SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Towards Data Efficient and Continual Semantic Segmentation

Submitted by

Lanyun ZHU

Thesis Advisor

Dr. De Wen SOH

Information Systems Technology and Design

A thesis submitted to the Singapore University of Technology and Design in
fulfillment of the requirement for the degree of Doctor of Philosophy

2025

PhD Thesis Examination Committee

TEC Chair:	Prof. Lu Wei
Main Advisor:	Prof. Soh De Wen
Internal TEC member 1:	Prof. Zhao Na
Internal TEC member 2:	Prof. Roy Ka-Wei Lee

To my parents and grandmother, for their unconditional love.

Abstract

Information Systems Technology and Design

Doctor of Philosophy

Towards Data Efficient and Continual Semantic Segmentation

by Lanyun ZHU

Semantic segmentation is a fundamental and important task in computer vision, which aims to classify each pixel in an image. The rapid development of deep learning has significantly advanced semantic segmentation and improved the accuracy, promoting its application in fields with high accuracy requirements for pixel-level prediction, such as autonomous driving and medical diagnosis.

Current works for semantic segmentation are typically based on a standard setup that all data is accessible beforehand and can be learned simultaneously. However, in many real-world applications, due to the ongoing and dynamic nature of data generation process and the need for business scalability, training data is often not available all at once but is instead provided incrementally across multiple stages. This setup introduces two main issues: first, in the early stages, the limited number of available training samples—due to the incomplete dataset—makes effective model training difficult; second, after training on data from new stages, the model may suffer from catastrophic forgetting, losing previously learned knowledge from earlier data. Addressing these challenges is crucial for developing high-performance segmentation algorithms under multi-stage training conditions.

This thesis aims to address the above challenges by focusing on continual learning and data-efficient few-shot learning for semantic segmentation. First, we propose a novel method for training segmentation models with limited data, in which we identify and resolve the issue of context bias caused by the varying backgrounds among different images. Next, we introduce the first large language model (LLM)-based approach for few-shot semantic segmentation, leveraging the extensive knowledge within LLMs to compensate for the limited information provided by few-shot training samples. Finally, we propose a novel sample selection mechanism using reinforcement learning, which automatically selects a small number of samples from past stages for replay in future training stages, thus effectively mitigating the problem of catastrophic forgetting. Experiments on multiple datasets and scenarios demonstrate that the proposed methods can enable effective data-efficient and continual semantic segmentation.

Publications

* refers to co-first author.

- **Lanyun Zhu**, Tianrun Chen, Deyi Ji, Jieping Ye and Jun Liu. LLaFS: When Large Language Models Meet Few-Shot Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2024.*

- **Lanyun Zhu**, Tianrun Chen, Jianxiong Yin, Simon See and Jun Liu. Addressing Background Context Bias in Few-Shot Segmentation through Iterative Modulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2024.*

- **Lanyun Zhu**, Tianrun Chen, Jianxiong Yin, Simon See and Jun Liu. Learning Gabor Texture Features for Fine-Grained Recognition. In *International Conference on Computer Vision (ICCV) 2023.*

- **Lanyun Zhu***, Tianrun Chen*, Jianxiong Yin, Simon See and Jun Liu. Continual Semantic Segmentation with Automatic Memory Sample Selection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2023.*

- Qianxiong Xu, Xuanyi Liu, **Lanyun Zhu**, Guosheng Lin, Cheng Long, Ziyue Li and Rui Zhao, "Hybrid Mamba for Few-Shot Segmentation", *Annual Conference on Neural Information Processing Systems (NeurIPS), 2024.*

- Deyi Ji, Feng Zhao, **Lanyun Zhu**, Wenwei Jin, Hongtao Lu and Jieping Ye. Discrete Latent Perspective Learning for Segmentation and Detection. In *International Conference on Machine Learning (ICML) 2024.*

- Tianrun Chen*, **Lanyun Zhu***, Chaotao Ding, Runlong Cao, Shangzhan Zhang, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao and Ying Zang. SAM-Adapter: Adapting Segment Anything in Underperformed Scenes. In *ICCV2023 1st Workshop on Visual Continual Learning.*

- Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, **Lanyun Zhu**, Xiaowei Zhou, Andreas Geiger and Yiyi Liao. Panoptic NeRF: 3D-to-2D Label Transfer for Panoptic Urban Scene Segmentation. In *International Conference on 3D Vision (3DV) 2022.*

