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

Abstract

Open-Vocabulary Segmentation (OVS) aims to segment classes that are not present in the training dataset. However, most existing studies assume that the training data is fixed in advance, overlooking more practical scenarios where new datasets are continuously collected over time. To address this, we first analyze how existing OVS models perform under such conditions. In this context, we explore several approaches such as retraining, fine-tuning, and continual learning but find that each of them has clear limitations. To address these issues, we propose ConOVS, a novel continual learning method based on a Mixture-of-Experts framework. ConOVS dynamically combines expert decoders based on the probability that an input sample belongs to the distribution of each incremental dataset. Through extensive experiments, we show that ConOVS consistently outperforms existing methods across pre-training, incremental, and zero-shot test datasets, effectively expanding the recognition capabilities of OVS models when data is collected sequentially. Code is available at: https://github.com/dongjunhwang/ConOVS

1 Introduction

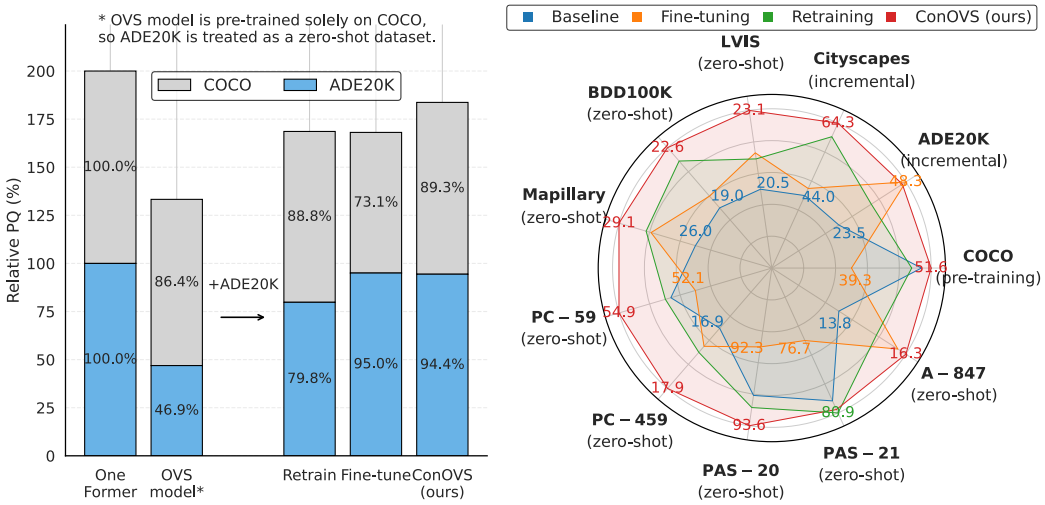
In fields such as robotics [9] and autonomous driving [24, 47], there is a growing demand for models that can segment novel objects not included in the training dataset. However, conventional closed-set segmentation models, which are restricted to recognizing only the classes seen during training, fall short in meeting this demand. To address this limitation, Open-Vocabulary Segmentation (OVS) has emerged, aiming to enable segmentation of unseen classes that are not included in the training dataset. OVS continues to be an active area of research, particularly through methods that leverage foundation models such as CLIP [52, 58].

Most previous studies [49, 52, 55, 59] on OVS assume a scenario in which the model is trained once using a pre-training dataset. However, in practice, trainable datasets often arrive sequentially as new data are collected over time. Considering this setting, we first discuss how existing OVS models perform under such conditions. To facilitate a clearer discussion, we measure the relative performance of OVS models on seen and unseen classes using a *reference baseline*. We adopt OneFormer [18] for this role. It represents the state-of-the-art in closed-set segmentation and shares the same ConvNeXt backbone [27] as the OVS model [52], enabling a fair comparison in model capacity.

The most straightforward approach is to use the existing OVS model as is. In our experiments (Figure 1a), the existing OVS model achieves 86.4% of OneFormer's performance on the pre-training dataset COCO [26]. In contrast, its performance drops to 46.9% on a new dataset ADE20K [57], which contains unseen classes as well. These results indicate that the OVS model fails to perform well on datasets it has not encountered during training.

† Corresponding author (jschoe@sogang.ac.kr).

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) (b)

Figure 1: (a) Comparison of the performance of the OVS model (fc-clip [52]), Retraining, Fine-tuning, and ConOVS against the closed-set segmentation model OneFormer. (b) Performance of the Baseline (fc-clip [52]), Fine-tuning, Retraining, and ConOVS on the pre-training, incremental, and zero-shot test datasets. PQ is used.

To determine whether this performance gap is due to the inherent difficulty of the unseen classes or simply because the model has not been trained on them, we retrain the OVS model using both the pre-training dataset and the new dataset. As shown in Figure 1a, the model's performance on ADE20K improves significantly from 46.9% to 79.8% relative to OneFormer. This result confirms that the low performance on the new dataset is primarily due to the lack of exposure during training. It also suggests that this limitation of the OVS model can be effectively mitigated by training on newly collected data.

However, retraining the model from scratch demands substantial computational resources. In particular, this approach becomes impractical when the pre-training data is no longer accessible or computational resources are limited. To address these limitations, we consider an alternative approach: transfer learning. Specifically, we fine-tune the pre-trained OVS model on a new dataset. However, as shown in Figure 1a and 1b, this approach also has a limitation. It leads to performance degradation not only on the pre-training dataset but also on zero-shot tasks. This issue appears to stem from a well-known drawback of fine-tuning, namely, *catastrophic forgetting*. Therefore, we consider continual learning (CL) methods, which are designed to address catastrophic forgetting. However, most CL approaches are developed under the assumption that the number of classes is finite, making them unsuitable for open-vocabulary tasks where the number of classes can be potentially infinite [60, 61].

