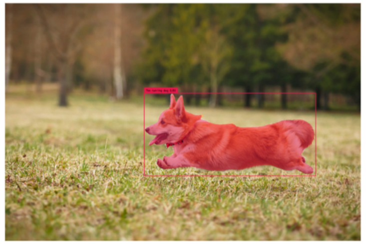
Prior work

- Recent **text2image** models can generate realistic images based on flexible text "captions"

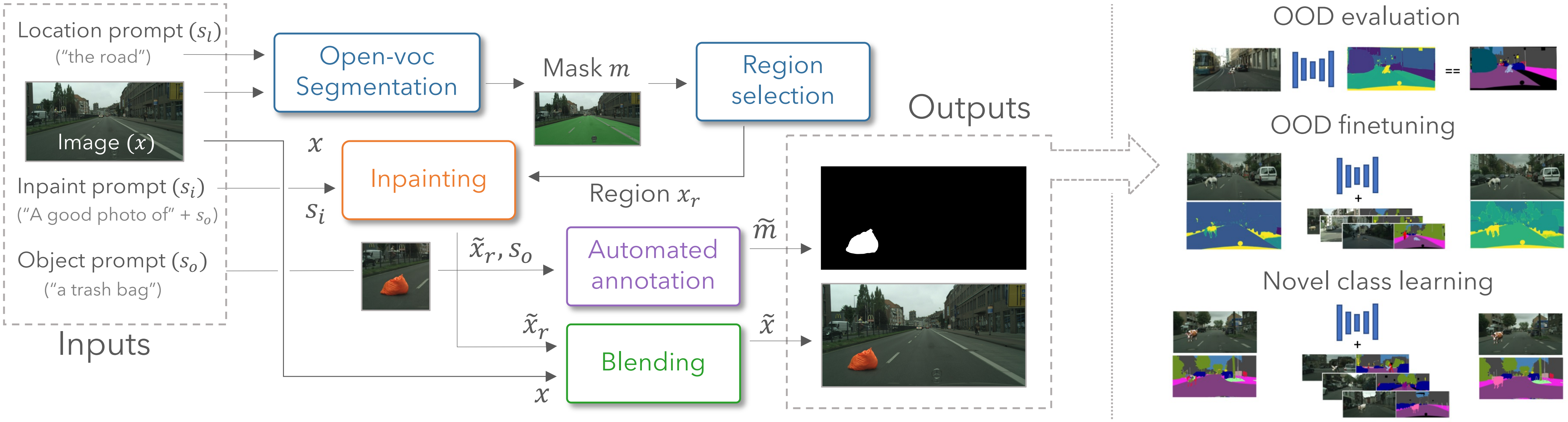


- These models can be fine-tuned to only **inpaint** content inside masked area consistent with image context

- **Open-vocabulary** models can detect and segment objects based on text prompts

Text Prompt	Input Image	GroundingDINO Annotated Image	Grounded-SAM Annotated Image
The running dog			

The POC pipeline (ours)



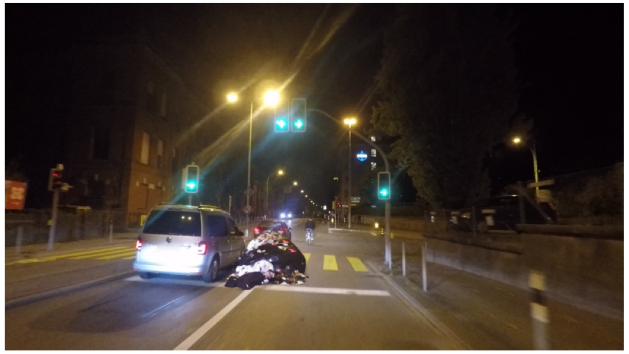
The POC datasets (ours)

