NEURAL INFORMATION PROCESSING SYSTEMS

AUCSeg:
AUC-oriented Pixel-level Long-tail Semantic Segmentation

Boyu Han, Qianqian Xu*, Zhiyong Yang, Shilong Bao,
Peisong Wen, Yangbangyan Jiang, Qingming Huang*

Key Lab. of Intelligent Information Processing, Institute of Computing Technology, CAS
School of Computer Science and Tech., University of Chinese Academy of Sciences
Peng Cheng Laboratory
Key Laboratory of Big Data Mining and Knowledge Management, CAS

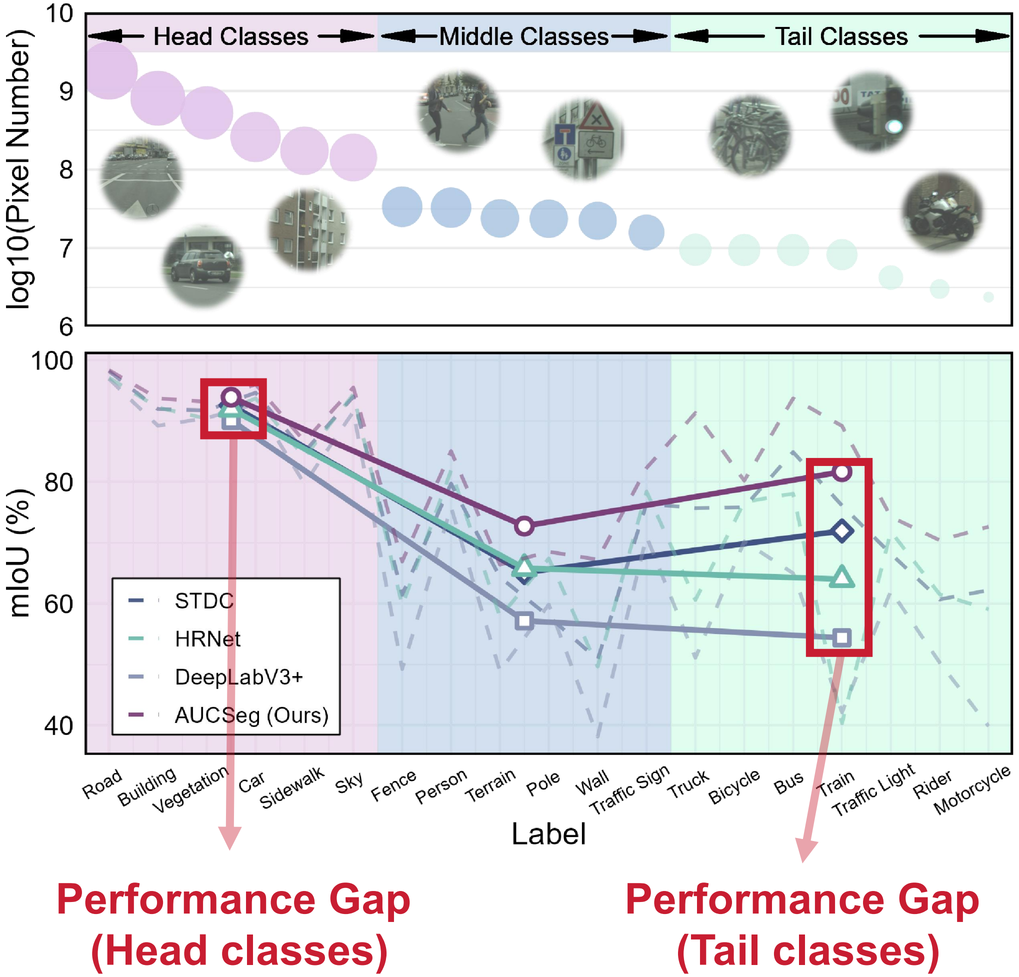
Project Page

Pixel-level Long-tail Semantic Segmentation (PLSS)

input

output

- Semantic Segmentation assigns a label to each pixel in the input image and is commonly used in fields such as autonomous driving and disease diagnosis.
- Different models show little variation in head class performance, with tail classes being the main factor limiting the model performance.



Related Works on PLSS

PLSS

1 **Developing carefully designed backbones for long-tail distributions[1,2]**
Leaving the effect of loss functions unconsidered

2 **Conducting empirical studies on the loss functions[3,4]**
Lacking their theoretical impact on the generalization performance

Goal

Can we find a theoretically grounded loss function for PLSS on top of SOTA backbones?

[1] Hanzhe Hu et al. Semi-supervised semantic segmentation via adaptive equalization learning. In NeurIPS, pages 22106–22118, 2021.
[2] Yuchao Wang et al. Balancing logit variation for long-tailed semantic segmentation. In CVPR, pages 19561–19573, 2023.
[3] Songyang Zhang et al. Distribution alignment: A unified framework for long-tail visual recognition. In CVPR, pages 2361–2370, 2021.
[4] Yuchao Wang et al. Balancing logit variation for long-tailed semantic segmentation. In CVPR, pages 19561–19573, 2023.