As a result, in scenarios where new datasets are continuously collected—as assumed in this paper—it is still unclear how to effectively utilize the incoming data, and finding a viable solution in OVS remains a non-trivial challenge. To address this, we propose ConOVS, a Mixture-of-Experts (MoE) based continual learning method that incrementally trains an existing OVS model on new datasets. Our method begins by fine-tuning the pre-trained OVS model to build a distinct expert for each new dataset. During inference, we estimate the probability that a given input sample is close to the distribution of each training dataset, based on their statistical representations. The model then computes an interpolation factor from these probabilities and dynamically combines the experts by interpolating their weights. This allows our method to produce an optimal model for predicting each input sample.

To simulate the scenario assumed in this paper, we sequentially introduce incremental datasets to an existing OVS model and evaluate the resulting models on three validation sets: the pre-training dataset, the incremental dataset, and zero-shot datasets. As shown in Figure 1b, our method not only significantly improves performance on the incremental dataset compared to standard retraining and fine-tuning, but also consistently enhances performance on both the pre-training and zero-shot datasets. Furthermore, compared to existing continual learning methods, our approach achieves superior performance across all three evaluation settings.

2 Related Works

2.1 Open-Vocabulary Segmentation

Recent open-vocabulary segmentation (OVS) research has focused on leveraging models capable of open-vocabulary classification, such as CLIP [36], to recognize classes that are not included in the training dataset. For example, fc-clip [52] identifies unseen classes by combining class embeddings from the model's decoder with those from CLIP. Moreover, a recent study [35] has explored an approach that retrieves LoRA modules trained on different datasets according to the input and utilizes them in conjunction with CLIP. Other methods further enhance the recognition of unseen classes by either applying visual grounding techniques like GradCAM [40] to CLIP [29, 42, 58] or distilling knowledge from both CLIP and the segmentation foundation model Segment Anything Model (SAM) [43, 54]. Meanwhile, there are also OVS approaches that do not rely on CLIP. For instance, methods such as X-Decoder [55, 62, 63] train both the encoder and decoder from scratch using segmentation datasets along with large-scale image–text pair datasets.

Most existing OVS studies are based on a scenario in which the model is trained only once. However, this setting inherently limits performance on unseen classes (see Section 1). To overcome this limitation, we analyze strategies for training OVS models in a scenario where new datasets are introduced sequentially.

2.2 Continual Learning

Acquiring additional knowledge in an already trained model is not straightforward. When a model is further trained on new data, it often tends to forget previously learned information while learning the new content [30]. This phenomenon is widely known as *catastrophic forgetting*. To address this issue, the field of continual learning (CL) has emerged. CL explores methods that enable models to learn from new data while retaining prior knowledge.

CL techniques are typically categorized into three types. First, replay-based methods store a subset of previously seen data and retrain the model using it to preserve prior knowledge [2, 38]. Second, regularization-based methods introduce penalty terms in the loss function to constrain parameter updates, preventing significant deviations during training on a new dataset [1, 21, 25]. Third, parameter-isolation-based methods mitigate interference by freezing previously learned parameters and allocating separate parameters for learning new data [20, 46]. Several approaches extend this idea into a Mixture-of-Experts (MoE) framework, where additional parameter sets are treated as distinct experts, and a gating module selects the appropriate expert based on the input [22, 45].

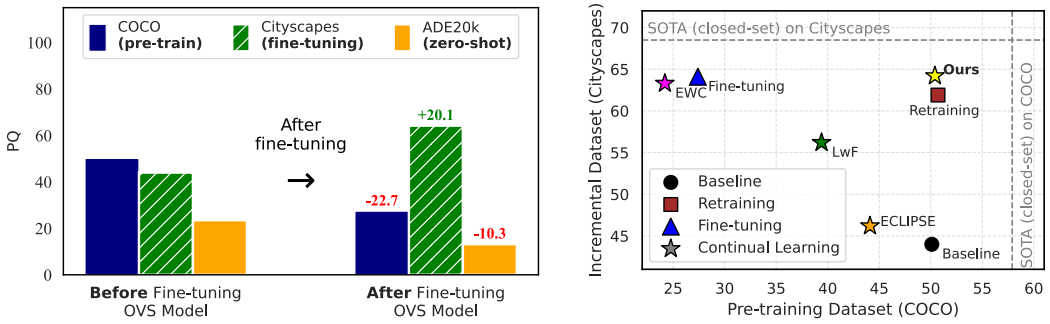
However, existing CL methods are designed under the assumption that the number of classes is finite, which limits their applicability in open-vocabulary settings [56, 60, 61]. Therefore, there is a need for novel approaches that enable continual learning in Open-Vocabulary Segmentation (OVS) scenarios, where new data are introduced incrementally. To address this, we propose a novel MoE-based continual learning technique that effectively expands the capacity of OVS models.

3 Motivation

In this section, we expand on the discussion from Section 1 and explore in greater detail how newly collected datasets can be leveraged to improve the performance of OVS models.