Cityscapes-POC



ACDC-POC



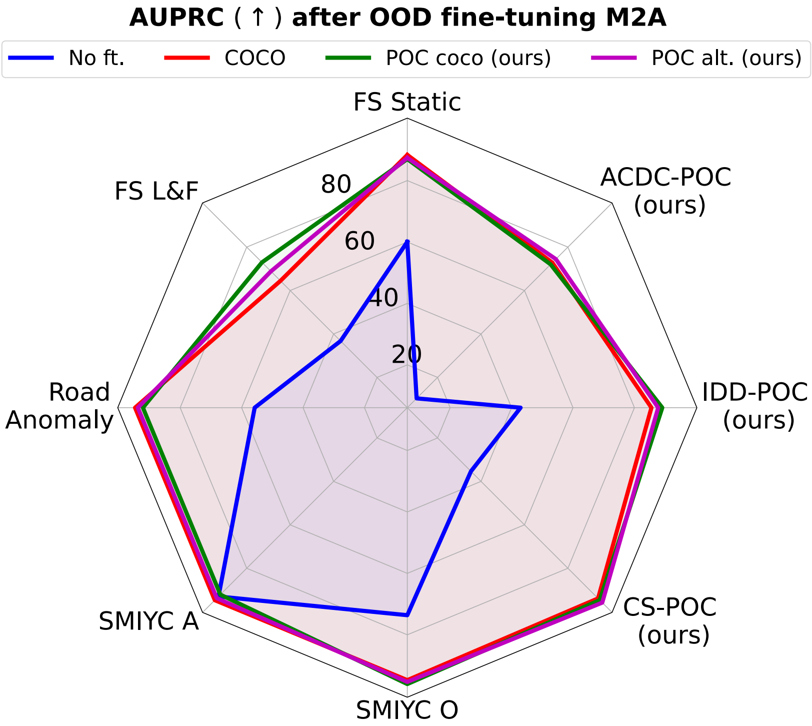


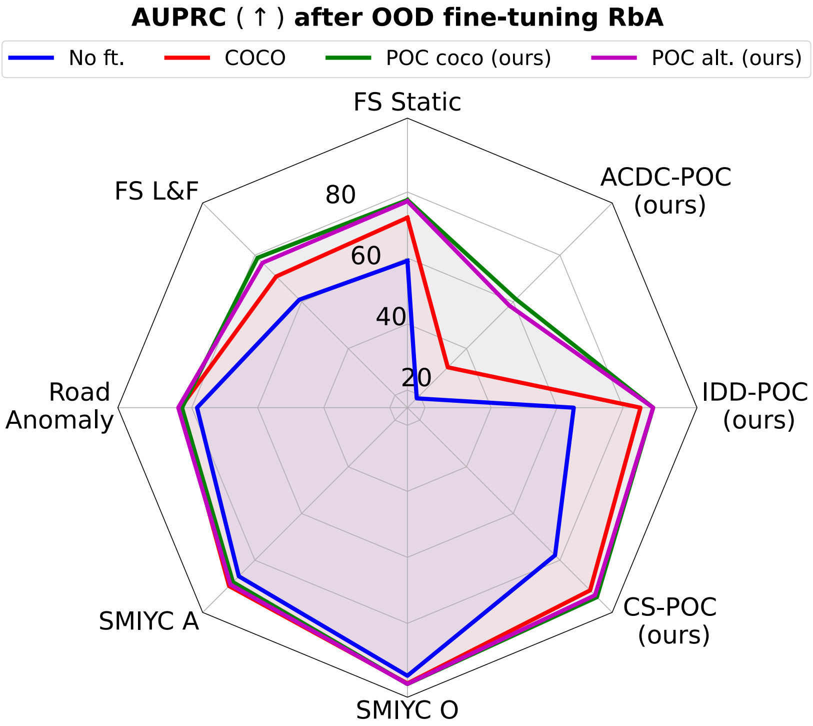
IDD-POC

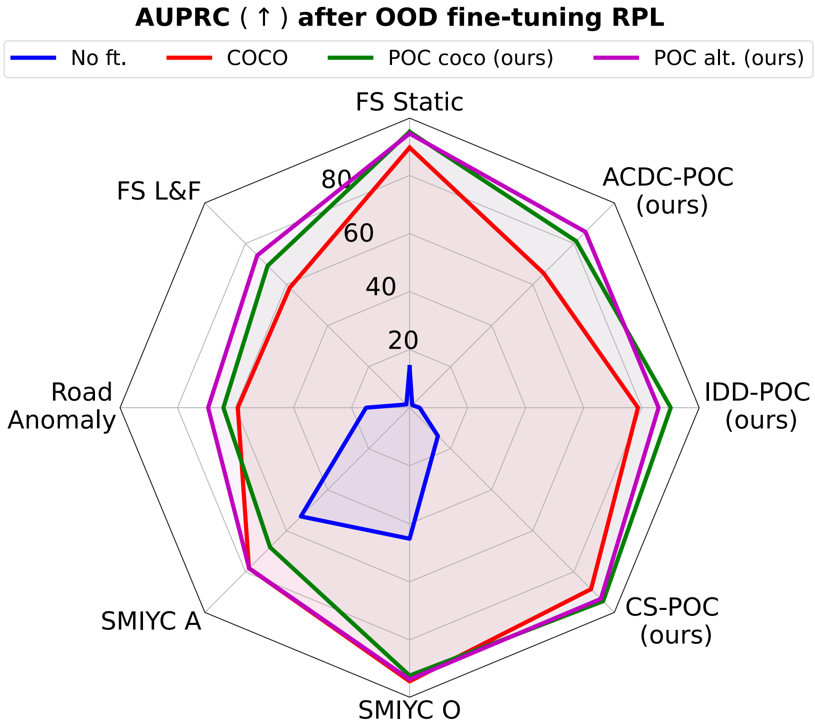


POC for Anomaly fine-tuning

State-of-the-art anomaly segmentation methods rely on **anomaly fine-tuning**