AUCSeg - overview

NEURAL INFORMATION PROCESSING SYSTEMS

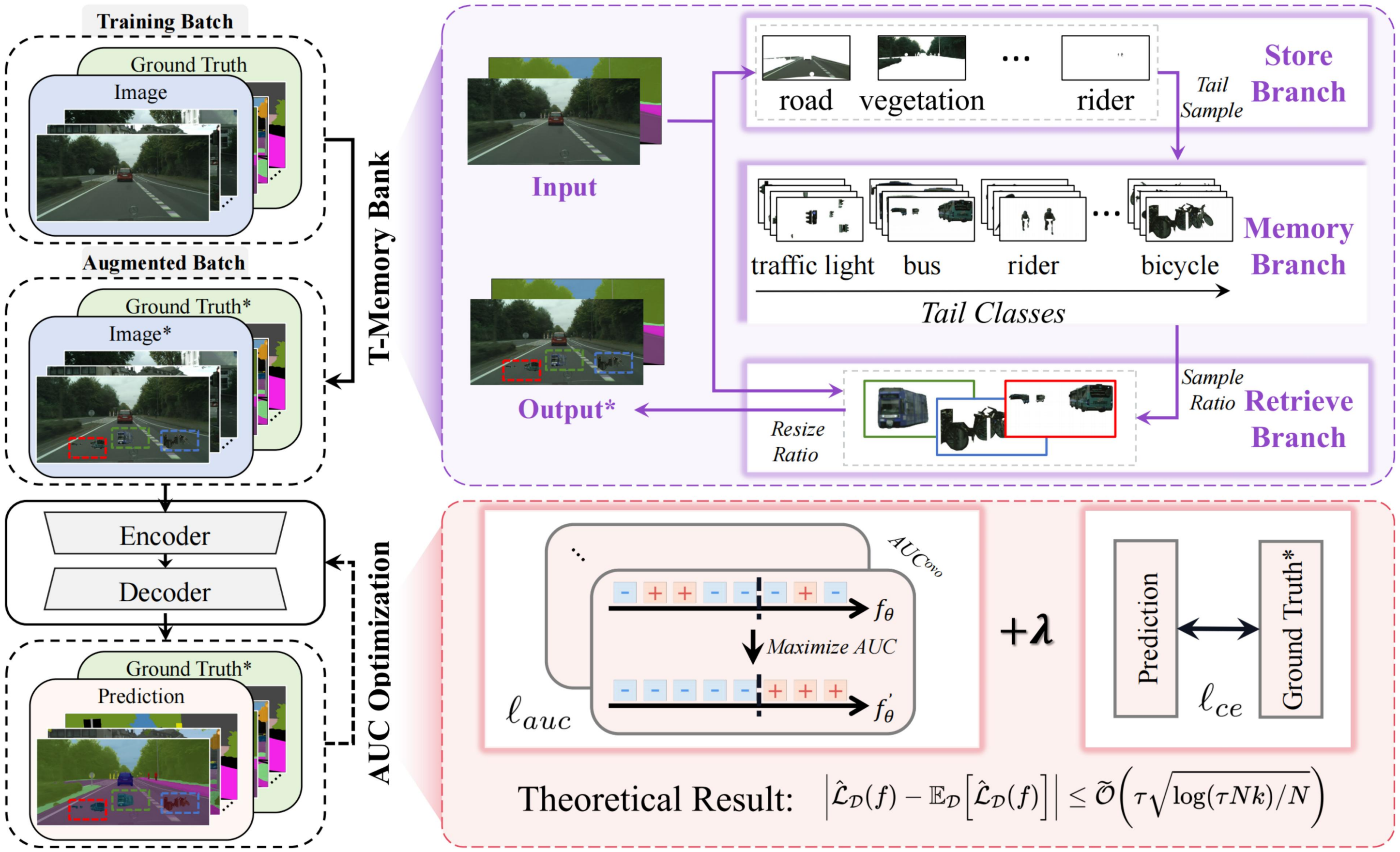
AUCSeg is a generic optimization method that can be directly applied to any SOTA backbone for semantic segmentation.

T-Memory Bank

An effective augmentation scheme to ensure efficient optimization of the proposed AUC loss.

AUC Optimization

A theoretically grounded loss function explored for PLSS.