The most straightforward approach is to retrain the model from scratch using a joint dataset that combines the original and newly collected data. In practice, this strategy effectively preserves performance on seen classes while substantially improving performance on unseen classes. However, it suffers from two major limitations: (1) it incurs significant computational costs, as the model must be retrained from scratch every time new data are added; and (2) retraining becomes entirely infeasible if access to the original dataset has expired.

Due to these limitations, fine-tuning the model using only the newly collected dataset may appear to be a practical alternative. However, this approach compromises the model's original performance. As shown in Figure 2a, fine-tuning the OVS model results in a significant drop in performance not only on the pre-training dataset but also on the zero-shot test dataset. Qualitative examples provided in the



(a) (b)

Figure 2: (a) Performance degradation on the pre-training and zero-shot datasets after fine-tuning. fc-clip is used. (b) Comparison of the performance of OneFormer [18], the baseline (fc-clip [52]), retraining, fine-tuning, three existing continual learning methods [20, 21, 25], and ConOVS on the pre-training and incremental datasets. All methods use the same iterations. PQ is used.

Appendix I further illustrate this phenomenon. This degradation is likely caused by a well-known issue in fine-tuning, known as catastrophic forgetting [21, 25].

Another potential direction is to apply continual learning (CL) methods to OVS models. However, most existing CL methods are built on the assumption of a finite set of classes, making them difficult to directly apply to open-vocabulary tasks [56, 60, 61]. For instance, [1, 20] apply CL to segmentation tasks by treating all unseen classes as background, which fundamentally conflicts with the goal of OVS models that aim to recognize potentially unlimited categories.

Even when existing CL methods are adapted for OVS (see Appendix A.2 for implementation details), our experimental results show that their effectiveness is limited. As shown in Figure 2b, OVS models trained with adapted CL methods perform significantly worse than the closed-set segmentation model OneFormer on both the pre-training and incremental datasets. We believe this arises because existing CL methods assume a closed-set segmentation with a finite label space, whereas OVS involves a potentially infinite label space, which these methods do not account for.

To address these issues, we propose **ConOVS**, a new continual learning method that sequentially improves the performance of OVS models. Specifically, ConOVS (1) reduces training cost by using only newly collected data, unlike retraining; (2) avoids catastrophic forgetting, unlike fine-tuning; and (3) effectively improves performance on the incremental and zero-shot test dataset, unlike existing CL methods.

4 Background

Open-Vocabulary Segmentation (OVS) aims to predict segmentation mask–class pairs from an input image $\boldsymbol{x}_{\text{img}}$ and a text description $\boldsymbol{x}_{\text{text}}$, which may include both seen (trained) and unseen classes. OVS models typically consist of three components: an image encoder, a text encoder, and a decoder, denoted as $f = \{f_{\text{img}}, f_{\text{text}}, f_{\text{dec}}\}$. The image encoder f_{img} produces an image embedding $\boldsymbol{z}_{\text{img}}$, and the text encoder f_{text} produces a text embedding $\boldsymbol{z}_{\text{text}}$. These are fed into the decoder f_{dec}, which, given N learnable object queries, outputs N pairs of predicted masks and class embeddings, $\{(\boldsymbol{m}_i, \boldsymbol{c}_i)\}_{i=1}^{N}$. Each $\boldsymbol{m}_i$ is a predicted mask, and $\boldsymbol{c}_i$ is its associated class embedding. Final class labels are assigned by matching each $\boldsymbol{c}_i$ to the most similar text embedding.

Continual Learning Setup. We consider a continual learning scenario in which datasets containing new classes arrive sequentially, and the set of seen classes gradually expands over time. The model f is first trained on a pre-training dataset $\mathcal{D}_{\text{pre}}$, and then incrementally updated using a sequence of datasets $\mathcal{D}_{\text{inc},1}, \mathcal{D}_{\text{inc},2}, \cdots$. At each time step $t \in \{1, 2, \cdots, n\}$, the model is trained only on $\mathcal{D}_{\text{inc},t}$, without access to $\mathcal{D}_{\text{pre}}, \mathcal{D}_{\text{inc},1}, \cdots, \mathcal{D}_{\text{inc},t-1}$. The class set $\mathcal{C}_t$ from each incremental dataset is added to the previously seen class set, resulting in $\mathcal{C}_{\text{seen}} = \bigcup_{s=1}^{t} \mathcal{C}_s \cup \mathcal{C}_{\text{pre}}$. The model is evaluated on the test sets of all datasets up to time t to assess both its ability to learn new classes and retain prior knowledge. To additionally evaluate generalization, we use a zero-shot test set $\mathcal{D}_{\text{zero}}$ containing unseen classes $\mathcal{C}_{\text{unseen}} \subset \mathcal{C}_{\text{total}} \setminus \mathcal{C}_{\text{seen}}$ that never appeared during training.

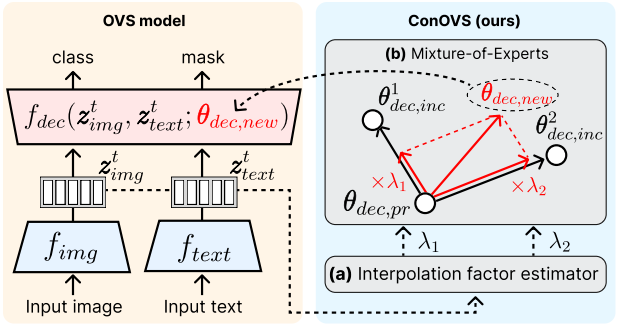


Figure 3: Overview of the inference process of our proposed method.

