Decoupling Continual Semantic Segmentation

Yifu Guo[1,*], Yuquan Lu[1,*], Wentao Zhang[1], Zishan Xu[2], Dexia Chen[1], Siyu Zhang[3], Yizhe Zhang[4], Ruixuan Wang[1,5,†]

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China
[3]Business college, Southwest university, Chongqing, China
[4] School of Computer Science and Engineering, University of Notre Dame, South Bend, USA
[5]Key Laboratory of Machine Intelligence and Advanced Computing, MOE, Guangzhou, China
guoyifu1019@gmail.com, wangruix5@mail.sysu.edu.cn

Abstract

Continual Semantic Segmentation (CSS) requires learning new classes without forgetting previously acquired knowledge, addressing the fundamental challenge of catastrophic forgetting in dense prediction tasks. However, existing CSS methods typically employ single-stage encoder-decoder architectures where segmentation masks and class labels are tightly coupled, leading to interference between old and new class learning and suboptimal retention-plasticity balance. We introduce DecoupleCSS, a novel two-stage framework for CSS. By decoupling class-aware detection from class-agnostic segmentation, DecoupleCSS enables more effective continual learning, preserving past knowledge while learning new classes. The first stage leverages pre-trained text and image encoders, adapted using LoRA, to encode class-specific information and generate location-aware prompts. In the second stage, the Segment Anything Model (SAM) is employed to produce precise segmentation masks, ensuring that segmentation knowledge is shared across both new and previous classes. This approach improves the balance between retention and adaptability in CSS, achieving state-of-the-art performance across a variety of challenging tasks.

Code — https://github.com/euyis1019/Decoupling-Continual-Semantic-Segmentation

Extended version — https://doi.org/10.48550/arXiv.2508.05065

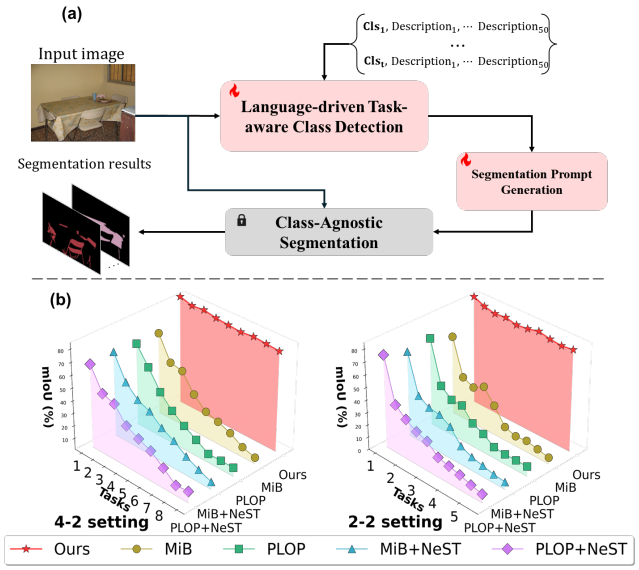

Figure 1: Overview of the proposed method. (a) The overall architecture. (b) Representative results on challenging settings (2-2 and 4-2) for CSS on Pascal VOC 2012.

Introduction

Continual Semantic Segmentation (CSS) addresses a practical scenario where new segmentation tasks with novel classes emerge over time (Cermelli et al. 2020a; Douillard et al. 2021; Yuan and Zhao 2024). A machine learning model must effectively learn these new classes while retaining previously learned old knowledge, ensuring that old knowledge is not forgotten. CSS has many real-world applications, such as autonomous driving (Camuffo and Milani 2023), medical imaging (González, Sakas, and Mukhopadhyay 2020; Wang, Wu, and Qin 2024a) and remote sensing (Tasar, Tarabalka, and Alliez 2019). As a dense prediction task, CSS is particularly susceptible to semantic shifts, where the model's understanding of class relationships may evolve over time. Furthermore, similar to other continual learning tasks, CSS faces the challenge of catastrophic forgetting, where previously learned knowledge can be influenced and overwritten by new information.

Several methods have been proposed to address the challenges in CSS. Data replay (Maracani et al. 2021a) is effective but requires storing raw data/features, leading to linear memory growth and potential privacy concerns. Regularization techniques (Cermelli et al. 2020b) penalize deviations in important model parameters for old classes, but often struggle to balance retention and plasticity. Pseudo-labeling (Douillard et al. 2021) and knowledge distilla-

*These authors contributed equally.
†Corresponding author.

tion (Su, Chen, and Wang 2024; Wang, Wu, and Qin 2024b) preserve old class knowledge by generating pseudo-labels or distilling knowledge from the old model, but still can cause background class drift when uncertain pixels are labeled as background. To mitigate this, the background weight transfer method (Park et al. 2024; Yu et al. 2025) initializes new class classifiers with background weights. However, it may mix new class features with background features, especially in cases of limited training data.

Most prior CSS methods employ a single-stage (encoder-decoder) segmentation paradigm, as seen in models such as DeepLab-V3 (Chen 2017). In these methods based on pixel-level multi-class classification, segmentation masks (intra-class consistency) and class labels (inter-class discrimination) are tightly coupled within a shared set of model parameters. This creates a significant challenge in the continual learning setting where the supervisory signals needed for inter-class discrimination between old and new categories are systematically removed, which partly causes catastrophic forgetting.

To tackle the CSS challenge, we propose a two-stage segmentation framework called DecoupleCSS in which class-aware perception (Figure 1a, red components as the first stage) is decoupled from Class-Agnostic Segmentation (CAS, gray component in Figure 1a as the second stage), enabling a separation of concerns where continual learning targets the class-aware detection phase, while the segmentation module can be shared across old and new tasks. In the first stage, class-specific semantic textual information is used to guide an adapted image encoder to extract class-relevant features. Note that different tasks share the same pre-trained frozen image encoder but have their own task-specific visual adapters which are optimized during continual learning. Such class-relevant visual features are then used to detect existence of classes in the input image and to generate class-aware and location-specific prompts. In the second stage, these prompts activate SAM to generates precise segmentation masks. Our contributions are summarized below.