AUCSeg - overview

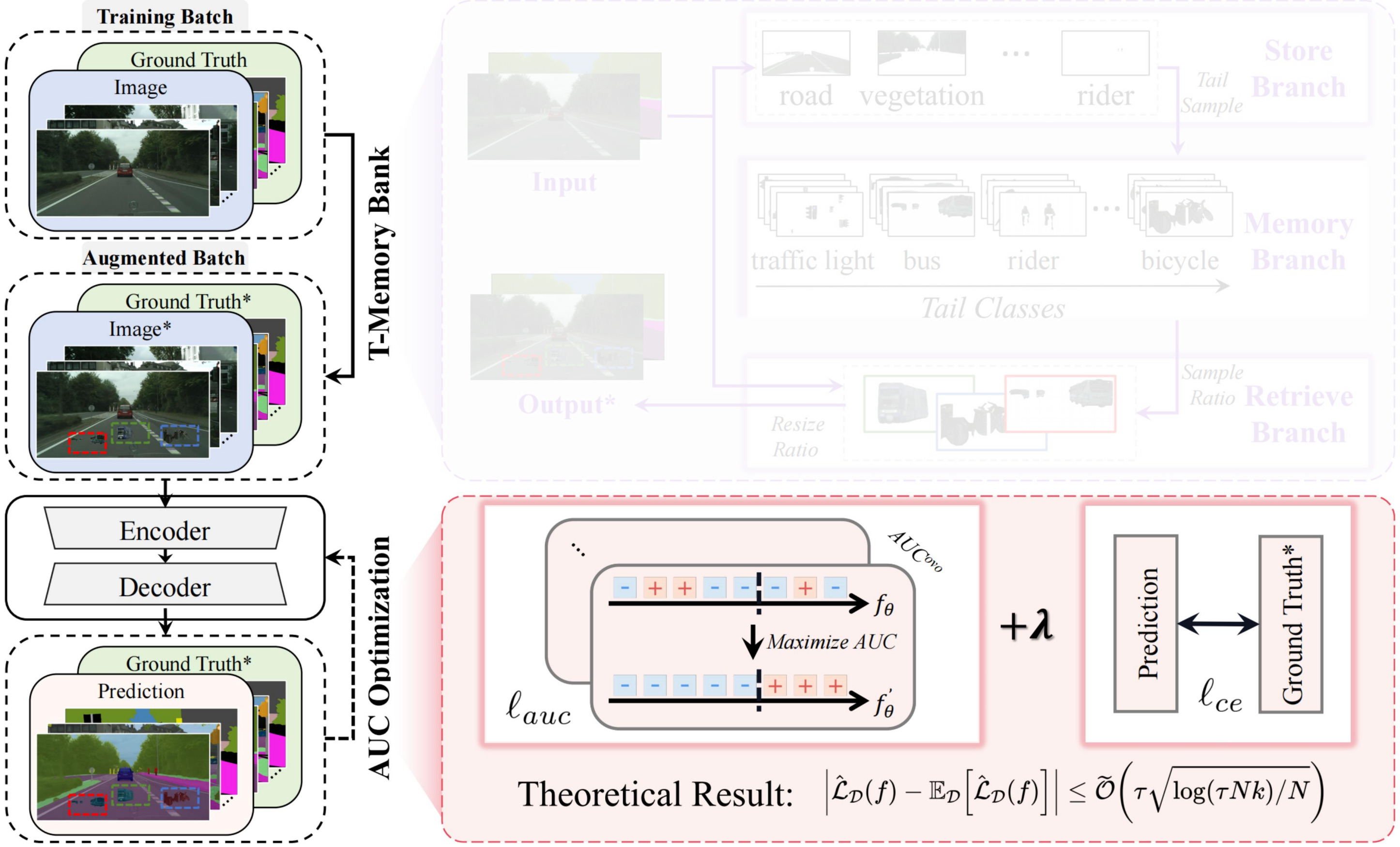
AUCSeg is a generic optimization method that can be directly applied to any SOTA backbone for semantic segmentation.

T-Memory Bank

An effective augmentation scheme to ensure efficient optimization of the proposed AUC loss.

AUC Optimization

A theoretically grounded loss function explored for PLSS.