- Tianrun Chen, Chaotao Ding, **Lanyun Zhu**, Ying Zang, Yiyi Liao, Zejian Li, and Lingyun Su. Reality3DSketch: Rapid 3D Modeling of Objects from Single Freehand Sketches. In *IEEE Transactions on Multimedia (TMM).*

- Yan Wang, Jian Cheng, Yixin Chen, Shuai Shao, **Lanyun Zhu**, Zhenzhou Wu, Tao Liu and Haogang Zhu. FVP: Fourier Visual Prompting for Source-Free Unsupervised Domain Adaptation of Medical Image Segmentation. In *IEEE Transactions on Medical Imaging (TMI).*

Acknowledgements

The past few years of my Ph.D. journey have been among the most significant and rewarding times of my life. The relentless pursuit of scientific research has not only honed my skills as a researcher but also deepened my understanding of the values of perseverance, dedication, and resilience. I am deeply grateful to all those who have accompanied and supported me along this journey; their encouragement and assistance have been the driving force behind my continued progress.

First and foremost, I would like to express my deepest gratitude to my former advisor, Professor Jun Liu, for providing me with the opportunity to pursue my Ph.D. and for his unwavering commitment to shaping me into a qualified computer vision researcher. I will never forget the countless hours he spent discussing every detail of my papers and meticulously revising each paragraph of my rebuttals. His diligence, perseverance, and rigorous attitude to scientific research have profoundly influenced me and will continue to benefit me throughout my life. I am especially grateful for his understanding and respect for my research directions and working style, which allowed me to pursue my work with joy and enthusiasm. It has been a great honor to spend three years under his guidance, an experience I will cherish as one of the most valuable in my life.

I would also like to express my heartfelt gratitude to my current advisor, Professor De Wen Soh. From the transition of advisors to completing my thesis and preparing for my defense, he has gone above and beyond to provide me with every possible assistance. Without his support and assistance through these intricate and challenging processes, I would have faced unimaginable difficulties and stress. I am deeply thankful for his respect for my research work and his unwavering support for my graduation request. Although our collaboration has been less than one year, the impact of his assistance and support will stay with me for a lifetime.

During these years of study, I have been fortunate to encounter many individuals who have supported and assisted me in my research. I would like to extend my gratitude to all other members in my group. Their kindness created the best possible research environment, their diligence has consistently inspired me to push forward, and their unique insights always provided me with new inspiration. I would like to express my gratitude to AI Singapore and its staff for the generous scholarship that supported my Ph.D.studies with a high standard of living. I am also grateful to Mr. Jianxiong Yin and Dr. Simon See from NVIDIA, as well as Mr. Deyi Ji and Dr. Jieping Ye from Alibaba, for providing the computational resources that greatly facilitated my experiments. I would like to express special thanks to my most important collaborator during my Ph.D study, Tianrun Chen. We engaged in numerous thought-provoking discussions and collaborated on many fascinating research projects. He is a brilliant researcher and an exceptional entrepreneur who is always willing to offer help and share his insights. I have learned a lot from him and have thoroughly enjoyed the time we spent working together over the past few years.

Outside the research community, I am fortunate to have a group of friends who love and support me unconditionally. I am especially grateful to Dr. Shunqi Liu for the comfort and encouragement she offered me during my lowest moments. Her kindness and patience helped me overcome anxiety and pain, and her stories and perspectives always allowed me to step back from research and appreciate other values and joys in life. She is also an outstanding researcher. Although we work in different fields, her ideas can always provide me with valuable inspiration. I sincerely wish her all the best in both her personal and professional life. More importantly, I want to express my gratitude to Qing Weipin for her love and support. Without her love and companionship, I would not have been able to complete this challenging Ph.D. journey so smoothly. She stood by me through the difficult days and nights of anxiety over my research work, and her understanding and respect allowed me to continuously pursue my greater ambitions with passion. I look forward to the day we can be reunited, with our two beloved cats, Zhu Menghua and Jin Zhengen.