- At the framework level, this study presents a novel perspective and approach for Continual Semantic Segmentation (CSS). We advocate using class-agnostic foundation models (e.g., SAM) as a cornerstone for practical CSS research and propose a separation of class-aware and class-agnostic components in CSS learning. This explicit decoupling enables effective, focused continual learning for detection (class-aware), while segmentation knowledge (class-agnostic) is shared across classes.
- At the method level, a novel task-specific class detection strategy and a novel class-specific prompt generation strategy are proposed to employ SAM for both new and previous tasks. Our method effectively integrates new classes while preserving prior knowledge.
- Our method effectively balances old knowledge retention with new knowledge learning and yields significantly improved accuracy and adaptability, demonstrating superior CSS performance across a diverse range of challenging CSS tasks (see representative results in Figure 1b).

Related Work

Continual Semantic Segmentation The challenges in CSS include catastrophic forgetting, stability-plasticity dilemma, and semantic (background) shift (Yuan and Zhao 2024). PLOP (Douillard et al. 2021) employs pseudo-labels generated by the prior model and multi-scale local distillation to preserve old knowledge. A series of works (Cermelli, Cord, and Douillard 2023; Park et al. 2024; Shang et al. 2023; Yang et al. 2023; Zhao, Yuan, and Shi 2023a,b) follow this way, aiming to enhance model stability by reducing information loss and preserving existing knowledge. For example, BalConpas (Chen et al. 2024) selectively distills the most relevant feature, and Cs^{2}KCA (Cong et al. 2024) uses pixel-level features as a prototype for each class. Though Exemplar Replay (Wang et al. 2022) has advanced significantly in CSS from three aspects: Sample Replay (Cha et al. 2021; Maracani et al. 2021b), Feature Replay (Liu et al. 2022; Yoon, Kang, and Cho 2022) and Auxiliary Data (Yu et al. 2023a), dynamic networks (Baek et al. 2022; Truong et al. 2023; Kalb et al. 2023; Xiao et al. 2023; Yang et al. 2022) stand out for their efficacy in preserving crucial parameters and enabling flexible task-specific adjustments (Gong et al. 2024), making them particularly advantageous in CSS. While recent CSS methods like CoMasTRe (Gong et al. 2024) and CoMFormer (Cermelli, Cord, and Douillard 2023) based on Mask2Former (Cheng et al. 2022) also explore decoupling strategies, with CoMasTRe focusing on objectness learning and classification separately, our framework follows a distinct detection-then-segmentation paradigm which shows superior performance.

Language-Driven Continual Learning Inspired by the observation that humans effectively acquire new visual knowledge through language and motivated by the broad applications of pre-trained vision-language models (VLMs) (Huang et al. 2023; Radford et al. 2021), previous studies have explored VLMs in continual learning classification tasks (Zhang et al. 2024). However, applications of VLMs for Continual Semantic Segmentation (CSS) are much less explored, primarily because the dense annotations required for segmentation cannot be directly leveraged by these models; only a few studies have explored their potential in Weakly Supervised Continual Semantic Segmentation (Yu et al. 2023b). To our best knowledge, our work is the first to leverage VLMs in dense-annotated CSS.

Method

This study focuses on class-incremental semantic segmentation (CISS), where a model learns a sequence of T semantic segmentation tasks. In the t-th task ($t = 1, 2, \ldots, T$), the model is updated to learn to segment each image of the task into regions of c_t new foreground classes and a background class, while preserving the ability to segment regions corresponding to the previously learned $c_1 + c_2 + \ldots + c_{t-1}$ classes. In each training image of task t, the background may contain objects of both previously learned foreground classes (from tasks 1 to $t-1$) and future classes (from tasks $t+1$ to T); this overlapped setting in CISS is more challenging than the disjoint scenario where future classes are absent

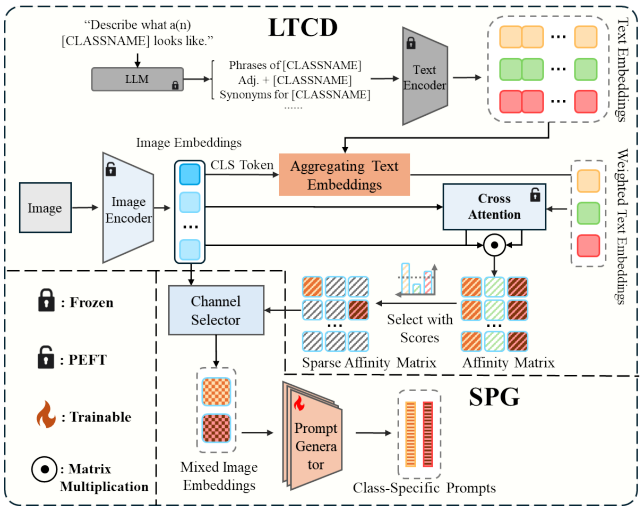

Figure 2: Workflow of the LTCD and SPG module.

during learning task t. After learning task t, the model is expected to segment any test image containing any subset of the $c_1 + c_2 + \ldots + c_t$ learned classes.

Framework Overview

The proposed CSS framework (Figure 2) is developed based on the insight that the process of segmenting an image can be decoupled into two independent stages, the existence detection of foreground classes in the image (Stage-I) and the accurate delineation of the foreground region(s) for each detected class (Stage-II). With more and more segmentation tasks learned, Stage-I should be able to detect existence of more and more classes. To achieve this goal, a novel language-driven and task-aware class detection (LTCD) module built on pre-trained large language and vision models is designed. This module can find existence of classes for each learned task with the help of task-specific adapters and a detection, and thus catastrophic forgetting of old knowledge for class detection is avoided.

For Stage-II, inspired by the class-agnostic segmentation capability of the recently developed foundation segmentation model SAM, we employ the pre-trained SAM to automatically annotate the region of each detected class. The challenge here is to provide location-relevant prompts informing SAM where to segment for each detected class. To solve this challenge, a segmentation prompt generation (SPG) module is designed to activate SAM for accurate region segmentation of each detected class. The SPG module bridges the LTCD and the CAS modules, taking the patch-level visual embeddings and language-guided class-wise image embeddings from the LTCD module as input.

In this framework, task-wise continual learning in the LTCD module and class-wise continual learning in the SPG module are responsible to continually learn knowledge of new classes, while the class-agnostic CAS module is responsible for accurate region annotation given location-relevant prompts. By isolating class detection from class-agnostic region annotation, our method provides a robust, modular solution for continual semantic segmentation.

