OVS Meets Continual Learning: Towards Sustainable Open-Vocabulary Segmentation

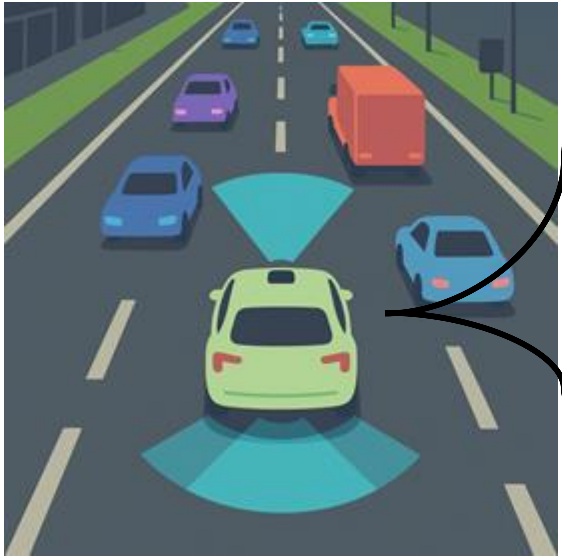
Dongjun Hwang[1], Yejin Kim[1], Minyoung Lee[1], Seong Joon Oh[2,3], Junsuk Choe[1]

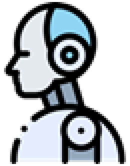
[1]Sogang University, [2]University of Tübingen, [3]Tübingen AI Center

NeurIPS 2025

Background: **Open-Vocabulary Segmentation (OVS)**

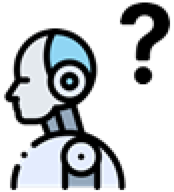
Example: **Autonomous Driving**





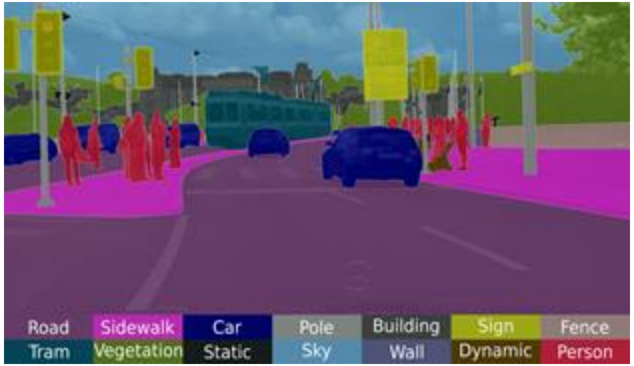
Predict

AI Model



Predict

1) Classes included in the training set

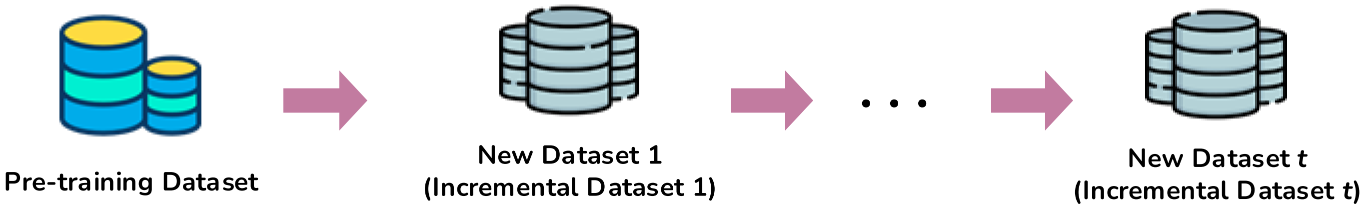


2) Classes not included in the training set

Goal of OVS: To recognize classes that are not included in the training set.

Motivation: **Limitation of OVS**

Most previous OVS studies have assumed **a single training scenario using a pre-training dataset.**



However, in practice, new datasets are often collected and **become available sequentially over time.**

This raises a natural question:
How should we handle such new datasets? Should we train on them?

Motivation: **Limitation of OVS**

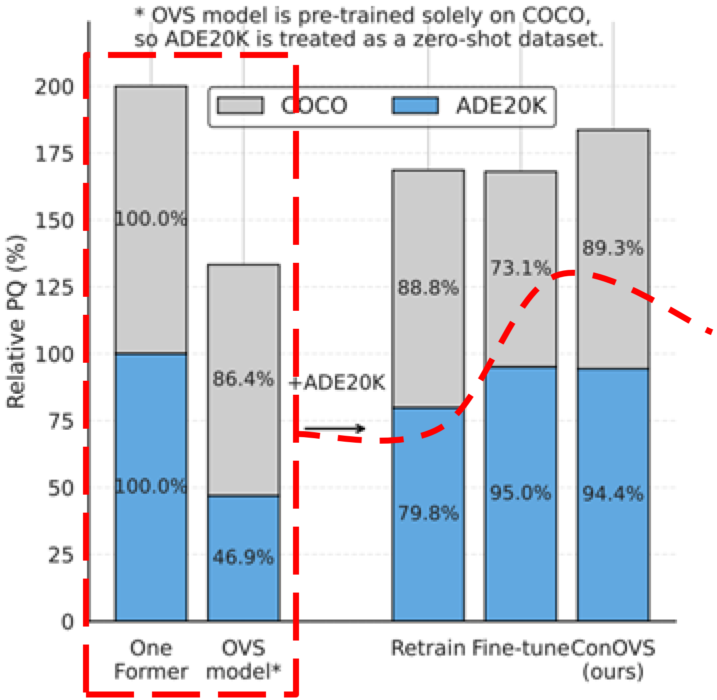
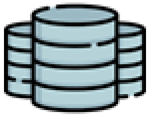