AUCSeg - overview

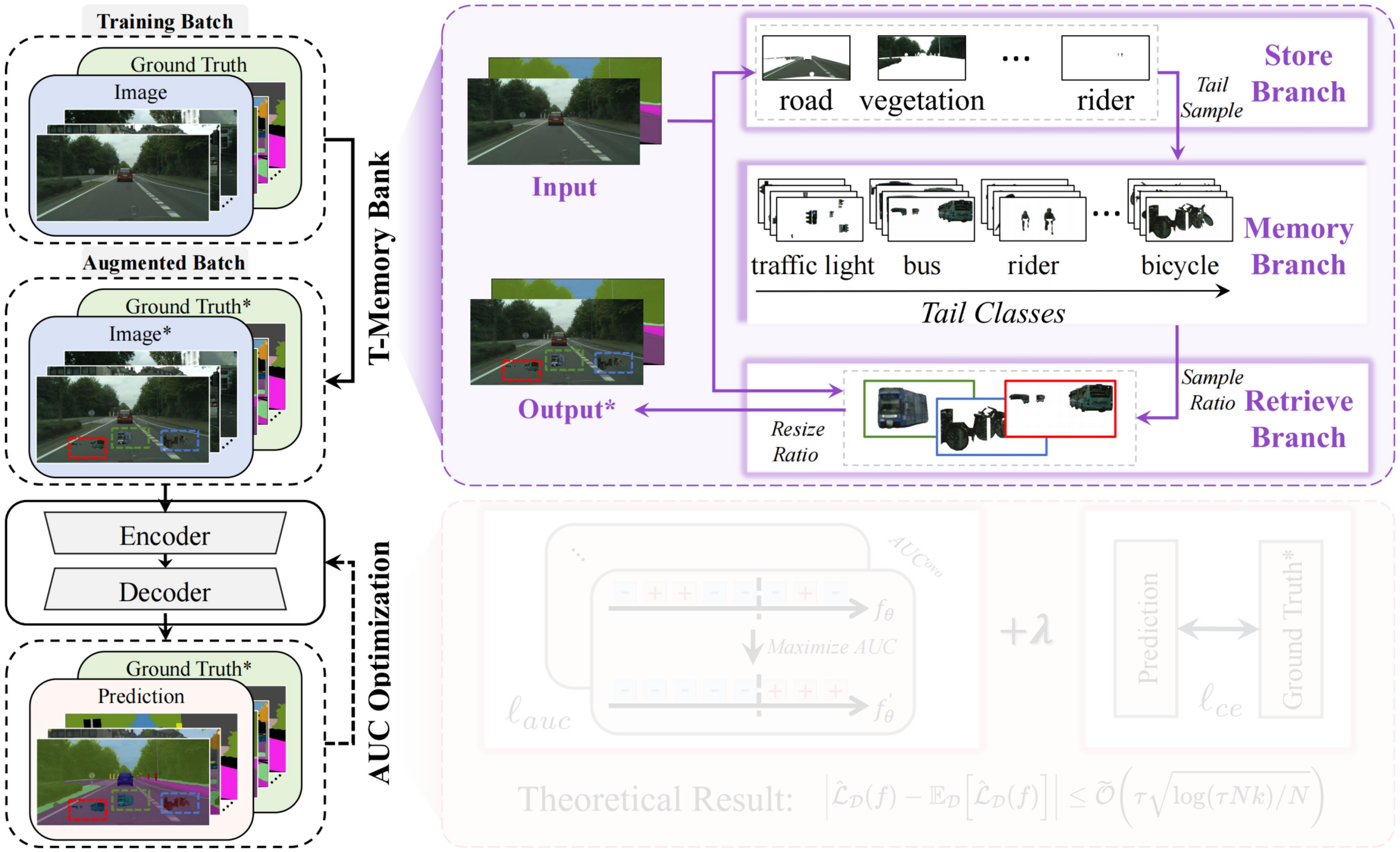
AUCSeg is a generic optimization method that can be directly applied to any SOTA backbone for semantic segmentation.

T-Memory Bank

An effective augmentation scheme to ensure efficient optimization of the proposed AUC loss.

AUC Optimization

A theoretically grounded loss function explored for PLSS.



AUCSeg - AUC Optimization

NEURAL INFORMATION PROCESSING SYSTEMS

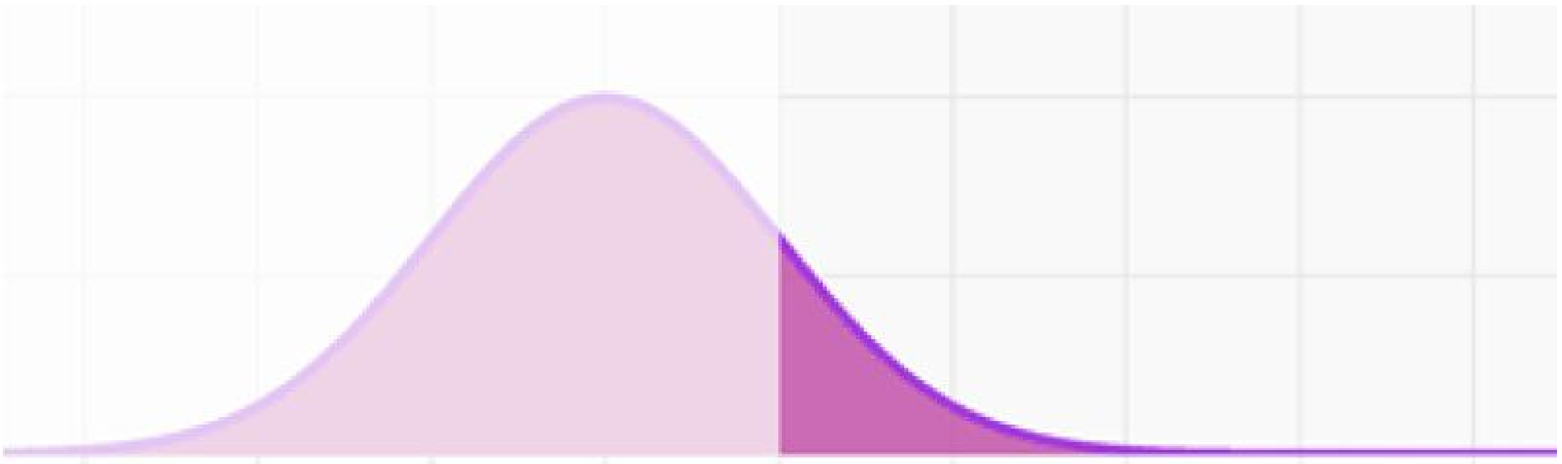
AUC optimization for binary classification