Language-driven Task-aware Class Detection

Suppose the model will be learning the t-th task including c_t new classes. The LTCD module utilizes textual prior knowledge of each class in task t to help adapt the pre-trained image encoder and the cross attention part for task t. In particular, the encoded textual prior is used in the text-image cross attention blocks to help extract the class-specific visual embedding from the adapted image encoder for rough class awareness. Note that only the task-specific adapters which are added to the pre-trained image encoder and the pre-trained text-image cross attention block are learnable.

Text Encoding For each of the c_t classes, a number of $M + 1$ descriptive phrases are generated. Among them, M phrases are generated by the large language model (OpenAI 2023) using a pre-defined prompt (Bai and Xia 2023; Pratt, Liu, and Farhadi 2022). Unlike conventional approaches that generate complete sentences with redundant linguistic elements, our method employs concise phrasal descriptions (e.g., adjective + class name) to capture essential visual attributes. This focused representation enhances feature distinctiveness and improves class discrimination capabilities. Each phrase is then sent to a pre-trained text encoder to obtain the corresponding text embedding. Consequently, for the k-th class ($k = 1, 2, \ldots, c_t$), a set of $M + 1$ text embeddings $\{\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, \ldots, \mathbf{g}_{k,M+1}\}$ are generated. The purpose of generating multiple versions of textual descriptions and corresponding text embeddings for each class is to increase diversity and enhance the robustness of the subsequent aggregation step.

Aggregating text embeddings. While using multiple textual embeddings enriches class representations by providing a wealth of contextual information, it simultaneously introduces irrelevant or misaligned semantics that do not correspond to the specific content of the input image. To mitigate this issue, we propose an adaptive re-weighting strategy that adjusts the influence of each text embedding based on its relevance to the visual content of the current input image. For an input image $\mathbf{x}_i$, the weight for each text embedding is

$$\alpha_{i,k,m} = \frac{\exp(s_{i,k,m})}{\sum_{j=1}^{M+1} \exp(s_{i,k,j})}, \tag{1}$$

where $m \in \{1, 2, \ldots, M+1\}$, $k \in \{1, 2, \ldots, c_t\}$, and the score $s_{i,k,j} = \cos(\mathbf{V}_i^{cls}, \mathbf{g}_{k,j})$ measures the cosine similarity between the *cls* token of visual embedding $\mathbf{V}_i$ for the input image $\mathbf{x}_i$ from the adapted image encoder (see following subsection) and the j-th text embedding $\mathbf{g}_{k,j}$ of class k. The final text embedding for class k is generated based on weighted sum of all $M + 1$ text embeddings, i.e., ,

$$\mathbf{e}_{i,k} = \sum_{m=1}^{M+1} \alpha_{i,k,m} \cdot \mathbf{g}_{k,m}. \tag{2}$$

Note that $\mathbf{e}_{i,k}$ is image-wise, i.e., , different input images would lead to different text embeddings $\mathbf{e}_{i,k}$ for same class k. The set of text embeddings $\mathbf{E}_i = [\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, \ldots, \mathbf{e}_{i,c_t}] \in \mathbb{R}^{c_t \times d}$ for the c_t classes of task t will be utilized along

with the visual embedding of input image $\mathbf{x}_i$ (see below) for language-driven class detection.

Task-Specific LoRAs In our framework, we initialize the pre-trained image encoder (with a Swin Transformer backbone) and cross-attention module from Grounding DINO which are endowed with robust feature extraction and cross-modal alignment capabilities. However, while these pre-trained components provide reliable general representations, they are not specifically optimized for the nuanced requirements of learning novel classes incrementally.

Here, task-specific LoRA adapters are added into both the image encoder and the cross-attention module to learn each new segmentation task. For the image encoder, these lightweight adapters are embedded in each self-attention layer, enhancing discriminative feature extraction for new classes while maintaining the output structure as $\mathbf{V}_i \in \mathbb{R}^{N\times d}$ for each input image $\mathbf{x}_i$, where N represents visual tokens and d is the embedding dimension. In the cross-attention module, which coordinates bidirectional information flow between modalities, LoRA adapters are incorporated in the linear projections that generate attention 'keys' and 'values'. This approach enables effective text-guided visual feature optimization through multiple interaction layers, producing enhanced representations $\mathbf{V}'_i \in \mathbb{R}^{N\times d}$ and $\mathbf{E}'_i \in \mathbb{R}^{c_t\times d}$ that better capture cross-modal relationships while maintaining computational efficiency.

Note that the learnable adapters are task-specific, which prevents negative transfer and catastrophic forgetting across incremental tasks. All learned adapters for previous tasks 1 to $t-1$ are preserved but not utilized when the model learns task t. However, during model inference, adapters of each learned task will be utilized to segment any test image.

Semantic Alignment and Token Selection This section details the process of establishing precise semantic correspondence between image regions and class concepts, a crucial step for class-aware segmentation. While the cross-attention module (described above) refines visual and textual embeddings ($\mathbf{V}'_i$ and $\mathbf{E}'_i$ respectively), an explicit alignment measure is needed to guide the segmentation process.

Affinity matrix construction. To quantify the alignment between visual tokens and textual class embeddings, we compute an affinity matrix, $\mathbf{S}_i$, for each image $\mathbf{x}_i$. This matrix represents the pairwise similarity between each visual token and each class embedding, computed as follows

$$\mathbf{S}_i = \cos(\mathbf{V}'_i, {\mathbf{E}'_i}^T) \in \mathbb{R}^{N\times c_t}, \tag{3}$$

where $\cos(\cdot,\cdot)$ denotes the cosine similarity measurement. The resulting $\mathbf{S}_i$ is a matrix where each element $\mathbf{S}_i[n,k]$ represents the alignment score (dot product) between the n-th visual token and the k-th class embedding. Higher values indicate stronger alignment.

Thresholding for salient token selection. Some image regions may be irrelevant to learned classes, e.g., background regions in images. To filter out these non-salient regions and focus on the most relevant visual tokens, we apply a thresholding operation to the affinity matrix $\mathbf{S}_i$, creating a sparse affinity matrix $\mathbf{S}'_i$

$$\mathbf{S}'_i[n,k] = \begin{cases} \mathbf{S}_i[n,k], & \text{if } \mathbf{S}_i[n,k] \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where τ is a pre-defined threshold. This operation effectively eliminates weak associations, retaining only visual tokens with strong alignment to at least one class. This sparsity reduces noise and computational cost in subsequent steps.