Figure: (a) Comparison of the performance of the OVS model, Retraining, Fine-tuning, and ConOVS against the closed-set segmentation model OneFormer.

Current OVS models still **do not achieve sufficient performance on unseen classes to be used as is** when new datasets become available.

In other words, **their generalization ability is not yet strong enough** to eliminate the need for further training.

In fact, they often underperform compared to closed-set segmentation models fine-tuned on specific datasets.



Incremental Dataset

Therefore, when new data becomes available, the model needs to be trained on it to expand its recognition capability.

Goal: Effectively learn new information while preserving previously acquired knowledge.

Then, how can we expand the OVS model's recognition ability using new (incremental) datasets?

Motivation: **Limitation of OVS**

COCO: Classes included in the training set
ADE20K: Classes not included in the training set

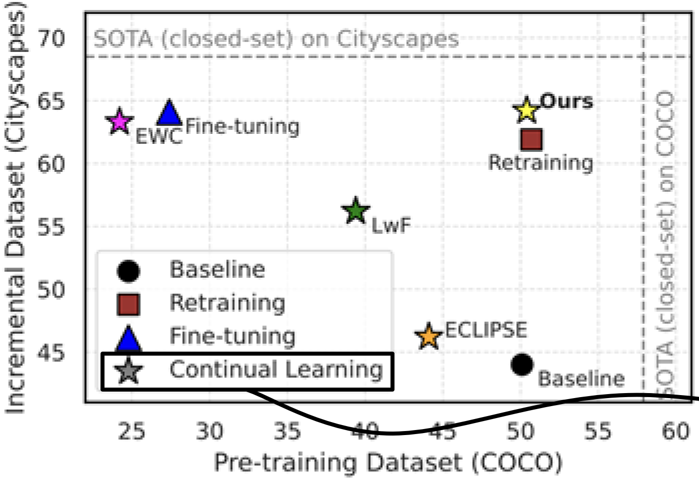
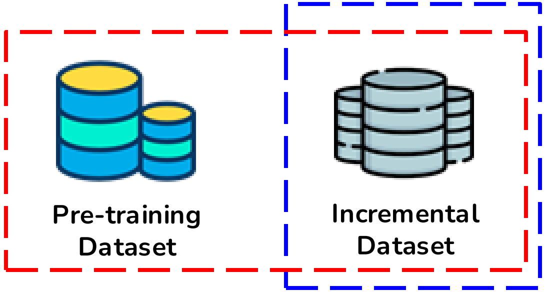


Figure: **Performance of each method.**

Retraining, fine-tuning, and existing continual learning methods are not effective solutions.

Retraining: Requires large computational resources.



Fine-tuning: Causes forgetting of previously learned knowledge.

Continual Learning: Often results in suboptimal performance.

Methodology: **ConOVS**

Training Phase

During training, we derive expert models and multivariate normal (MVN) distributions for each dataset.

- 1) We first train an OVS model from scratch using the pre-training dataset.
- 2.1) Then, we fine-tune only the decoder on each incremental dataset to obtain an expert model specific to that dataset.
- 2.2) For each dataset, we also compute the mean and covariance matrix of the image and text embeddings, which define the **MVN distributions**.