$$AUC(f_\theta) = \mathbb{P}\left(f_\theta(\mathbf{X}^+) > f_\theta(\mathbf{X}^-) | y^+ = 1, y^- = 0\right)$$

Distribution Insensitive

Maximizing its unbiased empirical estimation:

$$\hat{AUC}(h_\theta) = 1 - \underbrace{\frac{1}{n^+ n^-} \sum_{i=1}^{n^+} \sum_{j=1}^{n^-} \ell(h_\theta(\mathbf{X}^+) - h_\theta(\mathbf{X}^-))}_{\text{minimize}}$$



AUCSeg - AUC Optimization

AUC optimization for multi-class semantic segmentation

$$\ell_{auc} := \sum_{c=1}^{K} \sum_{c' \neq c} \underbrace{\sum_{\mathbf{X}_m^p \in \mathcal{N}_c} \sum_{\mathbf{X}_n^p \in \mathcal{N}_{c'}} \frac{1}{|\mathcal{N}_c||\mathcal{N}_{c'}|} \ell_{sq}^{c,c',m,n}}_{\text{binary AUC score}}$$

The proposed loss enjoys a well-guaranteed generalization bound

$$\left| \hat{\mathcal{L}}_{\mathcal{D}}(f) - \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{L}}_{\mathcal{D}}(f)\right] \right| \leq \boxed{\frac{8}{N} + \frac{\eta_{\text{inner}} + \eta_{\text{inter}}}{\sqrt{N}} \sqrt{A \log\left(2B\mu\tau N k + C\right)} + 3\left(\sqrt{\frac{1}{2N}} + K\sqrt{1 - \frac{1}{N}}\right)\sqrt{\log\left(\frac{4K(K-1)}{\delta}\right)}},$$

Number of classes

$$\Rightarrow \boxed{\tilde{\mathcal{O}}\left(\tau\sqrt{\log\left(\tau N k\right)/N}\right)}$$