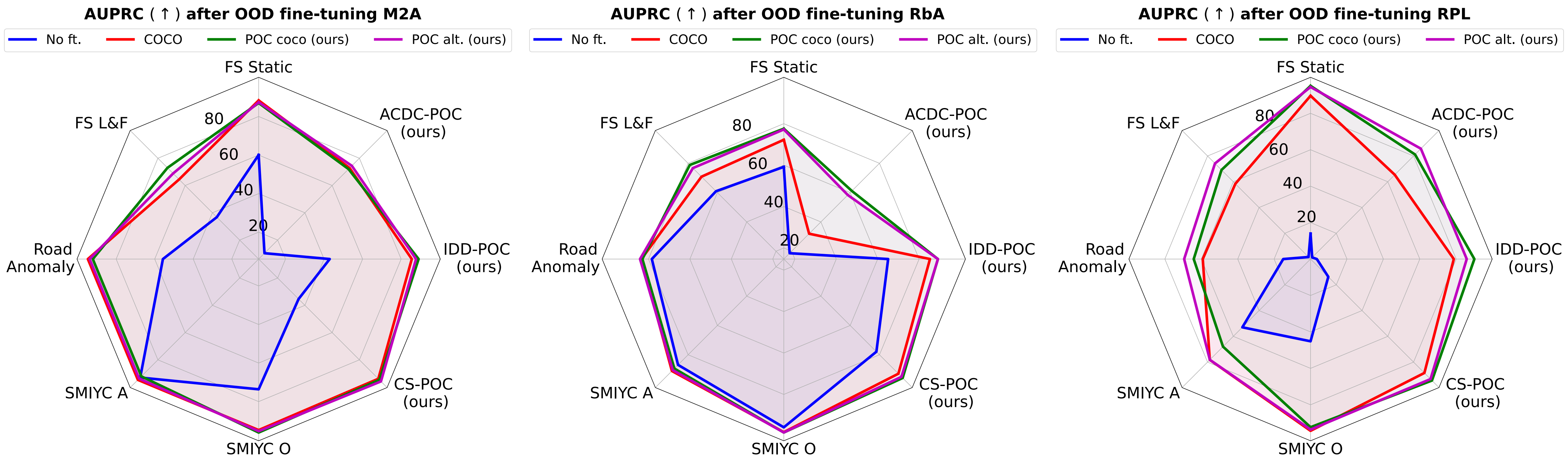




POC for Anomaly fine-tuning

State-of-the-art anomaly segmentation methods rely on **anomaly fine-tuning**

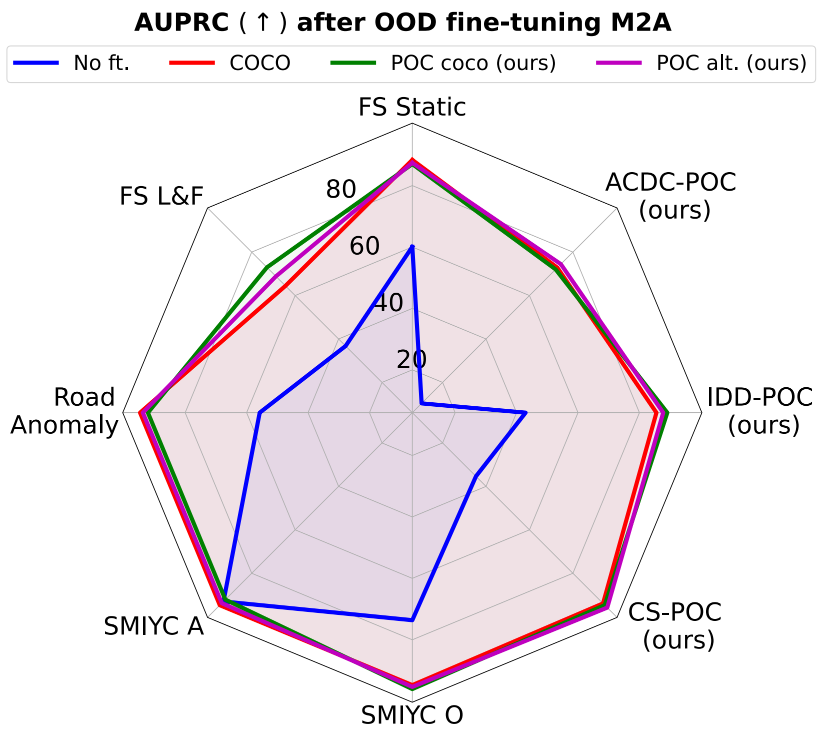
➢ Usually done by stitching COCO objects to Cityscapes images leading to unrealistic compositions