Methodology: **ConOVS**

Inference Phase

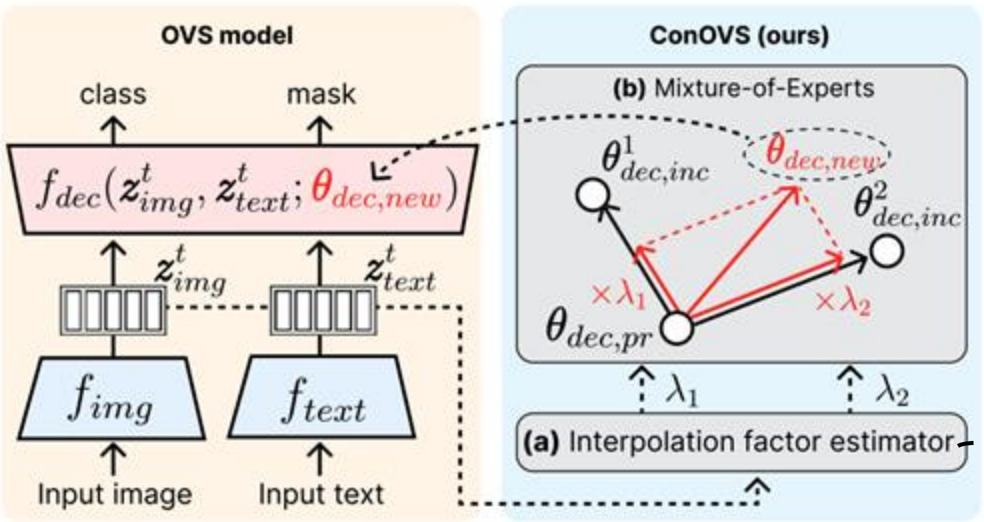


Figure: **Overview of the inference process of our proposed method.**

Algorithm 1 Interpolation factor estimator

Require: Input $(\boldsymbol{x}_{\text{img}}, \boldsymbol{x}_{\text{text}})$, encoders $f_{\text{img}}, f_{\text{text}}$, decoder f_{dec}; MVN parameters $\{\boldsymbol{\Phi}^i_{\text{img}}, \boldsymbol{\Phi}^i_{\text{text}}\}^n_{i=0}$; PDF $p(\cdot|\boldsymbol{\Phi})$

Ensure: Interpolation factor $\boldsymbol{\lambda}$

1: Extract embeddings: $\boldsymbol{z}_{\text{img}} \leftarrow f_{\text{img}}(\boldsymbol{x}_{\text{img}})$, $\boldsymbol{z}_{\text{text}} \leftarrow f_{\text{text}}(\boldsymbol{x}_{\text{text}})$

2: Estimate likelihoods: $\boldsymbol{l}_{\text{img}} \leftarrow \{p(\boldsymbol{z}_{\text{img}} \mid \boldsymbol{\Phi}^i_{\text{img}})\}$, $\boldsymbol{l}_{\text{text}} \leftarrow \{p(\boldsymbol{z}_{\text{text}} \mid \boldsymbol{\Phi}^i_{\text{text}})\}$

3: Compute: $\boldsymbol{p}_{\text{img}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{img}})$, $\boldsymbol{p}_{\text{text}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{text}})$

4: Combine: $\boldsymbol{\lambda} \leftarrow \max(\boldsymbol{p}_{\text{img}}, \boldsymbol{p}_{\text{text}})$

5: **return** $\boldsymbol{\lambda}$

Methodology: **ConOVS**

Inference Phase

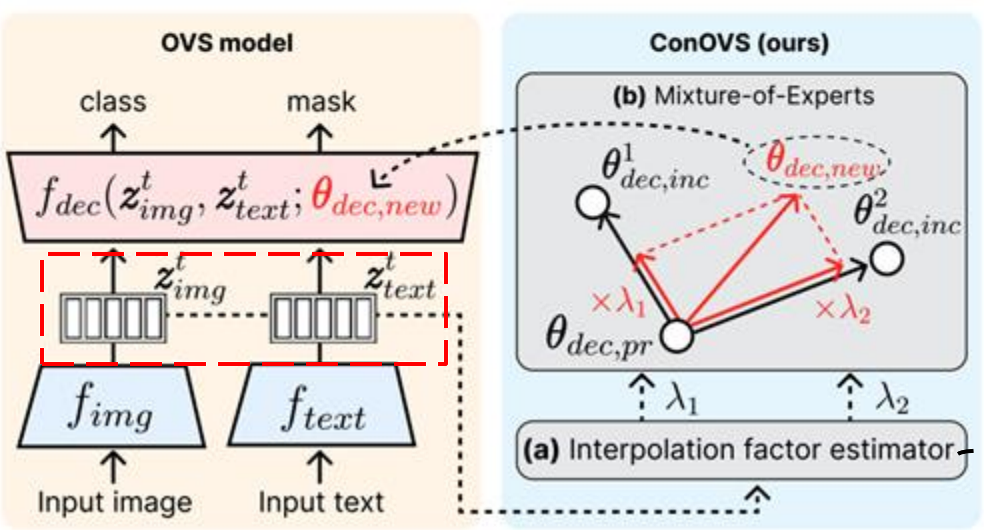


Figure: **Overview of the inference process of our proposed method.**

Algorithm 1 Interpolation factor estimator

Require: Input $(\boldsymbol{x}_{\text{img}}, \boldsymbol{x}_{\text{text}})$, encoders $f_{\text{img}}, f_{\text{text}}$, decoder f_{dec}; MVN parameters $\{\boldsymbol{\Phi}^i_{\text{img}}, \boldsymbol{\Phi}^i_{\text{text}}\}^n_{i=0}$; PDF $p(\cdot|\boldsymbol{\Phi})$

Ensure: Interpolation factor $\boldsymbol{\lambda}$

1: Extract embeddings: $\boldsymbol{z}_{\text{img}} \leftarrow f_{\text{img}}(\boldsymbol{x}_{\text{img}})$, $\boldsymbol{z}_{\text{text}} \leftarrow f_{\text{text}}(\boldsymbol{x}_{\text{text}})$