Selection of semantically aligned tokens. From the sparse affinity matrix $\mathbf{S}'_i$, we select the visual tokens that exhibit strong alignment with any class. The selection process is defined as follows:

Identify relevant rows: We identify the rows in $\mathbf{S}'_i$ that contain at least one nonzero element. The collection of these rows corresponds to visual tokens of interest, denoted $\mathcal{I}_{row}$.

Extract selected tokens: We extract the visual embeddings corresponding to the selected row indices. This forms the set of selected visual embeddings: $\mathbf{V}_i^{\text{sel}} = \{\mathbf{V}'_i[n,:] | n \in \mathcal{I}_{row}\}$.

Determine semantic associations: For each selected token in $\mathbf{V}_i^{\text{sel}}$, we determine its associated class by identifying the class with the highest alignment score in the corresponding row of the sparse affinity matrix $\mathbf{S}'_i$. This is represented by the set of index pairs in the form of (`selected token, class`), i.e., $\mathbf{C}_i = \{(n,k) | n \in \mathcal{I}_{row}, k = \arg\max_j \mathbf{S}'_i[n,j]\}$.

Class-Specific Prompt Generation

This section describes the generation of class-specific positional prompts from the selected visual embedding ($\mathbf{V}_i^{\text{sel}}$), which are then used for class-agnostic segmentation. We propose a Segmentation Prompt Generation module which employs class-specific generators, isolating the prompt generation process for each class. This design approach confers significant architectural advantages when compared to conventional CSS methods. Traditional CSS approaches typically formulate segmentation as a pixel-level multi-class classification problem, wherein decision boundaries must be continuously calibrated to accommodate previously learned classes. Such recalibration inevitably leads to **error accumulation** across sequential tasks. By decoupling the prompt generation process for each class, we effectively isolate the learning of class-specific features, preventing interference between classes during continual learning.

Channel Selection The Channel Selector organizes the selected visual embeddings, $\mathbf{V}_i^{\text{sel}}$, based on their semantic associations, $\mathbf{C}_i$ (defined previously). For each class k, it extracts the token embeddings most strongly associated with that class. Formally, the class-specific token set, $\mathcal{T}_k$, is defined as $\mathcal{T}_k = \{\mathbf{V}_i^{\text{sel}}[n,:] | (n,j) \in \mathbf{C}_i, j = k\}$.

Prompt Generation (pGen) To create a consistent input dimension for our prompt generator, we process these class-specific tokens as follows using a max token length Q_m for every class, i.e., $\hat{\mathbf{z}}_k = \text{Flatten}(\mathcal{T}_k) \oplus \mathbf{L}_k$, where Flatten$(\cdot)$ concatenates all tokens in $\mathcal{T}_k$ into a single vector and then adjusts the length to a fixed size Q_m by either padding with zeros or truncating. $\mathbf{L}_k$ is a learnable class-specific embedding, and $\oplus$ denotes element-wise addition. The resulting vector

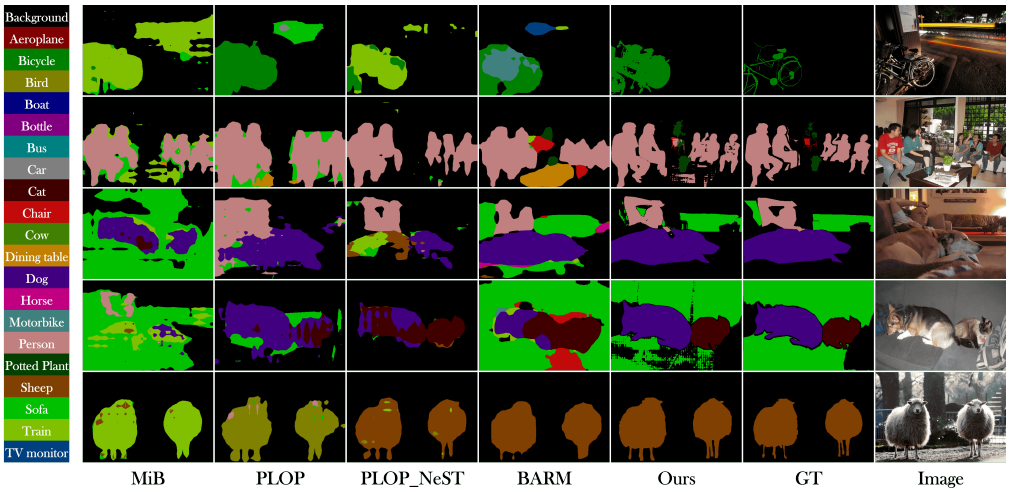

Figure 3: The visualization comparison from the last task on the Pascal VOC2012 10-1 setting.

$\hat{\mathbf{z}}_k$ has a fixed dimension d_z. By generating prompts specific to each class, *pGen could capture the unique segmentation characteristics of different classes*, which often exhibit distinct mask patterns and boundaries. This tailored prompt generation enhances the decoder's performance, enabling more precise segmentation across diverse object categories. The final step involves transforming this class-specific representation into positional prompts for SAM. This is formulated as $\mathbf{p}_k = \text{pGen}_k(\hat{\mathbf{z}}_k)$, where pGen_k is a class-specific two-layer MLP dedicated to class k, and $\mathbf{P}_k \in \mathbb{R}^{m \times d_p}$ represents m prompt tokens with dimension d_p (in our implementation, $m = 6$).

Class-Agnostic Segmentation and Aggregation

The Class-Agnostic Segmentation (CAS) module utilizes a segmentation foundation model, and we currently employ SAM (Segment Anything Model) (Kirillov et al. 2023) for this role. SAM works in a class-agnostic manner and can generate high-quality segmentation results for the input image solely based on spatial prompt conditions. Specifically, for each class-specific prompt $\mathbf{p}_k$ (from above section), SAM outputs a corresponding mask $\mathcal{M}_k$ and a confidence score. These individual class-specific masks are aggregated into a final semantic segmentation map.

It is worth noting that the CAS module is frozen in the whole continual learning process. However, when the model learns task t, the outputs from CAS are used to generate the segmentation loss function. In our framework, the same segmentation loss as that for SAM training is adopted to train the learnable class-specific prompt generators, and the loss is also used together with the asymmetric loss (Ridnik et al. 2021) to train the task-specific LoRAs in the LTCD module.