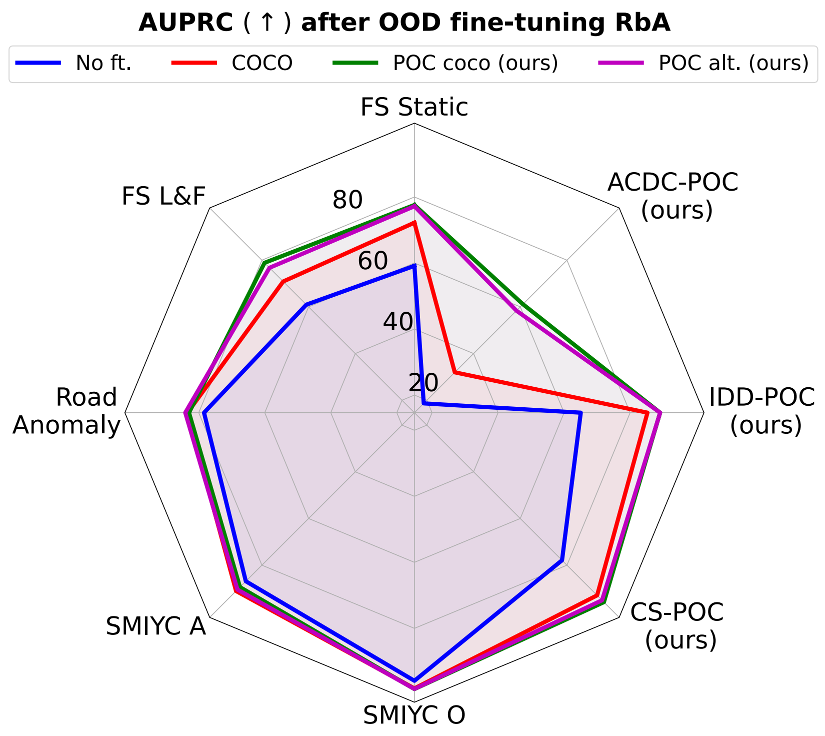


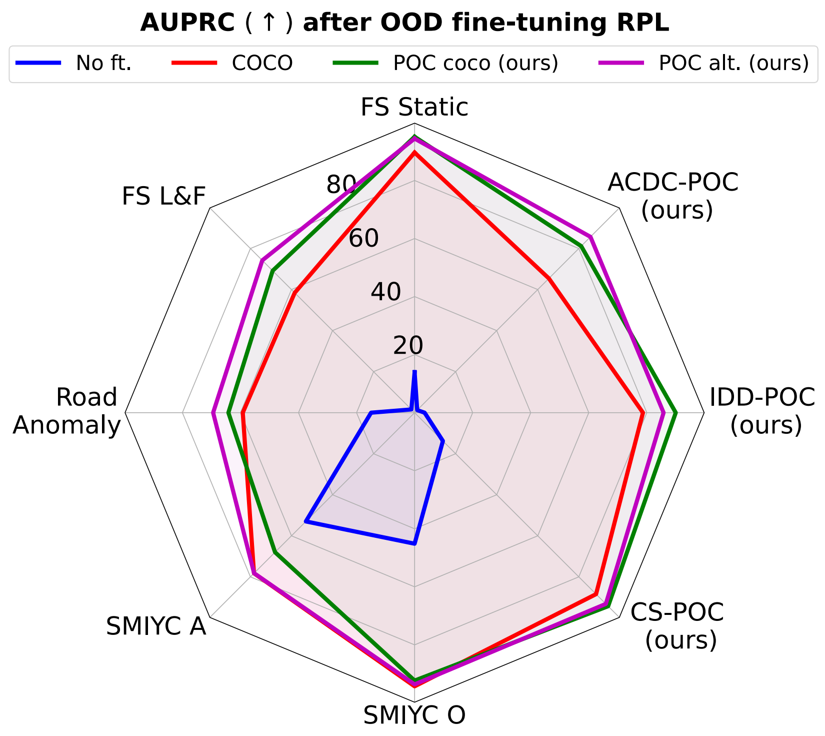
POC for Anomaly fine-tuning

State-of-the-art anomaly segmentation methods rely on **anomaly fine-tuning**

- Usually done by stitching COCO objects to Cityscapes images leading to unrealistic compositions
- We test our POC pipeline to generate fine-tuning samples with more realistic anomalies



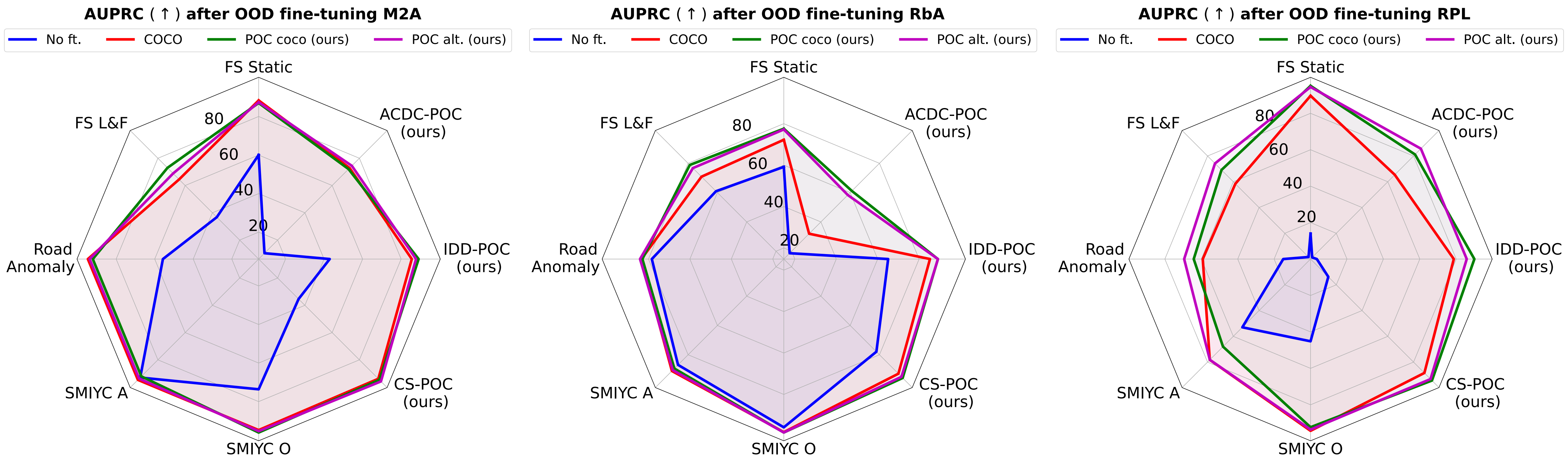




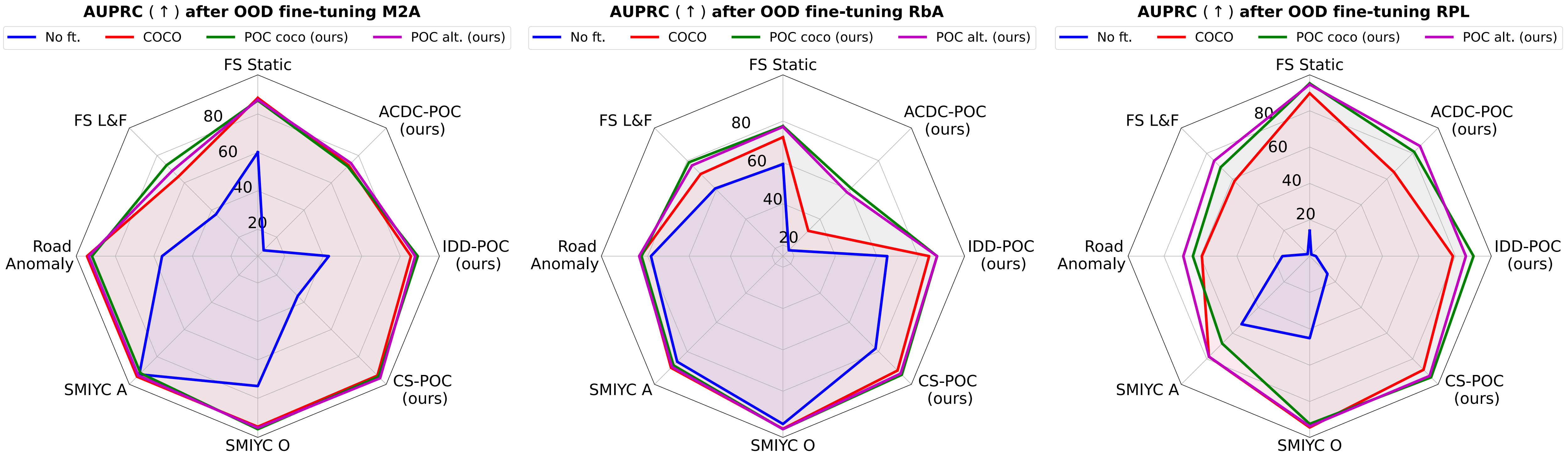
POC for Anomaly fine-tuning

State-of-the-art anomaly segmentation methods rely on **anomaly fine-tuning**

- Usually done by stitching COCO objects to Cityscapes images leading to unrealistic compositions
- We test our POC pipeline to generate fine-tuning samples with more realistic anomalies
- We generate two datasets, one with COCO objects and one with alternative objects.