2: Estimate likelihoods: $\boldsymbol{l}_{\text{img}} \leftarrow \{p(\boldsymbol{z}_{\text{img}} \mid \boldsymbol{\Phi}^i_{\text{img}})\}$, $\boldsymbol{l}_{\text{text}} \leftarrow \{p(\boldsymbol{z}_{\text{text}} \mid \boldsymbol{\Phi}^i_{\text{text}})\}$

3: Compute: $\boldsymbol{p}_{\text{img}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{img}})$, $\boldsymbol{p}_{\text{text}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{text}})$

4: Combine: $\boldsymbol{\lambda} \leftarrow \max(\boldsymbol{p}_{\text{img}}, \boldsymbol{p}_{\text{text}})$

5: **return** $\boldsymbol{\lambda}$

Methodology: **ConOVS**

Inference Phase

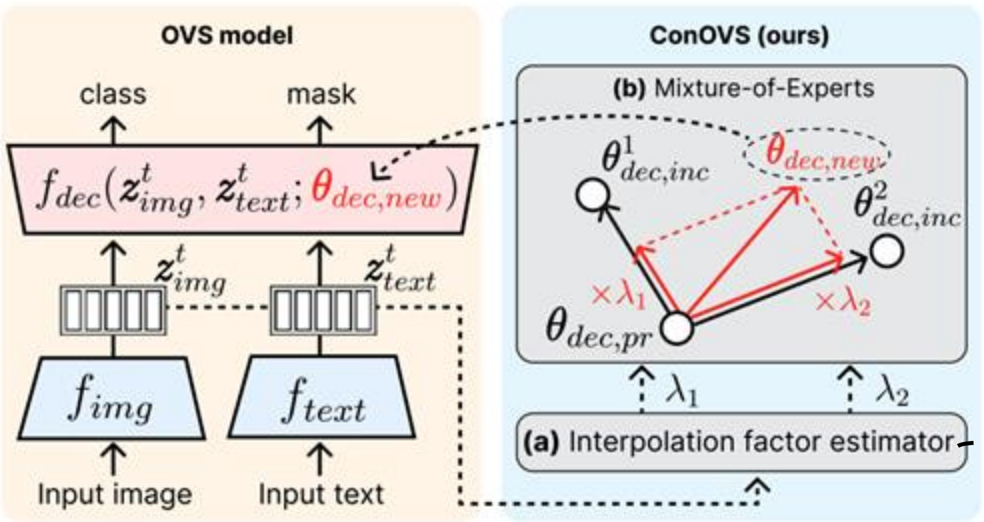


Figure: **Overview of the inference process of our proposed method.**

Algorithm 1 Interpolation factor estimator

Require: Input $(\boldsymbol{x}_{\text{img}}, \boldsymbol{x}_{\text{text}})$, encoders $f_{\text{img}}, f_{\text{text}}$, decoder f_{dec}; MVN parameters $\{\boldsymbol{\Phi}^i_{\text{img}}, \boldsymbol{\Phi}^i_{\text{text}}\}^n_{i=0}$; PDF $p(\cdot|\boldsymbol{\Phi})$

Ensure: Interpolation factor $\boldsymbol{\lambda}$

1: Extract embeddings: $\boldsymbol{z}_{\text{img}} \leftarrow f_{\text{img}}(\boldsymbol{x}_{\text{img}})$, $\boldsymbol{z}_{\text{text}} \leftarrow f_{\text{text}}(\boldsymbol{x}_{\text{text}})$

2: Estimate likelihoods: $\boldsymbol{l}_{\text{img}} \leftarrow \{p(\boldsymbol{z}_{\text{img}} \mid \boldsymbol{\Phi}^i_{\text{img}})\}$, $\boldsymbol{l}_{\text{text}} \leftarrow \{p(\boldsymbol{z}_{\text{text}} \mid \boldsymbol{\Phi}^i_{\text{text}})\}$

3: Compute: $\boldsymbol{p}_{\text{img}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{img}})$, $\boldsymbol{p}_{\text{text}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{text}})$

4: Combine: $\boldsymbol{\lambda} \leftarrow \max(\boldsymbol{p}_{\text{img}}, \boldsymbol{p}_{\text{text}})$