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 $\{\boldsymbol{\Phi}^i_{\text{img}}, \boldsymbol{\Phi}^i_{\text{text}}\}_{i=0}^n$; 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** $\boldsymbol{\lambda}$

5 The Proposed Method: ConOVS

In this section, we propose **ConOVS**, a novel MoE-based continual learning method designed to train OVS models in scenarios where new datasets are sequentially collected. For clarity, we describe the proposed method in two parts: *Training Phase* and *Inference Phase*.

5.1 Training Phase

During training, we derive *expert models* and *multivariate normal* (MVN) *distributions* for each dataset. Specifically, we first train an OVS model from scratch using the pre-training dataset. Then, we fine-tune only the decoder on each incremental dataset to obtain an expert model specific to that dataset. For each dataset, we also compute the mean and covariance matrix of the image and text embeddings, which define the MVN distributions. These are represented as $\boldsymbol{\Phi}^i_{\text{img}} = (\boldsymbol{\mu}^i_{\text{img}}, \boldsymbol{\Sigma}^i_{\text{img}})$ and $\boldsymbol{\Phi}^i_{\text{text}} = (\boldsymbol{\mu}^i_{\text{text}}, \boldsymbol{\Sigma}^i_{\text{text}})$ for each dataset $i \in \{0, \cdots, n\}$. Here, $i = 0$ corresponds to the pre-training dataset, while $i \in \{1, \cdots, n\}$ refers to the incremental datasets.

5.2 Inference Phase

We perform inference by dynamically combining expert models based on the MVN distributions derived during training. Specifically, we first compute task vectors $\boldsymbol{v}_i$ for each expert model, defined as the arithmetic difference between the decoder weights of the i-th incremental expert $\boldsymbol{\theta}^i_{\text{dec,inc}}$ and the pre-trained decoder weights $\boldsymbol{\theta}_{\text{dec,pr}}$. Given an input sample, we feed the image $\boldsymbol{x}_{\text{img}}$ and class descriptions $\boldsymbol{x}_{\text{text}}$ into the image and text encoders, respectively, to obtain the corresponding embeddings $\boldsymbol{z}_{\text{img}}$ and $\boldsymbol{z}_{\text{text}}$. We then evaluate the likelihoods of these embeddings under the MVN distributions for all datasets, and collect them into the vectors $\boldsymbol{l}_{\text{img}}, \boldsymbol{l}_{\text{text}} \in \mathbb{R}^{n+1}$.

After that, we apply the softmax operation to the log-likelihood vector to normalize the proximity scores of each domain into the $[0, 1]$ range. This decision is motivated by a prior study [16], which reported that merging performance degrades when the interpolation factor exceeds 1. Finally, we compute the element-wise maximum of the two probability vectors to obtain the final interpolation factor vector $\boldsymbol{\lambda} \in \mathbb{R}^{n+1}$. The detailed procedure is provided in Algorithm 1, and ablation studies on the choice of softmax and element-wise maximum are presented in Appendix F.

The final decoder weights $\boldsymbol{\theta}_{\text{dec,new}}$ are computed as:

$$\boldsymbol{\theta}_{\text{dec,new}} = \boldsymbol{\theta}_{\text{dec,pr}} + \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i. \tag{1}$$

That is, the decoder is dynamically constructed by linearly combining task vectors $\boldsymbol{v}_i$ with interpolation weights λ_i, relative to the pre-trained decoder (see Figure 3b). Note that while λ_0 is not directly used in this computation, it is included in the softmax operation and thus indirectly affects the other $\boldsymbol{\lambda}$ elements. As a result, when the input is close to the pre-training distribution, λ_0 approaches 1, pushing the remaining λ_i values toward 0.

The effectiveness and justification of this design are empirically validated in Section 6.

Table 1: Comparison of performance across Baselines (fc-clip, X-Decoder), Retraining, Fine-tuning, four existing continual learning methods, and ConOVS when the incremental dataset is (a) Cityscapes or (b) ADE20K. PQ is used.

(a) Cityscapes

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**

(b) ADE20K

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**

6 Experiments

Learning Sequences. This study assumes a scenario where trainable datasets arrive sequentially and evaluates OVS models that are incrementally trained on them. In the main paper, we examine three learning sequences. In Scenario 1 **(S1)**, the model is pre-trained on COCO [26], incrementally trained on Cityscapes [7], and evaluated on ADE20K [57] as the zero-shot test set. In Scenario 2 **(S2)**, the model is again pre-trained on COCO but incrementally trained on ADE20K, with Cityscapes used for zero-shot evaluation. In Scenario 3 **(S3)**, the model is pre-trained on COCO and incrementally trained on both Cityscapes and ADE20K. For zero-shot evaluation, we use a diverse collection of datasets: LVIS [10], BDD100K [51], Mapillary Vistas [33], PC-59, PC-459 [31], PAS-20, PAS-21 [8], and A-847 [57]. We further validate our method on a larger number of incremental datasets in Scenario 4 (**S4**), with the results provided in Appendix E. Evaluation is conducted on the test sets of the pre-training and incremental datasets, as well as the designated zero-shot test sets.

Implementation Details. We apply our method to two OVS models: fc-clip with ConvNeXt-L [27] and X-Decoder with Focal-L [50]. During the pre-training phase, fc-clip trains only the decoder, while X-Decoder trains both the encoder and decoder. In the fine-tuning phase, both models train only the decoder. The temperature T in the softmax is set to 0.01, and log-likelihood is used to compute probabilities from the MVN distributions. All experiments are run on two NVIDIA A5000 GPUs.