During model inference, given a test image, the task-specific and class-specific components of each task are plugged into the segmentation system to segment image regions specifically for those classes estimated appearing in the input image. Such inference process is run over all tasks and the results are collected for the final segmentation. The final segmentation is created by aggregating individual class masks. While pixel ownership is unambiguous for the vast majority of pixels, any rare spatial overlaps are resolved using a confidence-based approach. The pixel is assigned to the class with the highest confidence score from the SAM decoder. Notably, this process requires no task ID, with time linear complexity with respect to the number of learned tasks T. Since only one adapter resides in memory at any time, maximum memory usage remains identical to that in single-task models. For time-sensitive applications, memory-time tradeoff techniques (Sheng et al. 2023; Lin et al. 2025; Huang et al. 2025) can be used to reduce inference latency.

Experiments

Experimental Setup

Datasets Following previous studies (Cermelli et al. 2020b; Douillard et al. 2021), we evaluate our method on PASCAL VOC2012 (Everingham et al. 2010) and ADE20K (Zhou et al. 2017). PASCAL VOC2012 contains 20 object classes while ADE20K presents a more challenging scenario with 150 classes. Detailed dataset statistics can be found in the extended version.

CSS Settings. The training process is divided into T tasks with certain protocol. The *overlapped* protocol, which is more challenging and realistic compared to the *disjoint* setting, allows images to contain old classes previously learned and future classes to be learned (Qiu et al. 2023). For these images, annotations are provided only for the current task's classes and image regions corresponding to both old and future classes are annotated as background. Our experiments use this *overlapped* setting throughout. For instance, in the VOC2012 10-1 (11 tasks) setting, the model first learns to segment 10 classes, then incrementally learns one new class in each of 10 new tasks.

Implementation details. The model was trained for 5 epochs on PASCAL VOC2012 and 20 epochs on ADE20K using AdamW (Loshchilov 2017) optimizer (initial learning rate 1×10^{-4}, weight decay 0.05) with a polynomial learn-

	Method	**19-1** (2 tasks)			**15-5** (2 tasks)			**15-1** (6 tasks)			**10-1** (11 tasks)		
		0-19	*20*	*all*	*0-15*	*16-20*	*all*	*0-15*	*16-20*	*all*	*0-10*	*11-20*	*all*
Replay	MicroSeg-M (Zhang et al. 2022)	—	—	—	82.9	60.1	77.5	82.0	47.3	73.3	78.9	59.2	70.1
	SSUL-M (Cha et al. 2021)	77.38	22.43	74.76	79.3	55.1	73.5	78.8	49.7	71.9	75.3	54.1	65.2
	RECALL (Maracani et al. 2021b)	67.9	53.5	68.4	66.6	50.9	64.0	65.7	47.8	62.7	59.5	46.7	54.8
	SATS-M (Qiu et al. 2023)	—	—	—	81.44	70.02	78.72	80.37	64.54	76.61	76.21	61.62	69.27
	IPSeg-M† (Yu et al. 2025)	—	—	—	83.3	73.3	80.9	83.5	75.1	81.5	80.3	76.7	78.6
Data-free	MiB* (Cermelli et al. 2020b)	69.91	20.63	67.45	75.48	49.41	68.47	36.71	12.12	30.82	12.20	13.19	12.61
	PLOP* (Douillard et al. 2021)	74.15	35.58	72.04	75.49	49.66	69.34	64.09	20.12	53.11	44.03	15.51	30.45
	MiB+NeST†* (Xie et al. 2024)	70.25	26.06	68.91	75.46	48.68	69.47	60.24	19.97	48.97	52.36	21.07	37.41
	PLOP+NeST†* (Xie et al. 2024)	76.09	47.93	73.82	76.11	48.47	68.44	48.97	23.28	48.18	54.21	17.83	36.91
	MicroSeg (Zhang et al. 2022)	—	—	—	81.9	54.0	75.2	80.5	40.8	71.0	73.5	53.0	63.8
	CoMFormer (Cermelli, Cord, and Douillard 2023)	75.35	24.06	72.91	74.68	54.30	71.12	70.78	32.24	61.60	—	—	—
	CoMasTRe† (Gong et al. 2024)	75.13	69.51	74.86	79.73	51.93	73.11	69.77	43.62	63.54	—	—	—
	IPSeg (Yu et al. 2025)	—	—	—	81.4	62.4	76.9	82.4	52.9	75.4	80.0	61.2	71.0
	SATS* (Qiu et al. 2023)	77.42	61.07	74.41	80.24	61.17	75.70	78.38	62.02	74.48	64.27	58.66	61.60
	BARM†* (Zhang and Gao 2024)	77.6	41.4	75.2	—	—	—	77.3	45.8	69.8	72.2	49.8	61.9
	SSUL (Cha et al. 2021)	—	—	—	79.7	55.3	73.9	78.1	33.4	67.5	74.3	51.0	63.2
	Ours	**82.92**	**83.71**	**83.95**	**84.03**	**81.68**	**83.47**	**83.81**	**82.12**	**83.40**	**84.03**	**82.12**	**83.12**

Table 1: Comparison with existing methods on PASCAL VOC2012 in mIoU (%). The 1st highest results among Replay methods and the 1st highest results among Data-free methods (excluding our method) are marked with underline. † means the latest methods proposed from 2024 to 2025. * is on behalf of our own implementation.