5: **return** $\boldsymbol{\lambda}$

Methodology: **ConOVS**

Inference Phase

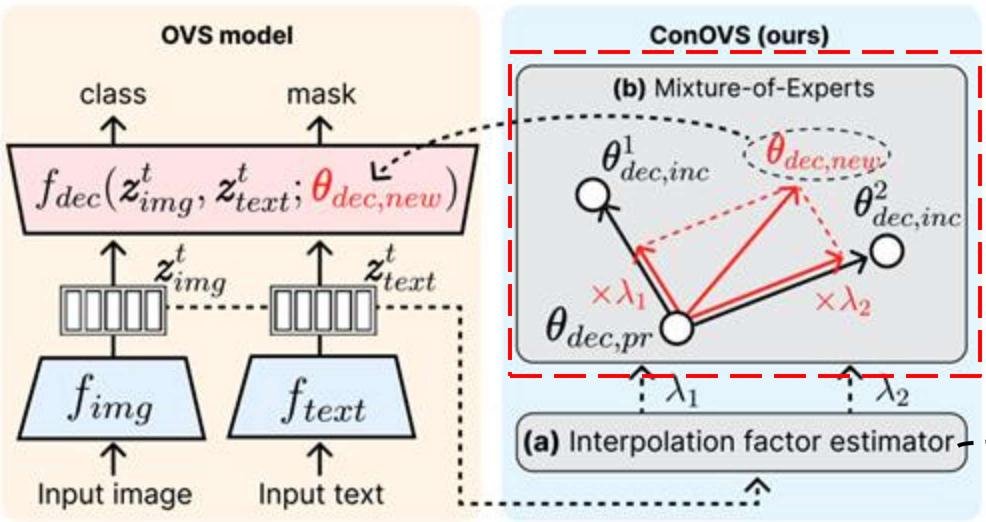


Figure: **Overview of the inference process of our proposed method.**

Algorithm 1 Interpolation factor estimator

Require: Input $(\boldsymbol{x}_{\text{img}}, \boldsymbol{x}_{\text{text}})$, encoders $f_{\text{img}}, f_{\text{text}}$, decoder f_{dec}; MVN parameters $\{\boldsymbol{\Phi}^i_{\text{img}}, \boldsymbol{\Phi}^i_{\text{text}}\}^n_{i=0}$; PDF $p(\cdot|\boldsymbol{\Phi})$

Ensure: Interpolation factor $\boldsymbol{\lambda}$

1: Extract embeddings: $\boldsymbol{z}_{\text{img}} \leftarrow f_{\text{img}}(\boldsymbol{x}_{\text{img}})$, $\boldsymbol{z}_{\text{text}} \leftarrow f_{\text{text}}(\boldsymbol{x}_{\text{text}})$

2: Estimate likelihoods: $\boldsymbol{l}_{\text{img}} \leftarrow \{p(\boldsymbol{z}_{\text{img}} \mid \boldsymbol{\Phi}^i_{\text{img}})\}$, $\boldsymbol{l}_{\text{text}} \leftarrow \{p(\boldsymbol{z}_{\text{text}} \mid \boldsymbol{\Phi}^i_{\text{text}})\}$

3: Compute: $\boldsymbol{p}_{\text{img}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{img}})$, $\boldsymbol{p}_{\text{text}} \leftarrow \text{softmax}(\boldsymbol{l}_{\text{text}})$

4: Combine: $\boldsymbol{\lambda} \leftarrow \max(\boldsymbol{p}_{\text{img}}, \boldsymbol{p}_{\text{text}})$

5: **return** $\boldsymbol{\lambda}$

Experiments: **Settings**

Type of Learning Sequence	Pre-training Dataset	Incremental Dataset	Zero-shot test Dataset
(**S1**) Scenario 1	COCO	Cityscapes	ADE20K
(**S2**) Scenario 2	COCO	ADE20K	Cityscapes
(**S3**) Scenario 3	COCO	Cityscapes, ADE20K	LVIS, BDD100K, Mapillary Vistas, PC-59, PC-459, PAS-20, PAS-21, A-847
(**S4**) Scenario 4	COCO	Cityscapes, ADE20K, BDD100K, Mapillary Vistas	LVIS, PC-59, PC-459, PAS-20, PAS-21, A-847

- **Learning Sequence:** We define four experimental scenarios (**S1, S2, S3, S4**) based on different learning sequences of the datasets.

- **Comparisons**: Retraining, Fine-tuning, ER (Experience Replay), LwF (Learning without Forgetting), EWC (Elastic Weight Consolidation), ECLIPSE (Continual learning method for the closed-set segmentation).

- **Evaluation Metrics**: We evaluate panoptic, instance, and semantic segmentation using PQ, mAP, and mIoU, respectively. Due to space constraints, we report only PQ in the main paper.

Experiments: **Scenario 1, 2, 3**

Scenario 1, 2: One Incremental Dataset

In scenarios S1 and S2, **our method consistently outperforms existing approaches across all datasets**, whether the incremental dataset is ADE20K or Cityscapes

Table: Comparison of performance when the incremental dataset is (left) Cityscapes (right) ADE20K.