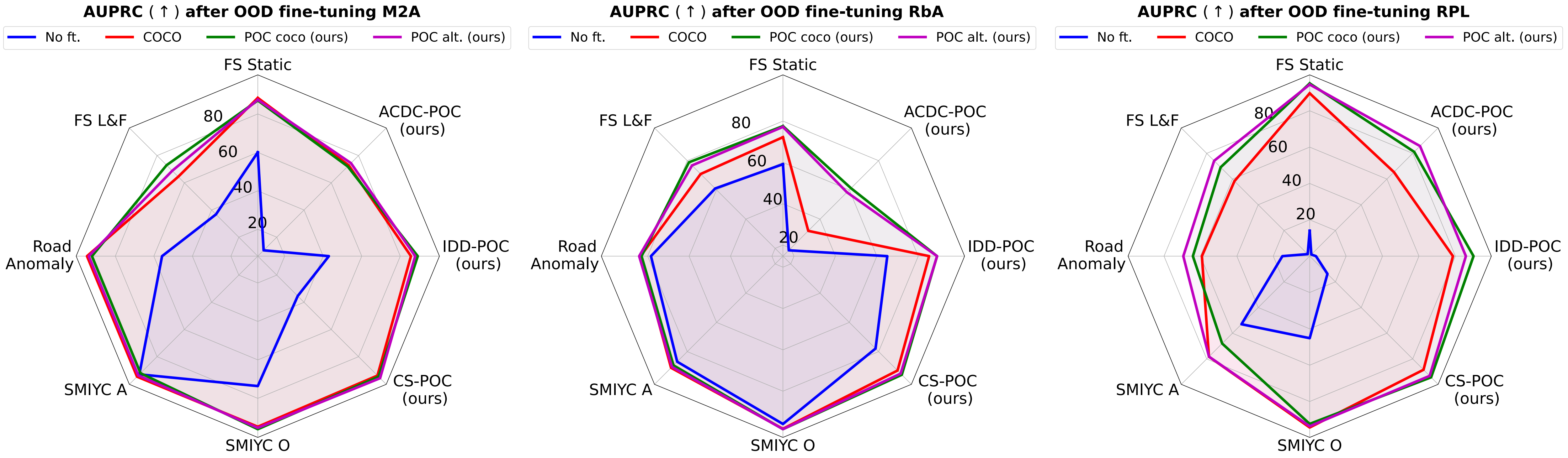
POC for Anomaly fine-tuning



Main takeaways:

- Fine-tuning with our synthetic data brings significant improvements

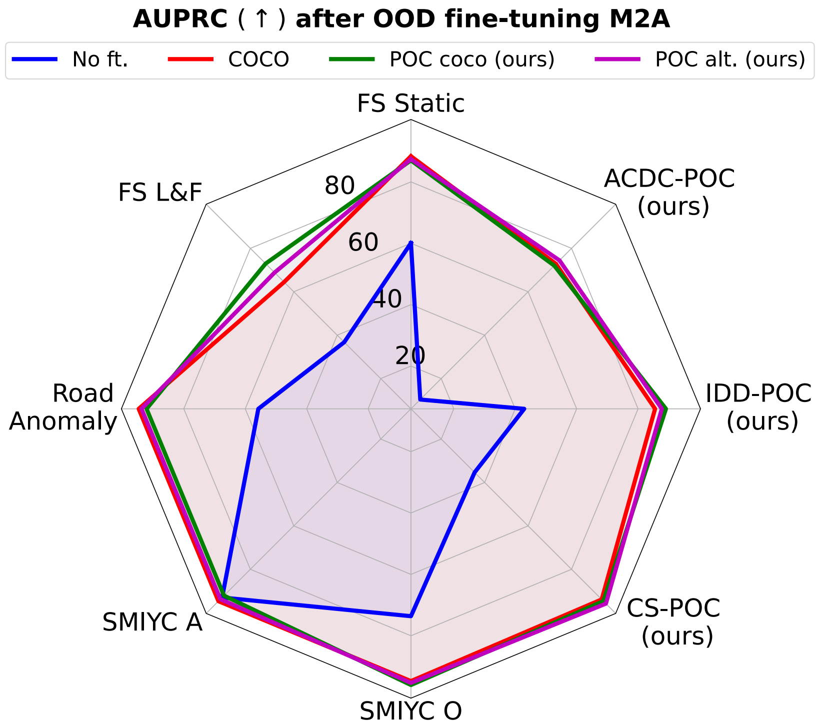
POC for Anomaly fine-tuning

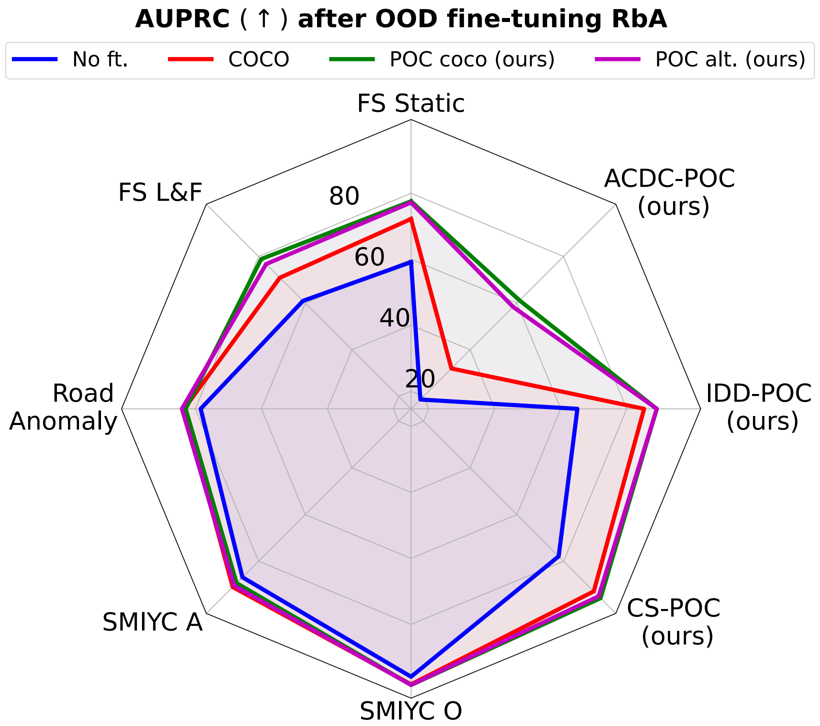


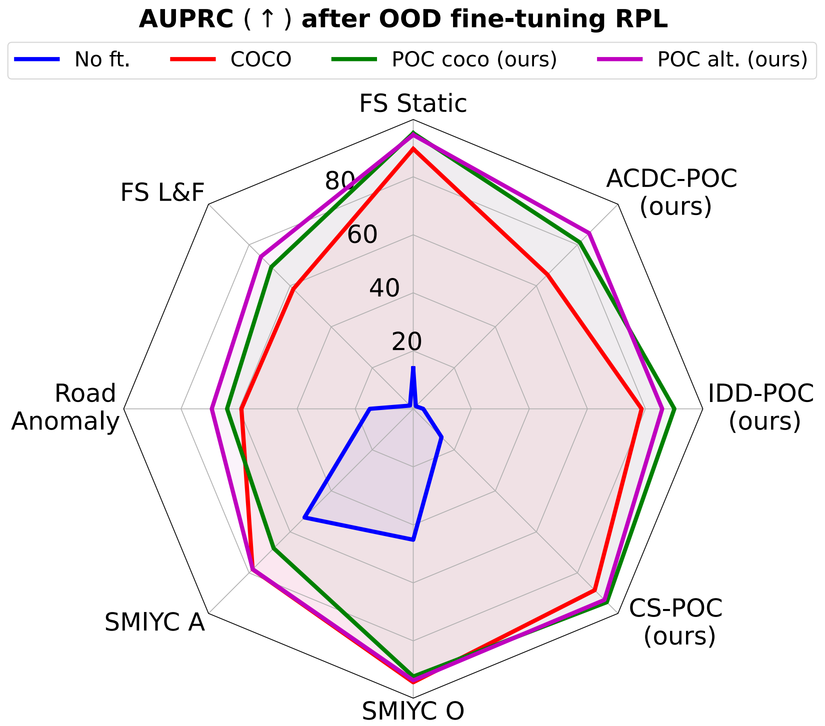
Main takeaways:

- Fine-tuning with our synthetic data brings significant improvements
- Anomaly ft. seems to be rather robust to choice of anomaly classes

POC for Anomaly fine-tuning



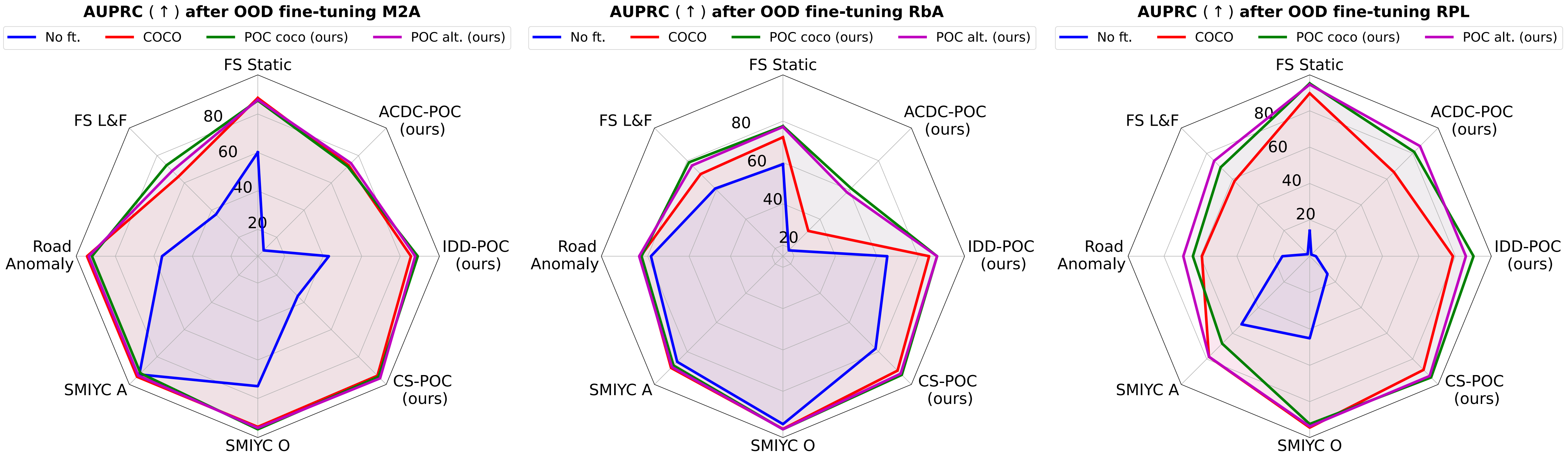




Main takeaways:

- Fine-tuning with our synthetic data brings significant improvements
- Anomaly ft. seems to be rather robust to choice of anomaly classes
- COCO ft. improvements in our POC evaluation sets is consistent with other benchmarks

POC for Anomaly fine-tuning

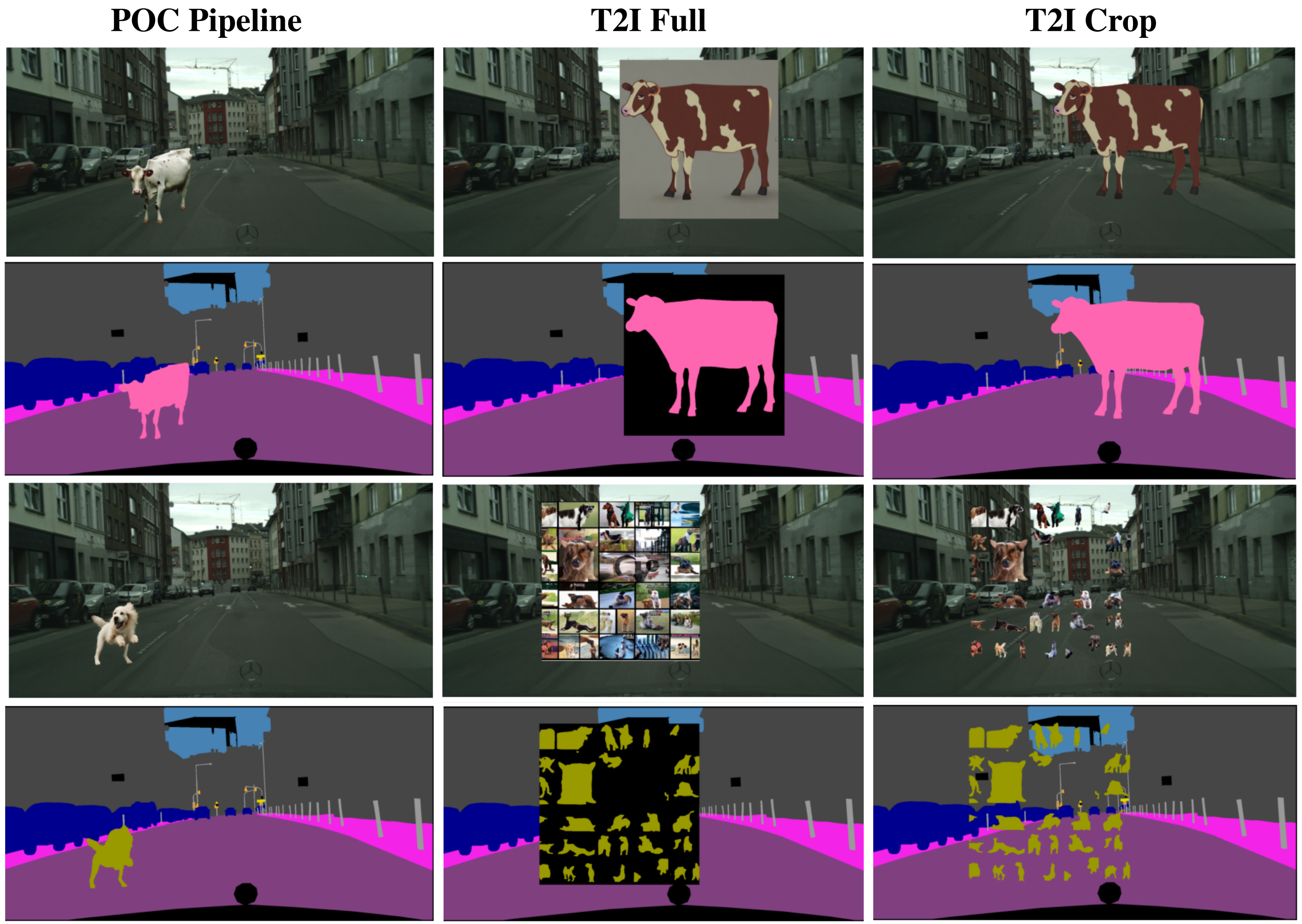


Main takeaways:

- Fine-tuning with our synthetic data brings significant improvements
- Anomaly ft. seems to be rather robust to choice of anomaly classes
- COCO ft. improvements in our POC evaluation sets is consistent with other benchmarks
- Drop in performance as we increase domain shift (with same model and anomaly classes)

POC to learn new classes

We extend Ciyscapes with animal classes and evaluate in Pascal dataset



POC to learn new classes

- Segmenter with POC data achieves competitive mIoU with a baseline trained directly on Pascal
- Good generalization needed to leverage POC images
- Adding known (synth) classes can boost generalization
- Segmenter also surpasses open-vocabulary GSAM on Pascal which is used to obtain the POC labels

Model	**Train set**	**CS** (19 cl.)	**CS ext.** (19 + 6 cl.)	**Pascal** (animal)	**Pascal** (citysc.)
DLV3+	Pascal (baseline)	–	–	80.57	85.81
	CS (baseline)	79.09	–	–	42.22
	T2I Full	77.13	73.2	28.32	23.51
	T2I Crop	78.23	81.49	26.15	25.09
	POC A	**79.98**	**84.09**	**30.43**	35.83
	POC CS+A	79.9	83.8	28.11	**53.57**
CNXT	Pascal (baseline)	–	–	94.43	93.92
	CS (baseline)	81.57	–	–	70.49
	T2I Full	82.26	82.99	60.41	65.6
	T2I Crop	82.43	86.04	61.09	67.55
	POC A	**82.94**	**86.68**	65.46	70.87
	POC CS+A	82.54	86.09	**69.75**	**82.43**
Segm.	Pascal (baseline)	–	–	94.75	91.43
	CS (baseline)	76.19	–	–	79.87
	T2I Full	77.19	79.46	81.96	74.32
	T2I Crop	77.62	81.27	75.95	75.44
	POC A	**78.48**	**82.28**	92.4	79.1
	POC CS+A	78.39	81.92	**93.14**	**89.55**
GSAM	Open Vocab.	42.0	41.06	75.13	76.08