Lastly but most importantly, I want to express my deepest gratitude to my parents and my grandmother. They are the people I love most in this world. My parents have given me the most selfless love, providing unwavering support at every step of my life. From the confusion I felt when I decided to pursue a Ph.D. in Singapore, to the pain I experienced when encountering challenges in my research, and the anxiety I faced when I needed to change advisors, they have always been there to help me overcome difficulties and firmly supported every decision I made. As for my grandmother, no words can truly capture the depth of her love for me. She may not fully understand what a Ph.D. degree is, but she is very happy that the grandson she raised is about to embark on a new chapter in life. This thesis is dedicated to them, to the ones I love most, and who love me most.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Background

1.1.1 Semantic Segmentation

Semantic segmentation refers to the task of classifying each pixel in an image. It is a fundamental task in computer vision and serves as the foundation for many other visual technologies. It is also widely used in various industrial applications. For example, image segmentation is essential for autonomous driving as it enables precise identification and differentiation of objects, such as vehicles, pedestrians, and road boundaries, ensuring accurate scene understanding of the car's nearby environments for making the safe decision. Medical diagnostics need to employ segmentation techniques for the precise identification and delineation of anatomical structures, tumors, and lesions, enabling accurate analysis and supporting early disease detection and treatment planning. Semantic segmentation is also widely used in industrial defect detection systems, as it enables precise localization and classification of low-quality regions on products, ensuring accurate quality control and improving manufacturing efficiency. Traditional semantic segmentation algorithms, such as the watershed algorithm [8] and threshold algorithm [3], are typically based on hand-crafted features followed by manually designed post-processing strategies. While these methods have achieved some success, hand-crafted features have inherent limitations, as it is challenging to ensure they are optimal for the segmentation task. As a result, traditional algorithms often fail to achieve satisfactory performance. In recent years, the rapid development of deep learning has led to significant breakthroughs in semantic segmentation. Convolutional networks, such as FCN [93] and UNet [117], have introduced an innovative paradigm for semantic segmentation based on deep neural networks. More recently, transformer networks, exemplified by MaskFormer [31], have further advanced semantic segmentation and achieved unprecedented accuracy.

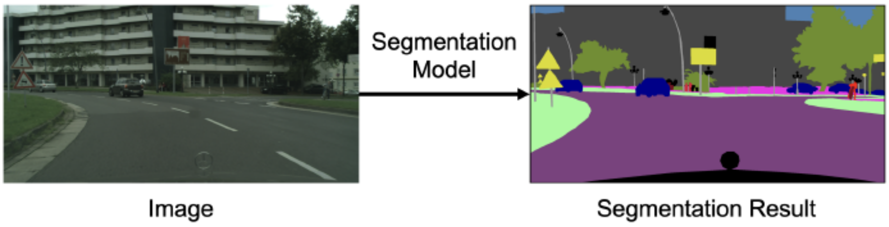


FIGURE 1.1: An example of semantic segmentation for self-driving images.

1.1.2 Multi-Stage Training

Just like other deep-learning-based methods, the success of DNN-based segmentation models heavily relies on training with large amounts of annotated data. Consequently, widely used segmentation datasets for academic research, such as Cityscapes [35], ADE20K [180], and Pascal VOC [39], typically contain tens or even hundreds of thousands of labeled images. Extensive training on these large datasets is crucial for DNN-based models to develop powerful feature extraction and segmentation capabilities, enabling them to perform robust and high-performance inference on diverse test images. However, in real-world applications, it is often infeasible to train models in a single step using complete, large-scale datasets. This is because training data is typically not available all at once but is instead provided incrementally in batches due to the ongoing and dynamic nature of data generation process. For example, autonomous vehicles continuously gather new data under diverse conditions such as fluctuating traffic volumes, vehicle behaviors, weather, and road usage, necessitating ongoing updates to segmentation models to adapt to an expanding range of traffic scenarios. Additionally, evolving business requirements can also drive the need for incremental data. For example, an e-commerce platform might expand from selling clothing to electronics, requiring new data to train models to recognize and segment electronic products. Moreover, the costs associated with data collection and annotation also enhances the necessity for multi-stage training. A notable example is in medical imaging analysis, where obtaining and annotating extensive sets of medical images, like MRI or CT scans, requires significant time and expense due to the need for annotation from radiology experts. Consequently, development teams, constrained by budget, resources, and time, may only be able to acquire and annotate small portions of data at a time, training the model in several stages to gradually optimize performance. These practical constraints in resources and the necessity for business scalability result in data often being provided incrementally in many real-world applications, necessitating multi-stage model training to continuously integrate new data. Developing effective and efficient image segmentation models for these multi-data and multi-training-stage scenarios is an important research problem.