Method	CL	COCO (pre-training)	Cityscapes (incremental)	ADE20K (zero-shot)
fc-clip	✗	50.1	44.0	23.5
Fine-tuning	✗	-22.7	+20.1	-10.3
Retraining	✗	+0.6	+17.9	+1.7
ER	✓	-1.6	+19.0	+0.3
LwF	✓	-10.7	+12.2	-0.8
EWC	✓	-25.9	+19.3	-9.8
ECLIPSE	✓	-6.0	+2.2	+0.9
ConOVS (ours)	✓	+0.3	**+20.2**	**+2.5**
X-Decoder	✗	56.7	36.3	16.7
Fine-tuning	✗	-50.4	+26.6	-12.9
ConOVS (ours)	✓	-0.4	+26.6	+0.1

Method	CL	COCO (pre-training)	ADE20K (incremental)	Cityscapes (zero-shot)
fc-clip	✗	50.1	23.5	44.0
Fine-tuning	✗	-7.7	**+24.1**	-3.0
Retraining	✗	+1.4	+16.5	-1.2
ER	✓	+0.4	+21.5	-3.5
LwF	✓	-3.8	+13.7	-1.0
EWC	✓	-11.1	+20.7	-2.6
ECLIPSE	✓	-0.5	+0.2	-5.9
ConOVS (ours)	✓	**+1.7**	+23.8	**+0.9**
X-Decoder	✗	56.7	16.7	36.3
Fine-tuning	✗	-37.3	+28.2	-3.7
ConOVS (ours)	✓	-1.5	**+29.2**	**+1.4**

Scenario 3: Two Incremental Datasets

In scenario S3, **our method consistently achieves superior performance** compared to both fine-tuning and retraining.

Table: Performance comparison in scenario S3.

Method	Learning Sequence	COCO (pre-training)	ADE20K (incremental)	Cityscapes (incremental)
fc-clip	-	50.1	23.5	44.0
Fine-tuning	ADE → City	20.8	15.4	65.2
Fine-tuning	City → ADE	39.3	48.3	46.0
Retraining	COCO, City, ADE	48.6	35.5	60.5
ConOVS (ours)	City, ADE	**51.6**	**47.0**	**64.3**

Experiments: **Scenario 3**

Scenario 3: Two Incremental Datasets

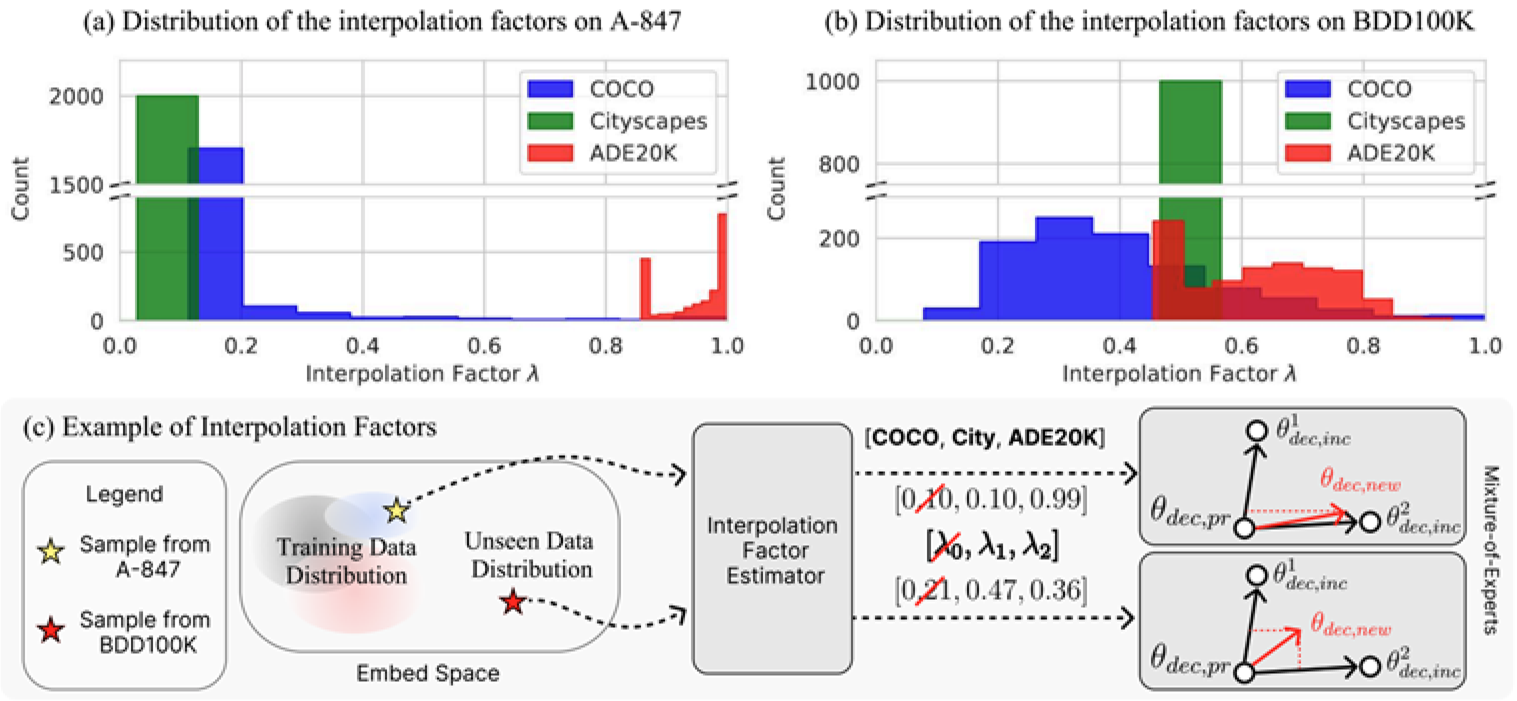
Method	Learning Sequence	LVIS (mAP)	BDD100K (PQ)	Mapillary (mIoU)	PC-59 (mIoU)	PC-459 (mIoU)	PAS-20 (mIoU)	PAS-21 (mIoU)	A-847 (mIoU)
fc-clip	-	20.5	19.0	26.0	53.0	16.9	93.1	80.2	13.8
Fine-tuning	City → ADE	21.7	19.7	27.8	52.1	17.2	92.3	76.7	16.0
Fine-tuning	ADE → City	10.4	21.3	24.2	45.9	13.5	87.4	70.7	11.5
Retraining	COCO, City, ADE	21.5	21.8	28.0	53.2	17.3	93.3	80.9	15.2
ConOVS (ours)	City, ADE	**23.1**	**22.6**	**29.1**	**54.9**	**17.9**	**93.6**	**80.7**	**16.3**