Degree of imbalance

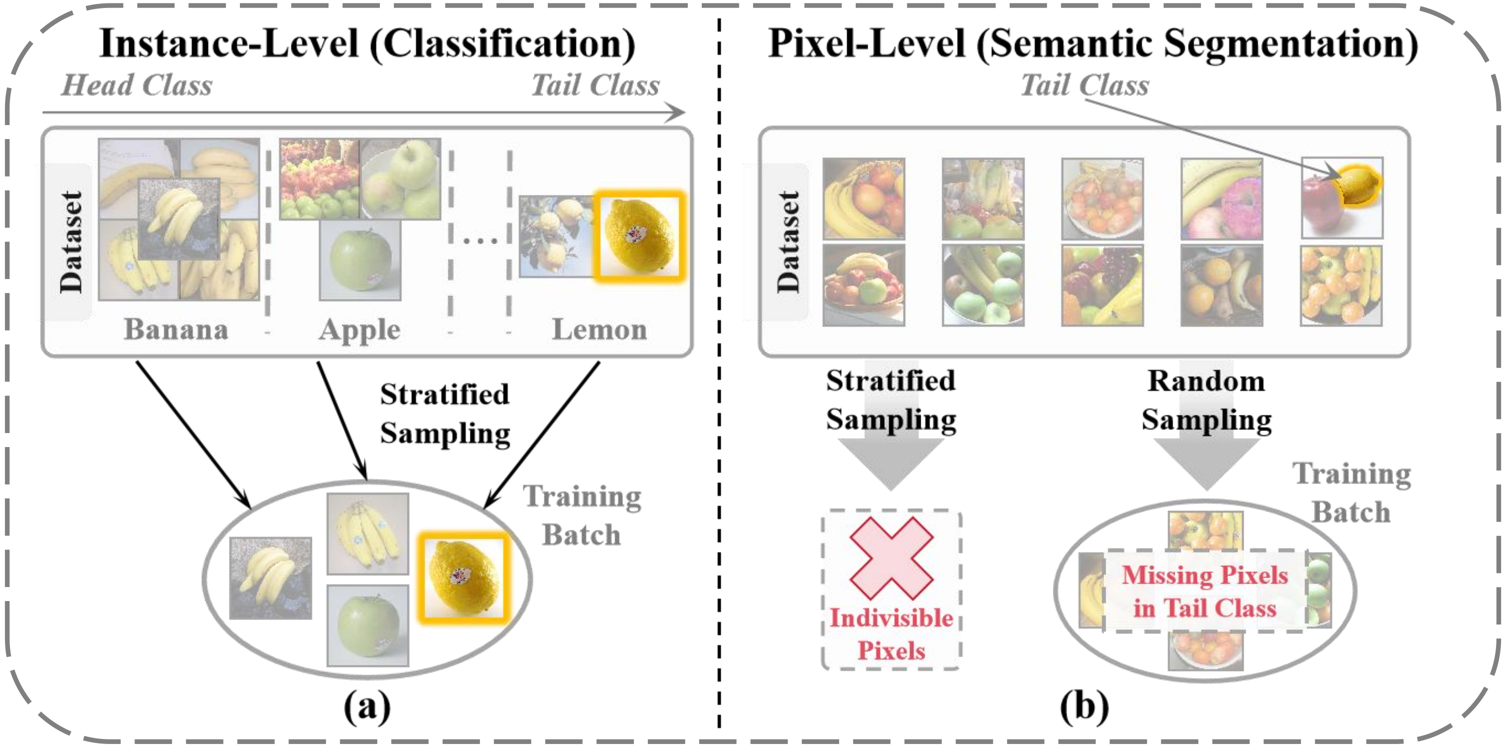
$$\eta_{\text{inner}} = \frac{48\mu\tau \ln N}{N}, \quad \eta_{\text{inter}} = 2\sqrt{2}\tau, \quad \tau = \left(\max_{c\in[K]} \frac{n_{\max}^{(c)}}{n_{\text{mean}}^{(c)}}\right)^2, \; n_{\max}^{(c)} = \max_{\mathbf{X}} n(\mathbf{X}^{(c)}), \; n_{mean}^{(c)} = \sum_{i=1}^{N} n(\mathbf{X}_i^{(c)})$$

AUCSeg - Tail-class Memory Bank

Two commonly used sampling methods:

- Stratified sampling
- Random sampling

Not suitable for this task



Cityscapes

$$K = 19$$

$$\delta = 0.01$$

$$\min_i p_i = 1\%$$

$$B = 759$$

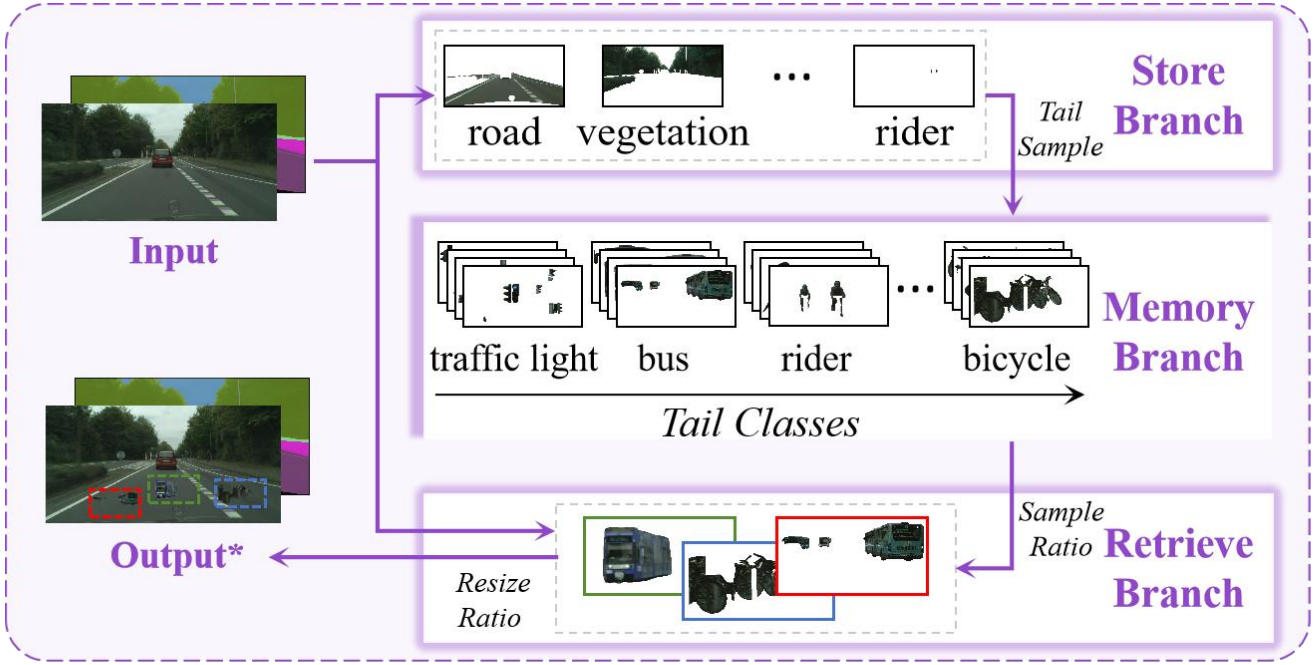
Proposition 1. *Consider a dataset $\mathcal{D}$ that includes images with K different pixel categories. Let p_i represent the probability of observing a pixel with label i in a given image. Randomly select B images from $\mathcal{D}$ as training data, where*