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, with the others in the Appendix J. Some zero-shot test datasets support only specific segmentation tasks; for example, LVIS supports only instance segmentation. In such cases, we evaluate performance only on the supported task.

6.1 Main Results

In this section, we compare the performance of the proposed ConOVS and other approaches under the three scenarios. We first analyze the results for scenarios S1 and S2, followed by scenario S3. We then provide a more in-depth analysis of our method, including an investigation into the behavior of the interpolation factors. All methods were trained with the same number of iterations to ensure a fair comparison, and detailed information on the training cost of each method is provided in Appendix D.1.

In scenarios **S1** and **S2**, where only a single incremental dataset is used for training, our method consistently outperforms existing approaches across all datasets, whether the incremental dataset is ADE20K or Cityscapes (see Table 1). In particular, compared to retraining, our method almost maintains or even improves performance on the pre-training dataset, despite not using it during additional training (e.g., Retraining: +1.4 vs. Ours: +1.7 in S2). It also achieves superior performance on the incremental dataset itself (e.g., Retraining: +16.5 vs. Ours: +23.8 in S2). Moreover,

compared to fine-tuning and conventional continual learning, our method improves performance on the incremental dataset without compromising performance on the pre-training dataset. This improvement is attributed to the dynamic interpolation of expert models in our method, which helps mitigate catastrophic forgetting.

Our method also achieves the best performance on the zero-shot test dataset. For instance, in scenario S2, performance on the Cityscapes improves by +0.9, whereas all other methods show performance drops. This result indicates that our method enhances recognition of a wider range of classes while preserving previously learned knowledge.

In scenario **S3**, our method consistently achieves superior performance compared to both fine-tuning and retraining. Specifically, as shown in Table 2, fine-tuning performs well only on the most recently trained dataset, whereas our method consistently achieves strong results on all three datasets. By contrast, retraining shows lower performance than our method, likely due to its need for more iterations to converge. In comparison, our method yields better results with the same number of training iterations, demonstrating greater training efficiency. Note that the analysis related to the number of training iterations in retraining is provided in Appendix G.3.

Table 2: Performance comparison in scenario S3. The best performance for each dataset is underlined. "City→ADE" means fine-tuning on Cityscapes first, then ADE20K. PQ is used.

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**

Table 3: Performance comparison on 8 unseen datasets in scenario S3. The best performance for each dataset is underlined. PQ is used.

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 Table 3, it achieves superior performance across all eight zero-shot test datasets. This result suggests that the dynamic interpolation of expert models in our method facilitates recognition of a broader range of unseen classes.

Table 4: Comparison of performance on seen and unseen classes in the zero-shot test dataset ADE20K. mIoU is used. (b) Comparison of PQ, SQ, and RQ between fc-clip and ConOVS in the zero-shot test dataset ADE20K.

(a)

Method	Seen Classes	Unseen Classes
fc-clip	35.0 (+0.0)	28.6 (+0.0)
ConOVS (ours)	**37.9** (+2.9)	**30.9** (+2.3)

(b)

Method	PQ	SQ	RQ
fc-clip	23.5 (+0.0)	61.7 (+0.0)	28.3 (+0.0)
ConOVS (ours)	**25.9** (+2.4)	**73.1** (+11.4)	**31.2** (+2.9)

Evaluation of the Truly Unseen Classes. Some classes in the zero-shot test datasets may overlap with those in the training data. For instance, ADE20K shares 38 of its 150 classes with COCO. To more accurately assess zero-shot performance, we separately evaluate the model on truly unseen classes that do not appear in the training data. Therefore, we split ADE20K into seen and unseen subsets and measure performance on each in scenario S1.

As shown in Table 4a, our method improves performance by a similar margin on both seen and unseen classes (seen: +2.9, unseen: +2.3). This suggests that the performance gain is not solely from improved recognition of seen classes, but also reflects better generalization to unseen classes.

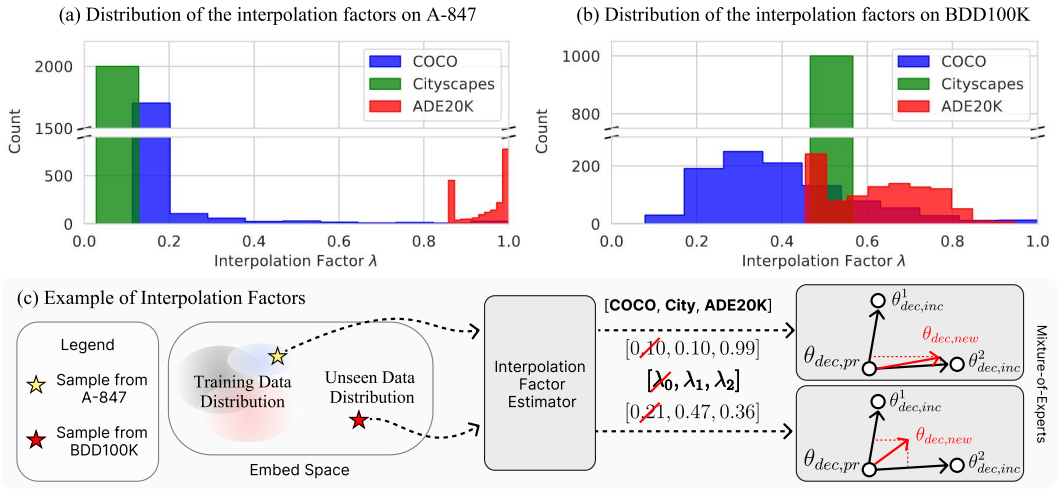


Figure 4: Interpolation factor behavior across different input sample distributions.