In addition, our method also consistently **outperforms other approaches in various zero-shot evaluations.**

As shown in the above table, it achieves superior performance across all eight zero-shot test datasets.

Experiments: **Analysis**

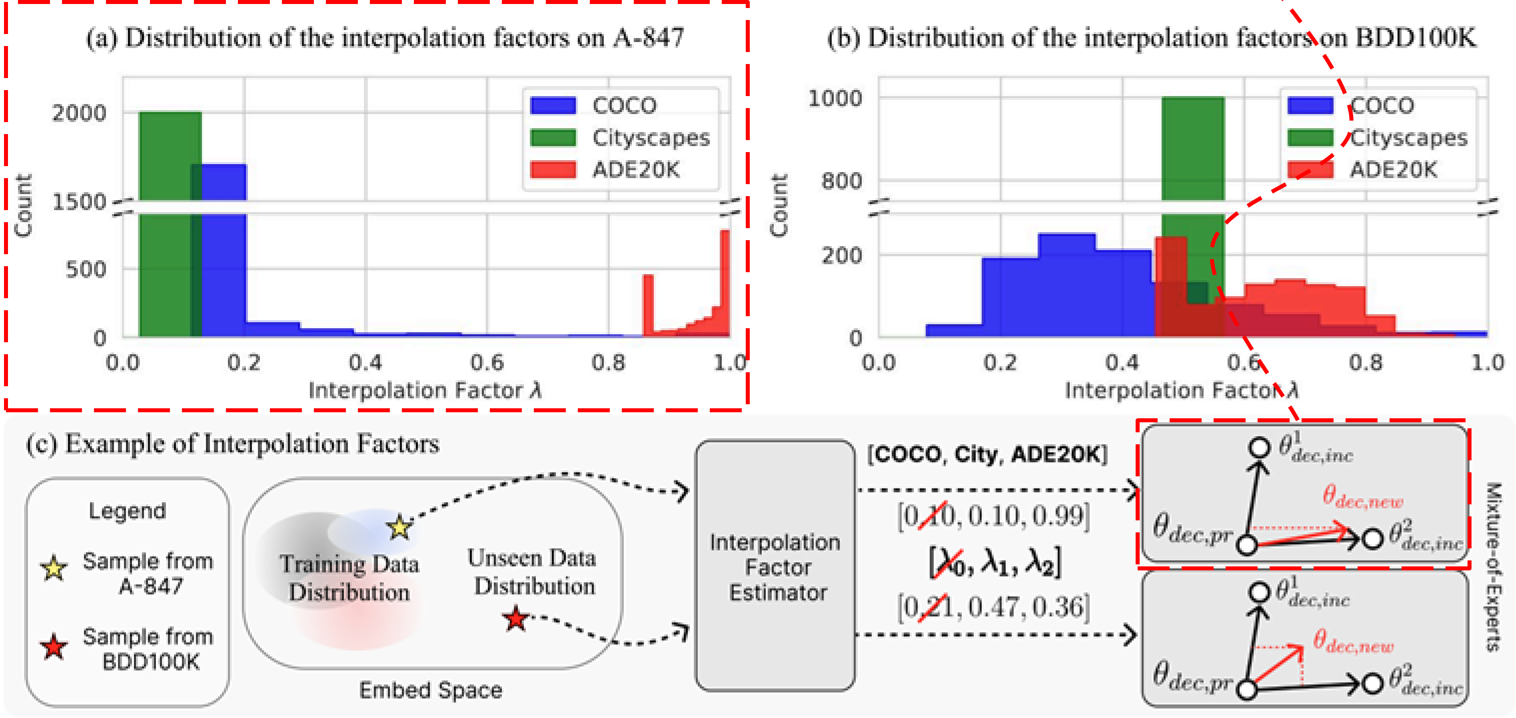
Understanding the Behavior of the Interpolation Factor.