$$B = \Omega\left(\frac{\log(\delta/K)}{\log(1-\min_i p_i)}\right).$$

Then with probability at least $1-\delta$, for any $c \in [K]$, there exists $\mathbf{X}$ in the training data that contains pixels of label c.

AUCSeg - Tail-class Memory Bank

Tail-class Memory Bank identifies missing tail classes of all images involved in a mini-batch and randomly replaces some pixels in the image with missing classes based on stored historical class information in T-Memory Bank.



Previous Memory Bank

Storing the instance-level or image-level features

Tail-class Memory Bank

Storing the original pixels for each object

Results - Quantitative

Method	ADE20K Overall	ADE20K Head	ADE20K Middle	ADE20K Tail	Cityscapes Overall	Cityscapes Head	Cityscapes Middle	Cityscapes Tail	COCO-Stuff 164K Overall	COCO-Stuff 164K Head	COCO-Stuff 164K Middle	COCO-Stuff 164K Tail
DeepLabV3+	31.95	75.88	51.96	26.01	66.53	90.11	57.16	54.36	29.11	51.11	32.93	24.82
EncNet	32.12	75.34	51.60	26.32	71.34	91.62	60.76	63.03	27.31	49.89	30.41	23.09
FastFCN	29.78	74.20	49.44	23.86	63.97	90.37	52.43	51.22	28.37	50.60	32.52	23.96
EMANet	32.83	75.77	50.03	27.36	70.93	91.69	60.61	61.97	28.48	49.73	29.97	24.85
DANet	33.83	74.62	51.01	28.52	65.77	89.66	55.30	54.26	26.83	49.60	31.14	22.29
HRNet	31.83	75.35	49.98	26.19	73.40	91.98	65.79	64.00	28.65	48.00	30.74	25.16
OCRNet	29.64	74.00	49.40	23.72	66.95	90.24	63.18	50.21	28.67	51.04	32.41	24.33
DNLNet	33.24	75.90	51.16	27.69	70.68	91.98	59.90	61.66	30.23	50.71	33.05	26.41
PointRend	17.77	67.18	37.60	11.46	60.67	89.79	53.92	41.49	11.17	21.17	13.64	9.04
BiSeNetV2	10.26	60.38	28.72	4.10	73.04	92.00	63.52	64.93	10.30	34.96	12.71	5.92
ISANet	29.53	74.34	48.77	23.64	70.63	91.67	61.50	60.43	26.37	48.87	30.78	21.86
STDC	30.17	73.36	48.02	24.58	76.30	92.58	65.09	71.94	29.83	51.74	33.40	25.61
SegNeXt	47.45	80.54	**60.35**	43.28	82.41	**94.08**	72.46	80.92	42.42	**57.05**	41.71	40.33
VS	24.72	75.30	48.02	17.86	55.40	92.16	52.52	26.36	24.27	47.80	30.38	19.19
LA	31.16	77.07	53.43	24.77	62.75	92.98	64.79	35.09	28.56	49.67	33.16	24.21
LDAM	33.11	74.06	51.26	27.65	65.95	92.72	69.27	40.17	42.39	56.85	41.59	40.34
Focal Loss	47.68	80.54	59.04	43.73	82.44	93.90	**72.79**	80.89	41.98	56.87	41.51	39.79
DisAlign	48.15	80.33	59.14	44.31	81.94	93.61	72.12	80.36	42.10	55.20	41.24	40.28
BLV	46.76	79.96	58.96	42.67	81.81	93.84	71.83	80.05	42.17	56.83	41.52	40.06
AUCSeg (Ours)	**49.20**	**80.59**	59.45	**45.52**	**82.71**	93.91	72.72	**81.67**	**42.73**	56.95	**41.93**	**40.72**