Analysis of Improvements in Unseen Classes. To better understand the source of performance improvements in unseen classes, we analyzed results on the zero-shot dataset ADE20K by comparing the PQ, SQ, and RQ scores of the baseline and our proposed method. As shown in Table 4b, incorporating ConOVS into the baseline model improves both PQ and RQ. The most notable gain, however, is observed in SQ, which evaluates the quality of the predicted segmentation masks. These results indicate that the improvements in unseen classes are primarily driven by enhanced segmentation quality rather than improved mask classification.

Understanding the Behavior of the Interpolation Factor. We analyze how the proposed method adapts to different input sample distributions. To this end, we examine the distribution of interpolation factors $\boldsymbol{\lambda}$ estimated by the interpolation factor estimator across two zero-shot test datasets. One is A-847, which shares a similar distribution with the incremental training dataset ADE20K, and the other is BDD100K, which differs significantly from all training datasets.

As shown in Figure 4a, the interpolation factors for A-847 tend to be close to 0 or 1. In particular, the expert trained on ADE20K receives a λ value close to 1, while other experts receive values close to 0. This shows that when input samples are similar to a previously trained distribution, our method selectively activates the corresponding expert to maximize performance (see Figure 4c top-right).

In contrast, as illustrated in Figure 4b, the interpolation factors for BDD100K are more evenly distributed between 0 and 1. This suggests that the input samples do not clearly belong to any of the known training distributions. 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. This allows the model to leverage knowledge from various datasets and produce more accurate predictions even for samples from unfamiliar domains (see Figure 4c bottom-right).

6.2 Ablation Study

In this section, we conduct ablation studies to analyze the contribution of each component in the proposed method. All experiments are conducted in scenario S1.

Ablation Study of Image and Text Distribution. Our method computes the interpolation factor of an input sample using the MVN distributions of image and text embeddings for each training dataset. To analyze how the interpolation factors are affected by the distribution design, we compare three configurations: image only, text only, and combined image-text.

As shown in Table 5a, using both image and text distributions yields the best performance on the incremental dataset. This suggests that combining both modalities enables more accurate estimation of the input sample's proximity to training distributions, leading to better expert selection.

Table 5: (a) Comparison of the interpolation factor estimator when using both image and text distributions versus using only one of them. PQ is used. (b) Performance comparison when the MVN distribution is replaced with K-means clustering or KDE. fc-clip and PQ are used.

(a)

Distribution	COCO (pre-training)	Cityscapes (incremental)	ADE20K (zero-shot)
image only	51.5	43.4	25.8
text only	**51.9**	60.7	25.9
image + text	51.6	**64.3**	26.0

(b)

Methods	COCO (pre-training)	Cityscapes (incremental)	ADE20K (zero-shot)
k-means clustering	42.4	64.1	26.1
kernel density estimation	48.1	57.4	26.1
MVN distribution	**50.4**	**64.3**	26.0

Evaluating Alternative Approaches against the MVN Distribution. We evaluate and compare two alternative techniques to the MVN distribution used in our method for estimating interpolation factors. Specifically, we replace the MVN distribution with K-means clustering or Kernel Density Estimation (KDE), and analyze the resulting performance changes. Detailed descriptions of the K-means and KDE are provided in the Appendix B.5.

As shown in Table 5b, both K-means and KDE yield lower performance on the pre-training and incremental dataset. These results suggest that the MVN distribution enables more accurate estimation of interpolation factors for in-distribution data. We attribute this to its relatively simple structure and low dimensionality, which make it less sensitive to outliers than K-means or KDE.

Replacing Softmax with Argmax. The proposed method uses the softmax function to compute interpolation factors for each dataset. We compare the performance on eight zero-shot datasets when replacing the softmax function with the argmax operation. Table 6 presents the evaluation results. The experimental results show that softmax consistently outperforms argmax across all zero-shot datasets (e.g., on LVIS, argmax: 21.3, softmax: 23.1).

Table 6: Performance comparison between the argmax and softmax operations in the interpolation factor estimator. We use fc-clip with our method and fine-tune it on both Cityscapes and ADE20K. PQ is used.

Decision Rule	Incremental Dataset	LVIS (mAP)	BDD100K (PQ)	Mapillary (mIoU)	PC-59 (mIoU)	PC-459 (mIoU)	PAS-20 (mIoU)	PAS-21 (mIoU)	A-847 (mIoU)
Argmax	Cityscapes, ADE20k	21.3	18.3	26.9	53.1	17.0	93.2	80.2	**16.3**
Softmax	Cityscapes, ADE20k	**23.1**	**22.6**	**29.1**	**54.9**	**17.9**	**93.6**	**80.7**	**16.3**

Specifically, on datasets such as LVIS and BDD100K, softmax demonstrates clearly superior performance. However, for PAS-20, PAS-21, and A-847, the performance difference between softmax and argmax is minimal. This occurs because, when the input sample is close to the distribution of the pre-training or incremental dataset, the interpolation factor obtained from softmax tends to be close to 0 or 1. As a result, softmax behaves similarly to argmax.