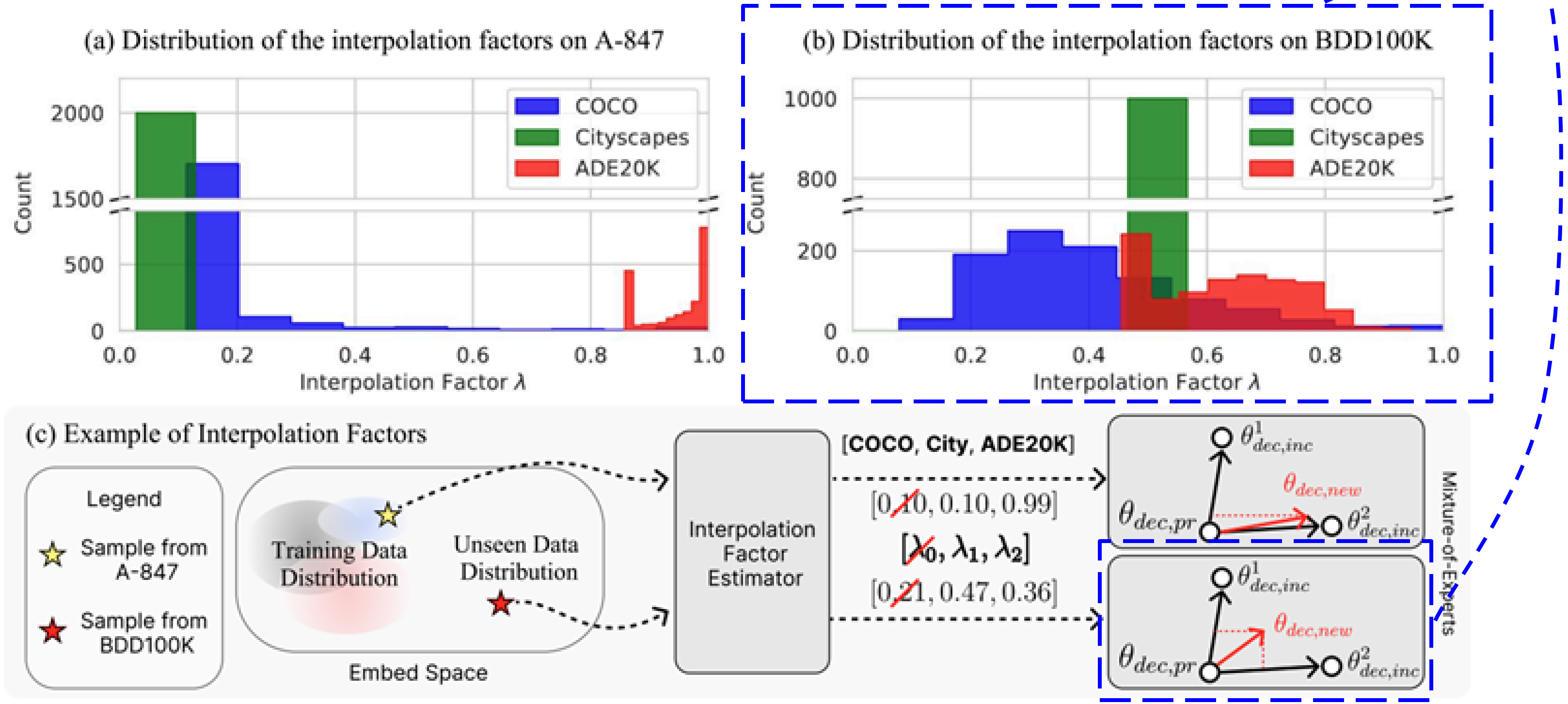
Experiments: **Analysis**

Understanding the Behavior of t

When input samples are similar to a previously trained distribution, our method selectively activates the corresponding expert to maximize performance.



In such cases, our method disperses the λ values to avoid over-reliance on a single expert. Instead, it combines the weights of multiple experts based on the probability that the input sample belongs to each distribution.



Conclusion

- We identify that existing Open-Vocabulary Segmentation (OVS) methods perform poorly on unseen data, a limitation overlooked by prior work.

- To address this, we define a new setting where **OVS models are incrementally trained with new datasets.**

- We find that retraining, fine-tuning, and continual learning are **inefficient or ineffective under this setting.**

- We propose **ConOVS, an MoE-based continual learning method** that dynamically merges expert decoders by estimating the dataset distribution of each input, and we validate its effectiveness through extensive evaluations across diverse sequential learning settings.

- **Future work**: We can expand this work to Open-Vocabulary object Detection (OVD) as ConOVS fine-tune the decoder and usually OVD utilize the encoder-decoder framwork.