	Method	**100-50** (2 tasks)			**100-10** (6 tasks)			**100-5** (11 tasks)		
		0-100	*101-150*	*all*	*0-100*	*101-150*	*all*	*0-100*	*101-150*	*all*
Replay	IPSeg-M† (Yu et al. 2025)	43.8	31.5	39.7	43.0	30.9	39.0	43.2	30.4	38.9
	SSUL-M (Cha et al. 2021)	41.5	48.0	33.7	41.6	19.9	34.4	41.6	20.1	34.5
	TIKP† (Yu et al. 2024)	42.2	20.2	34.9	41.0	19.6	33.8	37.5	17.6	30.9
Data-free	MiB* (Cermelli et al. 2020b)	39.02	16.73	31.29	36.68	9.81	27.77	34.22	5.26	24.29
	PLOP* (Douillard et al. 2021)	40.38	13.41	31.52	39.46	12.58	30.10	38.12	7.32	27.39
	MiB+NeST†* (Xie et al. 2024)	38.84	23.11	33.55	38.79	19.10	32.24	38.39	17.46	31.23
	PLOP+NeST†* (Xie et al. 2024)	40.78	22.78	34.84	39.42	20.50	33.21	37.83	16.89	30.53
	CoMFormer (Cermelli, Cord, and Douillard 2023)	44.70	26.20	38.40	40.60	15.60	32.30	39.50	13.60	30.90
	CoMasTRe† (Gong et al. 2024)	45.73	26.02	39.20	42.32	18.42	34.41	40.82	15.83	32.55
	BARM†* (Zhang and Gao 2024)	42.0	23.0	35.7	41.1	23.1	35.2	40.5	21.2	34.1
	BalConpas-R† (Chen et al. 2024)	50.8	30.4	44.0	48.1	25.3	40.5	43.9	22.7	36.9
	IPSeg† (Yu et al. 2025)	43.2	29.0	38.4	42.5	27.8	37.6	43.1	26.2	37.6
	SSUL (Cha et al. 2021)	41.9	20.1	34.6	40.7	19.0	33.5	41.3	16.0	32.9
	Ours	**57.72**	**51.21**	**56.80**	**58.19**	**52.03**	**56.92**	**57.53**	**55.62**	**56.89**

Table 2: Comparison with existing methods on ADE20K in mIoU (%).

ing rate schedule. For the LTCD module, we set the similarity threshold τ to 0.3 for PASCAL VOC2012 and 0.2 for ADE20K, ensuring an appropriate quantity of selected visual tokens in $\mathbf{V}_i^{\text{sel}}$, with $M = 30$ descriptive phrases per class and LoRA rank $L = 32$. The SPG module uses $m = 6$ prompt tokens with a single hidden layer, and pGen input lengths of 512 and 1024 for PASCAL VOC2012 and ADE20K respectively, with the longer length for ADE20K accommodating its complex scene composition and higher object density.

Main Results

Tables 1 and 2 present the evaluation results of the proposed method on PASCAL VOC2012 and ADE20K. On VOC2012, our method achieves the best performance across all the four common CSS splitting settings, with 83.12% in the challenging 10-1 setting, surpassing the previous SOTA method IPSeg (using data replay strategy) by a significant margin of 4.52%. The visual examples in Figure 3 show that our method has good mask quality while maintaining resilience to semantic shift and catastrophic forgetting. Similarly on the more challenging ADE20K dataset, our method also excels in all the 100-50, 100-10 and 100-5 settings. Notably, our method significantly outperforms the SOTA method with a 17.99% mIoU increase in the 100-5 setting. These results demonstrate its strong performance in learning new knowledge while preserving old knowledge.

Furthermore, in the more challenging CSS settings where the number of classes learned in the first task is the same as that in each subsequent task, including the settings 2-2 (10 tasks), 4-2 (9 tasks), and 4-4 (5 tasks), the superiority of our method becomes more evident. As shown in Figure 1b, our method consistently maintained both plasticity and stability in these highly challenging scenarios, outperforming competing methods by huge gaps of 74.94%, 73.35%, and 57.46% respectively.

Ablation Studies

Ablation studies were performed to verify the validity of three key components in our framework: LoRA, pGen, and semantic aggregation. As shown in Table 3, removing each of three components leads to worse performance in all task settings. In addition, when class-specific pGen was replaced by classes-shared pGen, a dramatic drop particularly in 10-1 and 15-1 settings was observed. This drop happened because, with the shared pGen, the system had to adjust for

Components			19-1 (2 tasks)			10-1 (11 tasks)			15-1 (6 tasks)		
L	CG	S	*0-19*	*20*	*all*	*0-10*	*11-20*	*all*	*0-15*	*16-20*	*all*
✗	✗	✗	25.16	62.79	26.95	22.32	25.81	23.98	25.42	26.28	25.62
✓	✓	✗	79.37	80.26	79.41	81.22	80.51	80.88	80.35	80.82	80.46
✓	✗	✓	29.54	77.41	31.81	28.76	27.43	27.89	30.14	28.17	28.83
✗	✓	✓	76.13	77.97	76.21	77.64	76.33	77.01	78.38	77.12	78.03
✓	✓	✓	82.92	83.71	83.95	84.03	82.12	83.12	83.81	82.12	83.40

Table 3: Ablation study on PASCAL VOC2012, evaluated in terms of mIoU (%). The components include LoRA (L), Class-specific pGen (CG), and semantic aggregation (S).

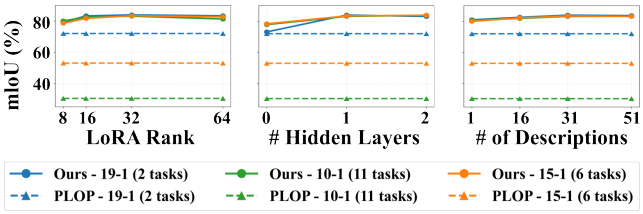

Figure 4: Sensitivity study on PASCAL VOC2012.

each new class during training. All these results confirm the necessity of the proposed components in our framework.

Sensitivity Studies

The sensitivity of our method to hyper-parameter choice is also evaluated. As demonstrated in Figure 4, when varying the LoRA ranks within the range $[16, 64]$, the number of hidden layers in the class-specific pGen module within the range $[0, 2]$, the number of textual descriptions for each class based on the LLM within the range $[1, 51]$, our method performs stably well and always better than the representative strong baseline, supporting the robustness of our method to hyper-parameter choice.

Additional Studies

Additional experiments show that incorporating powerful foundation models like SAM is insufficient for effective CSS, and that the proposed class-specific prompt generation in our method is necessary.

As shown in Figure 5, applying SAM as a post-processing module to existing CSS methods yields only marginal improvements, with less than 6% mIoU across various settings. This limited gain reveals fundamental issues when SAM is applied directly to CSS outputs: blurred edges in CSS predictions compromise SAM's segmentation, and SAM's class-agnostic nature cannot correct semantic inconsistencies. In contrast, our method with the proposed prompt generation module, which generates precise class-aware positional embeddings, overcomes these limitations by intelligently guiding SAM to produce refined segmentation masks.