Hyperparameter Sensitivity Analysis. Our method uses a softmax operation to compute the interpolation factor, and we analyze the effect of the softmax temperature hyperparameter T. The temperature T directly influences the distribution of the interpolation factor: a low T smooths the factor values, while a high T pushes them toward extreme values of 0 or 1. Table 7 summarizes how this behavior affects performance.

Table 7: Effect of softmax temperature T on performance across datasets. mIoU is used.

T	COCO (pre-training)	ADE20K (incremental)	Cityscapes (zero-shot)	Total
0.0001	50.7	35.4	43.8	129.9
0.001	51.2	42.2	**43.9**	137.3
0.01	**51.8**	47.3	43.7	**142.8**
0.1	51.3	**47.5**	43.2	142.0
1.0	51.2	47.4	43.2	141.8

When T is small, the interpolation factor λ becomes overly smoothed, which prevents the expert models for each dataset from being utilized. This leads to performance degradation on the incremental dataset. In contrast, when T is large, λ converges to values close to 0 or 1, resulting in the selective use of a single expert model. This degrades performance on the zero-shot dataset. These findings suggest that appropriately integrating multiple models is essential for effective generalization to zero-shot datasets, and that extreme interpolation factors hinder this process.

Decoder Interpolation. Unlike our method, which fine-tunes the entire decoder for each dataset, existing MoE-based continual learning methods [22, 45] primarily adopt Visual Prompt Tuning (VPT), where only a small subset of parameters is trained for each incremental dataset. This approach differs from ours in two key aspects: expert models consist of only partial decoder parameters, and a single expert is selected at inference time instead of performing interpolation. To assess the effectiveness of our full decoder fine-tuning strategy, we replace it with the VPT-based approach and compare their performance.

Specifically, we implement the prompt tuning method based on [45] as follows: (1) for each incremental dataset, we train only the decoder's object queries and positional embeddings and store them in a prompt pool; (2) during inference, we compute interpolation factors for each dataset using the same procedure as our method; (3) we identify the dataset with the highest interpolation factor; and (4) retrieve the corresponding object queries and positional embeddings from the prompt pool and apply them to the decoder for prediction.

Table 8: Performance comparison when the decoder interpolation in our method is replaced with a visual prompt tuning-based approach. fc-clip and PQ are used.

Method	COCO (pre-training)	Cityscapes (incremental)	ADE20K (zero-shot)
Prompt Tuning	43.3	48.9	24.4
Decoder Interpolation	**50.4**	**64.3**	**26.0**

As shown in Table 8 and the experimental results, the prompt tuning variant consistently underperforms our method across pre-training, incremental, and zero-shot test datasets. This suggests that full decoder fine-tuning enables more effective adaptation to new datasets compared to VPT, which is constrained by its limited number of trainable parameters. Moreover, interpolating multiple experts provides greater flexibility and representational power than selecting a single expert, further supporting the advantage of our approach.

7 Limitation

Our method generates a unique decoder weight for each input sample, which can limit its applicability when the inference batch size exceeds one—a common constraint in other MoE-based continual learning approaches [41, 45]. However, since only the decoder varies per input and the encoder is shared across samples, the encoder can process inputs in batches. The resulting embeddings are then decoded individually using their corresponding weights. This design reduces the batch size limitation by supporting batched encoder processing and per-sample decoding.

8 Conclusion

This paper identifies the performance limitations of existing Open-Vocabulary Segmentation (OVS) methods on unseen data, an aspect that has been largely overlooked in prior work. To address this issue, we introduce a new learning scenario in which newly collected datasets are incrementally used to further train the OVS model. Under this setting, we show that conventional approaches—such as retraining, fine-tuning, and continual learning—are either impractical or difficult to apply effectively.

To overcome these challenges, we propose **ConOVS**, a novel MoE-based continual learning method for OVS. In ConOVS, predictions are made by dynamically combining the decoders of expert models based on the probability that the input sample belongs to the distribution of each training dataset. We validate the effectiveness of our method through extensive evaluations across various sequential learning scenarios and compare it against existing approaches. Experimental results show that ConOVS consistently achieves superior performance on pre-training, incremental, and zero-shot test datasets, demonstrating its ability to effectively expand the recognition capability of OVS models.

Broader Impacts. The proposed method can be applied to real-world applications such as robotics, where new objects continuously appear in the environment. However, if the pre-training dataset is biased, the model may continue to produce skewed predictions even after additional training, as it is explicitly designed to preserve previously learned knowledge. It is therefore important to be aware of this characteristic of the proposed technique, as a lack of such awareness may lead to unexpected model behavior.

Acknowledgement

We would like to thank Yeji Park, Beomyun Kwon, and Joonkyung Kim for the insightful discussions and valuable feedback during the development of this work. This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2025-25441313, Professional AI Talent Development Program for Multimodal AI Agents, Contribution: 50%) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00350430, Mitigating Hallucinations for Trustworthy Large Vision-Language Model: Datasets, Evaluation, Learning, and Inference, Contribution: 50%).

References

[1] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020.

[2] Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in neural information processing systems*, 34: 10919–10930, 2021.

[3] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

[4] Zhiyuan Chen and Bing Liu. *Lifelong machine learning*. Springer Nature, 2022.

[5] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16901–16911, 2024.

[6] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4123, 2024.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[9] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[10] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[11] Martin Hahner, Dengxin Dai, Christos Sakaridis, Jan-Nico Zaech, and Luc Van Gool. Semantic understanding of foggy scenes with purely synthetic data. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3675–3681. IEEE, 2019.