1.1.3 Challenge

Compared to the single-stage training mode where all data is accessed and trained simultaneously, the aforementioned multi-stage approach presents two key challenges. First, since data is provided incrementally, the dataset is often incomplete in the early phases, making the amount of data available for training in these stages very limited. This scarcity of data can lead to poor model performance and inadequate generalization. However, there is often a pressing need to train and deploy the model at these early stages to meet urgent business requirements or to provide preliminary functionalities, which poses significant challenges. Second, when new batches of data are acquired and incorporated, the model needs to be trained on it to improve its ability to recognize and segment new categories in the new data. However, traditional training methods may cause the model to forget previously learned information from older data after learning new ones—a phenomenon known as "catastrophic forgetting." This

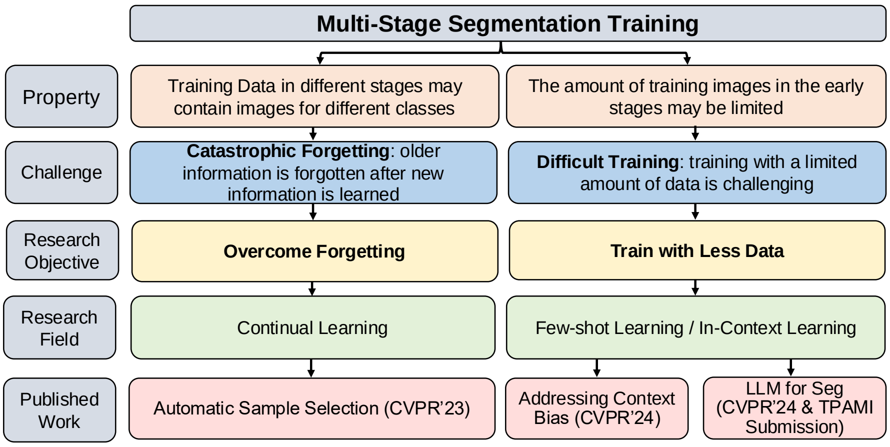

Multi-Stage Segmentation Training
Property
Training Data in different stages may contain images for different classes
The amount of training images in the early stages may be limited
Challenge
Catastrophic Forgetting: older information is forgotten after new information is learned
Difficult Training: training with a limited amount of data is challenging
Research Objective
Overcome Forgetting
Train with Less Data
Research Field
Continual Learning
Few-shot Learning / In-Context Learning
Published Work
Automatic Sample Selection (CVPR'23)
Addressing Context Bias (CVPR'24)
LLM for Seg (CVPR'24 & TPAMI Submission)

FIGURE 1.2: An overall structure of this thesis.

forgetting can affect the model's performance on older classes, compromising its reliability and stability. Given these challenges, we argue that it is critical to develop data-efficient training methods that can work with small sample sizes, and to design approaches that can preserve the model's memory of previously learned information after training on new samples of new classes. These techniques are essential for ensuring the high performance of the aforementioned multi-stage segmentation methods.

1.2 Motivation

Based on the above discussion, this thesis focuses on two research directions: (1) **few-shot segmentation (FSS)** [188, 182, 189, 161, 159], which aims to enable the model to grasp the ability to segment a class with very few or even just one training sample, and (2) **continual semantic segmentation (CSS)** [186, 192], which addresses the challenge of preventing the model from forgetting its ability to segment old classes after learning new ones. These two research directions respectively aim to address the aforementioned two issues: the low performance resulting from insufficient training data, and the problem of catastrophic forgetting where previously learned knowledge is often lost.

Several past methods have been proposed to address few-shot segmentation and continual semantic segmentation. However, these methods frequently exhibit significant limitations that lead to unsatisfactory performance. In the following sections, we respectively discuss the issues with existing few-shot and continual semantic segmentation methods, which are the motivations of our novel approaches proposed in this thesis.

1.2.1 Few-shot Segmentation

Traditional approaches in few-shot segmentation typically employ a support-feature-guided mechanism, where class features are extracted from a few labeled images, named as support images, to aid in segmenting an unlabeled image, known as the query image. However, we have identified two primary issues with this mechanism. First, there is often a lack of similarity or alignment in class features between support and query images. This misalignment can occur as background elements within an image may affect its foreground class features due to the large receptive field of the backbone network. Consequently, varying backgrounds in the support and query images can lead to different influences, resulting in their divergent class features that impairs the effectiveness of the support-feature-guidance paradigm. A similar problem has been reported in other research domains such as person re-identification [128], but to the best of our knowledge, it has not been explicitly emphasized and addressed by researchers in the task of few-shot segmentation. Consequently, it remains a critical yet unresolved challenge that requires attention. Secondly, segmenting the query image based solely on features from a limited set of support images presents challenges, since the few-shot samples can only provide a narrow, incomplete, and possibly biased set of information. Consequently, frameworks that depend exclusively on such restricted data are inherently constrained by informational limitations, making it difficult to achieve sufficiently high accuracy. Therefore, the further advancement of few-shot segmentation urgently requires an entirely new framework, which should be capable of utilizing richer and more comprehensive information, thereby breaking through the existing framework's bottlenecks to reach better results.