POC to learn new classes

- Segmenter with POC data achieves competitive mIoU with a baseline trained directly on Pascal
- Good generalization needed to leverage POC images
- Adding known (synth) classes can boost generalization
- Segmenter also surpasses open-vocabulary GSAM on Pascal which is used to obtain the POC labels

Model	**Train set**	**CS** (19 cl.)	**CS ext.** (19 + 6 cl.)	**Pascal** (animal)	**Pascal** (citysc.)
DLV3+	Pascal (baseline)	–	–	80.57	85.81
	CS (baseline)	79.09	–	–	42.22
	T2I Full	77.13	73.2	28.32	23.51
	T2I Crop	78.23	81.49	26.15	25.09
	POC A	**79.98**	**84.09**	**30.43**	35.83
	POC CS+A	79.9	83.8	28.11	**53.57**
CNXT	Pascal (baseline)	–	–	94.43	93.92
	CS (baseline)	81.57	–	–	70.49
	T2I Full	82.26	82.99	60.41	65.6
	T2I Crop	82.43	86.04	61.09	67.55
	POC A	**82.94**	**86.68**	65.46	70.87
	POC CS+A	82.54	86.09	**69.75**	**82.43**
Segm.	Pascal (baseline)	–	–	94.75	91.43
	CS (baseline)	76.19	–	–	79.87
	T2I Full	77.19	79.46	81.96	74.32
	T2I Crop	77.62	81.27	75.95	75.44
	POC A	**78.48**	**82.28**	92.4	79.1
	POC CS+A	78.39	81.92	**93.14**	**89.55**
GSAM	Open Vocab.	42.0	41.06	75.13	76.08

POC to learn new classes

- Segmenter with POC data achieves competitive mIoU with a baseline trained directly on Pascal
- Good generalization needed to leverage POC images
- Adding known (synth) classes can boost generalization
- Segmenter also surpasses open-vocabulary GSAM on Pascal which is used to obtain the POC labels

Model	**Train set**	**CS** (19 cl.)	**CS ext.** (19 + 6 cl.)	**Pascal** (animal)	**Pascal** (citysc.)
DLV3+	Pascal (baseline)	–	–	80.57	85.81
	CS (baseline)	79.09	–	–	42.22
	T2I Full	77.13	73.2	28.32	23.51
	T2I Crop	78.23	81.49	26.15	25.09
	POC A	**79.98**	**84.09**	**30.43**	35.83
	POC CS+A	79.9	83.8	28.11	**53.57**
CNXT	Pascal (baseline)	–	–	94.43	93.92
	CS (baseline)	81.57	–	–	70.49
	T2I Full	82.26	82.99	60.41	65.6
	T2I Crop	82.43	86.04	61.09	67.55
	POC A	**82.94**	**86.68**	65.46	70.87
	POC CS+A	82.54	86.09	**69.75**	**82.43**
Segm.	Pascal (baseline)	–	–	94.75	91.43
	CS (baseline)	76.19	–	–	79.87
	T2I Full	77.19	79.46	81.96	74.32
	T2I Crop	77.62	81.27	75.95	75.44
	POC A	**78.48**	**82.28**	92.4	79.1
	POC CS+A	78.39	81.92	**93.14**	**89.55**
GSAM	Open Vocab.	42.0	41.06	75.13	76.08

POC to learn new classes

- Segmenter with POC data achieves competitive mIoU with a baseline trained directly on Pascal
- Good generalization needed to leverage POC images
- Adding known (synth) classes can boost generalization
- Segmenter also surpasses open-vocabulary GSAM on Pascal which is used to obtain the POC labels

Model	**Train set**	**CS** (19 cl.)	**CS ext.** (19 + 6 cl.)	**Pascal** (animal)	**Pascal** (citysc.)
DLV3+	Pascal (baseline)	–	–	80.57	85.81
	CS (baseline)	79.09	–	–	42.22
	T2I Full	77.13	73.2	28.32	23.51
	T2I Crop	78.23	81.49	26.15	25.09
	POC A	**79.98**	**84.09**	**30.43**	35.83
	POC CS+A	79.9	83.8	28.11	**53.57**
CNXT	Pascal (baseline)	–	–	94.43	93.92
	CS (baseline)	81.57	–	–	70.49
	T2I Full	82.26	82.99	60.41	65.6
	T2I Crop	82.43	86.04	61.09	67.55
	POC A	**82.94**	**86.68**	65.46	70.87
	POC CS+A	82.54	86.09	**69.75**	**82.43**
Segm.	Pascal (baseline)	–	–	94.75	91.43
	CS (baseline)	76.19	–	–	79.87
	T2I Full	77.19	79.46	81.96	74.32
	T2I Crop	77.62	81.27	75.95	75.44
	POC A	**78.48**	**82.28**	92.4	79.1
	POC CS+A	78.39	81.92	**93.14**	**89.55**
GSAM	Open Vocab.	42.0	41.06	75.13	76.08

Summary

- Leverage generative models to build a pipeline to insert novel objects into images realistically.
- Build new anomaly segmentation datasets.
- Show POC images for anomaly fine-tuning can improve over object stitching with recent methods.
- Use POC to extend an existing dataset with new classes obtaining competitive performance.