[12] Haiyang Huang, Newsha Ardalani, Anna Sun, Liu Ke, Shruti Bhosale, Hsien-Hsin Lee, Carole-Jean Wu, and Benjamin Lee. Toward efficient inference for mixture of experts. *Advances in Neural Information Processing Systems*, 37:84033–84059, 2024.

[13] Dongjun Hwang, Jung-Woo Ha, Hyunjung Shim, and Junsuk Choe. Entropy regularization for weakly supervised object localization. *Pattern Recognition Letters*, 169:1–7, 2023.

[14] Dongjun Hwang, Hyoseo Kim, Doyeol Baek, Hyunbin Kim, Inhye Kye, and Junsuk Choe. Curriculum learning with class-label composition for weakly supervised semantic segmentation. *Pattern Recognition Letters*, 188:171–177, 2025.

[15] Dongjun Hwang, Seong Joon Oh, and Junsuk Choe. Small object matters in weakly supervised object localization. *Neurocomputing*, page 130494, 2025.

[16] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[17] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 699–715. Springer, 2020.

[18] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2989–2998, 2023.

[19] Rawal Khirodkar, Brandon Smith, Siddhartha Chandra, Amit Agrawal, and Antonio Criminisi. Sequential ensembling for semantic segmentation. *arXiv preprint arXiv:2210.05387*, 2022.

[20] Beomyoung Kim, Joonsang Yu, and Sung Ju Hwang. Eclipse: Efficient continual learning in panoptic segmentation with visual prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3346–3356, 2024.

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[22] Minh Le, Huy Nguyen, Trang Nguyen, Trang Pham, Linh Ngo, Nhat Ho, et al. Mixture of experts meets prompt-based continual learning. *Advances in Neural Information Processing Systems*, 37:119025–119062, 2024.

[23] Minhyeok Lee, Suhwan Cho, Jungho Lee, Sunghun Yang, Heeseung Choi, Ig-Jae Kim, and Sangyoun Lee. Effective sam combination for open-vocabulary semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26081–26090, 2025.

[24] Kaican Li, Kai Chen, Haoyu Wang, Lanqing Hong, Chaoqiang Ye, Jianhua Han, Yukuai Chen, Wei Zhang, Chunjing Xu, Dit-Yan Yeung, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. In *European Conference on Computer Vision*, pages 406–423. Springer, 2022.

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[28] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[29] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *International Conference on Machine Learning*, pages 23033–23044. PMLR, 2023.

[30] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[31] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[32] Martin Mundt, Yongwon Hong, Iuliia Pliushch, and Visvanathan Ramesh. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *Neural Networks*, 160:306–336, 2023.

[33] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[34] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

[35] Reza Qorbani, Gianluca Villani, Theodoros Panagiotakopoulos, Marc Botet Colomer, Linus Härenstam-Nielsen, Mattia Segu, Pier Luigi Dovesi, Jussi Karlgren, Daniel Cremers, Federico Tombari, et al. Semantic library adaptation: Lora retrieval and fusion for open-vocabulary semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9804–9815, 2025.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[37] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016.

[38] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

[39] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7374–7383, 2019.

[40] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128:336–359, 2020.

[41] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023.

[42] Shuyang Sun, Runjia Li, Philip Torr, Xiuye Gu, and Siyang Li. Clip as rnn: Segment countless visual concepts without training endeavor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13171–13182, 2024.

[43] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3635–3647, 2024.

[44] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[45] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35: 5682–5695, 2022.

[46] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022.

[47] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *Conference on Robot Learning*, pages 384–393. PMLR, 2020.

[48] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[49] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.

[50] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022.

[51] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[52] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36, 2024.

[53] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[54] Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively. *arXiv preprint arXiv:2401.02955*, 2024.

[55] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1020–1031, 2023.

[56] Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19125–19136, 2023.

[57] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

[58] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022.

[59] Chaoyang Zhu and Long Chen. A survey on open-vocabulary detection and segmentation: Past, present, and future. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[60] Zhen Zhu, Weijie Lyu, Yao Xiao, and Derek Hoiem. Continual learning in open-vocabulary classification with complementary memory systems. *arXiv preprint arXiv:2307.01430*, 2023.

[61] Zhen Zhu, Yiming Gong, and Derek Hoiem. Anytime continual learning for open vocabulary classification. In *European Conference on Computer Vision*, pages 269–285. Springer, 2024.

[62] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127, 2023.

[63] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction provide a clear and accurate account of the paper's main contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The limitations are clearly discussed in a separate Limitations section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all necessary details, including hyperparameters and training setups, to ensure reproducibility of the main results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We disclose all necessary details for reproducibility, including code, and training scripts in the supplementary materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We detail all training and evaluation settings, including data splits, backbones, and hyperparameters, along with the rationale behind their choices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars since running multiple trials for every experimental setup would require substantial computational resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the implementation details section, we provide the necessary information to reproduce our experiments. All experiments were conducted using two NVIDIA A5000 GPUs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We rigorously follow the NeurIPS Code of Ethics in all aspects of our research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both potential positive and negative societal impacts are discussed in the Broader Impact section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper does not involve the release of any models or datasets with high risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in this work are publicly available and properly credited, with licenses and usage terms respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We ensured that the model, code are well documented for clarity and reproducibility.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Our work does not use LLMs in any important or non-standard way.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.