To further validate the prompt generation module, ground-truth class existence information and bounding boxes for each class region were obtained for each test image by Grounding DINO, and used as spatial prompts for SAM to produce segmentation masks. This approach achieves 84% mIoU on PASCAL VOC2012 and 57% mIoU on ADE20K. While this performance is comparable to our method (83% on VOC2012 and 56% on ADE20K), our approach does not require ground-truth tags during inference and operates under the constraints of continual learning. This demonstrates that our class-specific prompt generation effectively bridges detection and segmentation in CSS scenarios, achieving near-optimal performance without perfect class knowledge.

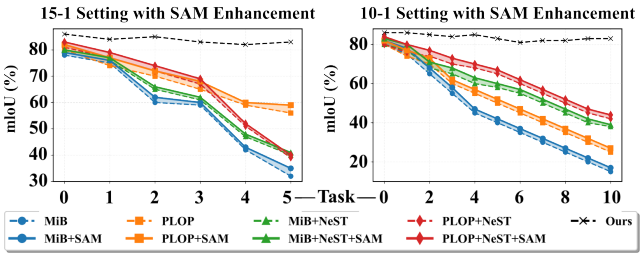

Figure 5: Segmentation performance on PASCAL VOC2012 with and without SAM enhancement.

Finally, we analyse the computational cost of our approach. Similar to continual learning methods employing task-specific modules, our main inference overhead comes from switching task-specific LoRA parameters. The inference time for a single task is 0.3 seconds per image, and the full process for six tasks in the (15-1) setting takes about 1.9 seconds per image; by comparison, single encoder–decoder CSS methods take around 0.5 seconds per image. Despite leveraging foundation models, our parameter growth remains modest: each class requires only 8MB for the pGen module (0.225% of total model size), and all tasks' LoRA adapters collectively need 22.18MB (0.62% of total model size). Each task adds relatively few trainable parameters, yet achieves state-of-the-art performance with gains in challenging settings, yielding practical applicability comparable to offline-trained models. Given our method's performance gains and affordable storage overhead, this trade-off is justified for non-realtime applications. Moreover, in a subsequent system we integrate the S-LoRA (Sheng et al. 2023) serving mechanism to handle multiple task-specific adapters concurrently, and empirically observe that per-image inference time grows sub-linearly with the number of tasks, indicating that the adapter-switching overhead can be effectively amortized at scale.

Conclusion

In this work, we propose a two-stage framework for Continual Semantic Segmentation (CSS) that decouples class-aware detection from class-agnostic segmentation. Our method achieves state-of-the-art performance across various CSS tasks. The superior performance stems from task-specific modeling with conflict-free formulation. By leveraging pre-trained text and image encoders, this study provides a practical solution for real-world applications such as autonomous driving and medical imaging, offering a scalable path for CSS. The main limitation of our method is the inference time due to the sequential switching of task-specific parameters. Future work could explore parameter merging techniques to address this scalability issue.

Acknowledgment

This work is supported in part by the National Natural Science of China (grant No. 62571559), the Major Key Project of PCL (grant No. PCL2025AS209), and Guangdong Excellent Youth Team Program (grant No. 2023B1515040025).

References

Baek, D.; Oh, Y.; Lee, S.; Lee, J.; and Ham, B. 2022. Decomposed knowledge distillation for class-incremental semantic segmentation. *Advances in Neural Information Processing Systems*, 35: 10380–10392.

Bai, X.; and Xia, Y. 2023. SAM++: Enhancing Anatomic Matching using Semantic Information and Structural Inference.

Camuffo, E.; and Milani, S. 2023. Continual Learning for LiDAR Semantic Segmentation: Class-Incremental and Coarse-to-Fine strategies on Sparse Data. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2447–2456.

Cermelli, F.; Cord, M.; and Douillard, A. 2023. CoMFormer: Continual Learning in Semantic and Panoptic Segmentation. In *CVPR*.

Cermelli, F.; Mancini, M.; Bulo, S. R.; Ricci, E.; and Caputo, B. 2020a. Modeling the Background for Incremental Learning in Semantic Segmentation. In *CVPR*.

Cermelli, F.; Mancini, M.; Bulò, S. R.; Ricci, E.; and Caputo, B. 2020b. Modeling the Background for Incremental Learning in Semantic Segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9230–9239.

Cha, S.; Yoo, Y.; Moon, T.; et al. 2021. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in neural information processing systems*, 34: 10919–10930.

Chen, J.; Cong, R.; Luo, Y.; Ip, H. H.-S.; and Kwong, S. 2024. Strike a Balance in Continual Panoptic Segmentation. *ECCV*.

Chen, L.-C. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1290–1299.

Cong, W.; Cong, Y.; Liu, Y.; and Sun, G. 2024. Cs2K: Class-specific and Class-shared Knowledge Guidance for Incremental Semantic Segmentation. *ECCV*, abs/2407.09047.

Douillard, A.; Chen, Y.; Dapogny, A.; and Cord, M. 2021. PLOP: Learning without Forgetting for Continual Semantic Segmentation. In *CVPR*.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2): 303–338.

Gong, Y.; Yu, S.; Wang, X.; and Xiao, J. 2024. Continual Segmentation with Disentangled Objectness Learning and Class Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3848–3857.

González, C.; Sakas, G.; and Mukhopadhyay, A. 2020. What is Wrong with Continual Learning in Medical Image Segmentation? *ArXiv*, abs/2010.11008.

Huang, J.; Feng, X.; Chen, Q.; Zhao, H.; Cheng, Z.; Bai, J.; Zhou, J.; Li, M.; and Qin, L. 2025. MLDebugging: Towards Benchmarking Code Debugging Across Multi-Library Scenarios. *arXiv preprint arXiv:2506.13824*.

Huang, X.; Huang, Y.-J.; Zhang, Y.; Tian, W.; Feng, R.; Zhang, Y.; Xie, Y.; Li, Y.; and Zhang, L. 2023. Open-Set Image Tagging with Multi-Grained Text Supervision.

Kalb, T.; Ahuja, N.; Zhou, J.; and Beyerer, J. 2023. Effects of architectures on continual semantic segmentation. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, 1–8. IEEE.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Lin, J.; Guo, Y.; Han, Y.; Hu, S.; Ni, Z.; Wang, L.; Chen, M.; Liu, H.; Chen, R.; He, Y.; Jiang, D.; Jiao, B.; Hu, C.; and Wang, H. 2025. SE-Agent: Self-Evolution Trajectory Optimization in Multi-Step Reasoning with LLM-Based Agents. arXiv:2508.02085.