Results - Quantitative

NEURAL INFORMATION PROCESSING SYSTEMS

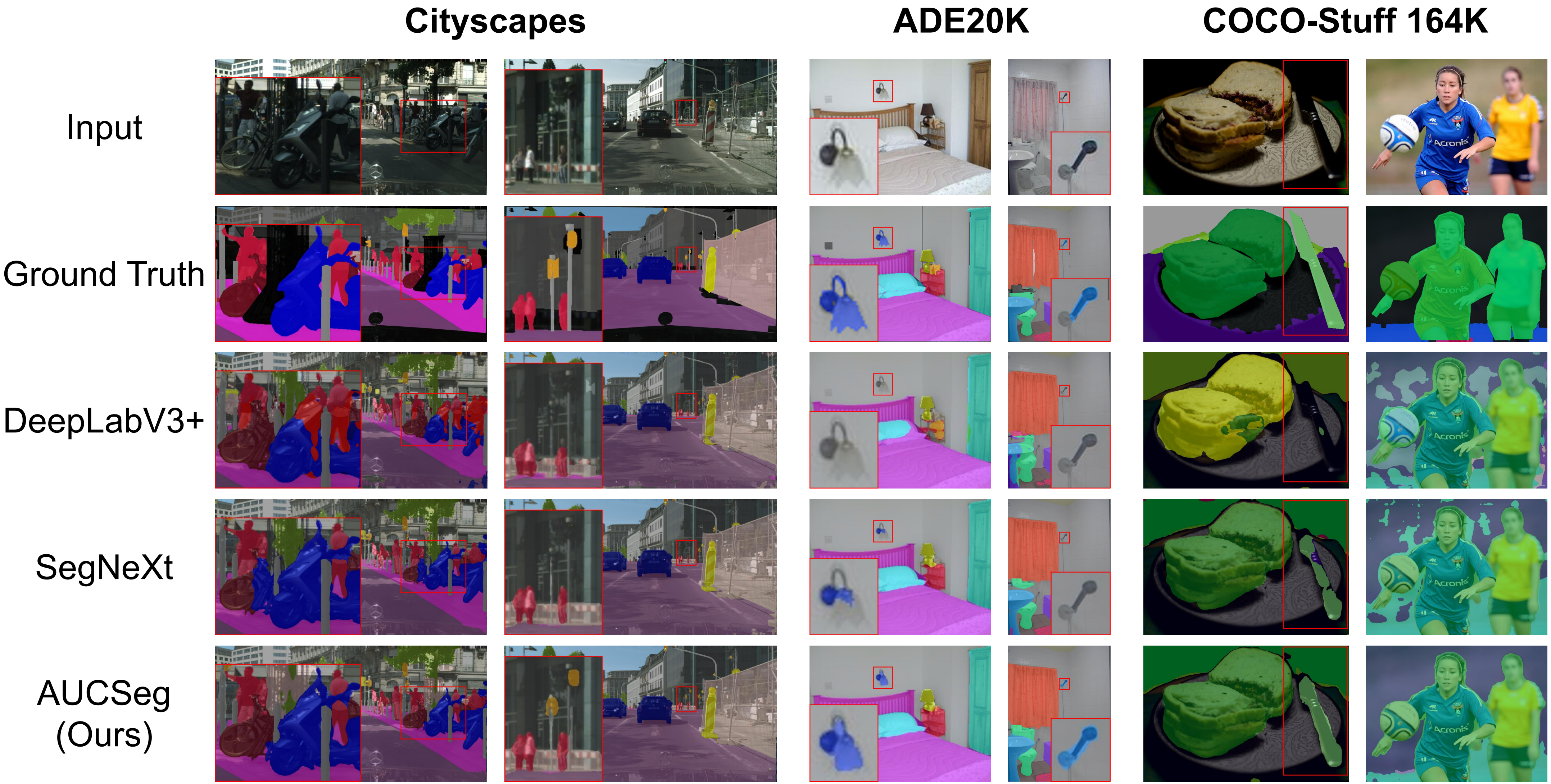
Backbone Extension:

Backbone	AUCSeg	Overall	Tail
DeepLabV3+	✗	31.95	26.01
	✓	**36.13**	**31.10**
EMANet	✗	32.83	27.36
	✓	**36.32**	**31.39**
OCRNet	✗	29.64	23.72
	✓	**34.82**	**29.75**
ISANet	✗	29.53	23.64
	✓	**35.07**	**30.13**

Backbone	AUCSeg	Overall	Tail
Tiny	✗	38.73	33.96
	✓	**39.00**	**34.52**
Small	✗	43.25	38.90
	✓	**43.29**	**39.18**
Base	✗	45.45	41.33
	✓	**46.37**	**42.49**
Large	✗	47.45	43.28
	✓	**49.20**	**45.52**

Results - Qualitative

NEURAL INFORMATION PROCESSING SYSTEMS



Conclusions

- ☐ **Methodologically**: propose a novel AUCSeg to address pixel-level long-tail semantic segmentation.
- ☐ **Theoretically**: demonstrate the generalization performance of AUCSeg in semantic segmentation.
- ☐ **Empirically**: comprehensive experiments justify the effectiveness of our method.

Paper | Code | Homepage

[1] MMSegmentaion: https://github.com/open-mmlab/mmsegmentation/tree/v0.24.1

[2] XCurve: https://github.com/statusrank/XCurve