Liu, J.; Bao, Y.; Xie, G.-S.; Xiong, H.; Sonke, J.-J.; and Gavves, E. 2022. Dynamic prototype convolution network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11553–11562.

Loshchilov, I. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Maracani, A.; Michieli, U.; Toldo, M.; and Zanuttigh, P. 2021a. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7026–7035.

Maracani, A.; Michieli, U.; Toldo, M.; and Zanuttigh, P. 2021b. RECALL: Replay-Based Continual Learning in Semantic Segmentation. *International Conference on Computer Vision,International Conference on Computer Vision*.

OpenAI, O. 2023. GPT-4 Technical Report.

Park, G.; Moon, W.; Lee, S.; Kim, T.-Y.; and Heo, J.-P. 2024. Mitigating Background Shift in Class-Incremental Semantic Segmentation. *ECCV*, abs/2407.11859.

Pratt, S.; Liu, R.; and Farhadi, A. 2022. What does a platypus look like? Generating customized prompts for zero-shot image classification.

Qiu, Y.; Shen, Y.; Sun, Z.; Zheng, Y.; Chang, X.; Zheng, W.; and Wang, R. 2023. SATS: Self-attention transfer for continual semantic segmentation. *Pattern Recognition*, 138: 109383.

Radford, A.; Kim, J.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Amanda, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable

Visual Models From Natural Language Supervision. *Cornell University - arXiv,Cornell University - arXiv*.

Ridnik, T.; Ben-Baruch, E.; Zamir, N.; Noy, A.; Friedman, I.; Protter, M.; and Zelnik-Manor, L. 2021. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, 82–91.

Shang, C.; Li, H.; Meng, F.; Wu, Q.; Qiu, H.; and Wang, L. 2023. Incrementer: Transformer for class-incremental semantic segmentation with knowledge distillation focusing on old class. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7214–7224.

Sheng, Y.; Cao, S.; Li, D.; Hooper, C.; Lee, N.; Yang, S.; Chou, C.; Zhu, B.; Zheng, L.; Keutzer, K.; et al. 2023. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*.

Su, Y.; Chen, S.; and Wang, Y.-G. 2024. Balanced Residual Distillation Learning for 3D Point Cloud Class-Incremental Semantic Segmentation. *arXiv preprint arXiv:2408.01356*.

Tasar, O.; Tarabalka, Y.; and Alliez, P. 2019. Incremental Learning for Semantic Segmentation of Large-Scale Remote Sensing Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP: 1–14.

Truong, T.-D.; Nguyen, H.-Q.; Raj, B.; and Luu, K. 2023. Fairness continual learning approach to semantic scene understanding in open-world environments. *Advances in Neural Information Processing Systems*, 36: 65456–65467.

Wang, H.; Wu, H.; and Qin, J. 2024a. Incremental Nuclei Segmentation from Histopathological Images via Future-class Awareness and Compatibility-inspired Distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11408–11417.

Wang, H.; Wu, H.; and Qin, J. 2024b. Incremental Nuclei Segmentation from Histopathological Images via Future-class Awareness and Compatibility-inspired Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11408–11417.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 139–149.

Xiao, J.-W.; Zhang, C.-B.; Feng, J.; Liu, X.; van de Weijer, J.; and Cheng, M.-M. 2023. Endpoints weight fusion for class incremental semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7204–7213.

Xie, Z.; Lu, H.; Xiao, J.-W.; Wang, E.; Zhang, L.; and Liu, X. 2024. Early Preparation Pays Off: New Classifier Pre-tuning for Class Incremental Semantic Segmentation. *ECCV*, abs/2407.14142.

Yang, X.; Zhou, D.; Liu, S.; Ye, J.; and Wang, X. 2022. Deep model reassembly. *Advances in neural information processing systems*, 35: 25739–25753.

Yang, Z.; Li, R.; Ling, E.; Zhang, C.; Wang, Y.; Huang, D.; Ma, K. T.; Hur, M.; and Lin, G. 2023. Label-guided knowledge distillation for continual semantic segmentation on 2d images and 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18601–18612.

Yoon, J.; Kang, D.; and Cho, M. 2022. Semi-supervised domain adaptation via sample-to-sample self-distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1978–1987.

Yu, C.; Zhou, Q.; Li, J.; Yuan, J.; Wang, Z.; and Wang, F. 2023a. Foundation model drives weakly incremental learning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23685–23694.

Yu, C.; Zhou, Q.; Li, J.; Yuan, J.; Wang, Z.; and Wang, F. 2023b. Foundation Model Drives Weakly Incremental Learning for Semantic Segmentation. arXiv:2302.14250.

Yu, X.; Fang, Y.; Zhao, Y.; and Wei, Y. 2025. IPSeg: Image Posterior Mitigates Semantic Drift in Class-Incremental Segmentation. *arXiv preprint arXiv:2502.04870*.

Yu, Z.; Yang, W.; Xie, X.; and Shi, Z. 2024. TIKP: Text-to-image knowledge preservation for continual semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16596–16604.

Yuan, B.; and Zhao, D. 2024. A survey on continual semantic segmentation: Theory, challenge, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, A.; and Gao, G. 2024. Background Adaptation with Residual Modeling for Exemplar-Free Class-Incremental Semantic Segmentation. *ECCV*, abs/2407.09838.

Zhang, W.; Huang, Y.; Zhang, W.; Zhang, T.; Lao, Q.; Yu, Y.; Zheng, W.-S.; and Wang, R. 2024. Continual Learning of Image Classes with Language Guidance from a Vision-Language Model. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhang, Z.; Gao, G.; Fang, Z.; Jiao, J.; and Wei, Y. 2022. Mining unseen classes via regional objectness: A simple baseline for incremental segmentation. *Advances in neural information processing systems*, 35: 24340–24353.

Zhao, D.; Yuan, B.; and Shi, Z. 2023a. Inherit with distillation and evolve with contrast: Exploring class incremental semantic segmentation without exemplar memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 11932–11947.

Zhao, D.; Yuan, B.; and Shi, Z. 2023b. Inherit with distillation and evolve with contrast: Exploring class incremental semantic segmentation without exemplar memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 11932–11947.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene Parsing Through ADE20K Dataset. In *CVPR*.