1.2.2 Continual Semantic Segmentation

For the task of continual semantic segmentation, which aims to adapt a previously learned model to accommodate newly added categories without forgetting old ones, a key challenge is identified and needs to be addressed. A straightforward approach to implementing continual semantic segmentation is to directly incorporate the samples of all past categories into the new training stages. However, due to privacy concerns and storage limitations, this approach is often impractical. A feasible solution, known as exemplar replay, involves storing only a small subset of samples from previous classes in a memory, which are then used during future training stages to improve the model's ability to handle old classes and reduce catastrophic forgetting. However, the capacity of this memory is typically limited, allowing only a few selected samples to be stored. Therefore, the careful selection of the most suitable samples for replay becomes crucial, and is a key focus of this thesis. While a lot of past studies have proposed various strategies for sample selection in continual semantic segmentation, such approaches have inherent limitations because most of them are manually designed and consider very few influence factors. However, the impact of a replay sample on mitigating catastrophic forgetting can be influenced by numerous factors, thus manually designing an optimal strategy that effectively accounts for these complex factors is very challenging and nearly impossible. Consequently, we argue that a more intelligent approach to memory sample selection is necessary—one that can consider a broader spectrum of factors and their intricate interdependencies, which remains an open problem and is crucial for the development of continual semantic segmentation.

1.3 Thesis Outline

This thesis is organized as follows:

- Chapter 2 introduces the previous work relevant to this thesis, including previous research about image segmentation, few-shot image segmentation, large language models for segmentation, and continual semantic segmentation.
- Chapter 3 presents our work of addressing background context bias in few-shot segmentation through iterative modulation. In this chapter, we first investigate the background context bias problem in FSS, which we find is a critical yet unresolved issue. To address this problem, we introduce a novel iterative approach, with each iteration containing a query prediction step followed by an innovatively proposed support modulation step and information cleansing step.
- Chapter 4 presents our work of leveraging large language models (LLMs) to address few-shot segmentation. In this chapter, we propose the first LLM framework specifically designed for few-shot segmentation, incorporating several novel components including a fine-grained instruction and a pseudo-sample-based training method to significantly enhance the LLM's capability for this task.
- Chapter 5 presents our work of a novel automatic sample selection method for continual semantic segmentation. In this chapter, we innovatively train an agent network using reinforcement learning to select appropriate samples for replay training in continual semantic segmentation tasks. A task-tailored state representation and action space are designed for the reinforcement learning mechanism in our method, making it better adapt to this task.
- Chapter 6 summarizes the contributions of this thesis and discusses potential future research directions.

1.4 Contributions

- We analyze the background context bias problem in few-shot segmentation for the first time, and introduce a novel iterative approach to address this critical problem. Each iteration of our method involves a query prediction step for query segmentation, a support modulation step to enhance the guidance effectiveness of support features, and an information cleansing step to prevent the accumulation of noisy information. Extensive experiments on multiple datasets demonstrate the high effectiveness of our method for few-shot segmentation.
- We propose LLaFS++, the first framework to address few-shot segmentation using large language models. In this framework, we introduce various innovative designs to make better use of LLMs in few-shot segmentation, including a task-tailored instruction, a fine-grained in-context instruction serving as multi-modal guidance, a pseudo-sample-based curriculum pretraining mechanism, and a novel inference method to mitigate prediction mistakes. We conduct comprehensive experiments and the results demonstrate that our approach achieves state-of-the-art performance with significant advantages. Furthermore, the applicability of our method is assessed across various tasks beyond few-shot segmentation such as few-shot object detection. The excellent performance across diverse tasks further demonstrates the generalization of LLaFS++.

- We formulate the sample selection operation of continual semantic segmentation as a Markov Decision Process, and introduce a novel and effective automatic paradigm for replay sample selection in continual semantic segmentation through reinforcement learning. This paradigm is enhanced by our proposed novel state representations containing multiple factors that can guide the selection decision, and a dual-stage action space to select samples and boost their replay effectiveness. Extensive experiments demonstrate that our automatic paradigm for sample replay can effectively alleviate the catastrophic forgetting issue in continual semantic segmentation and achieve state-of-the-art performance.

Chapter 2

Literature Review

2.1 Image Segmentation

Image segmentation is a fundamental task in the field of computer vision. Over the last decade, deep learning has brought significant advancements to this field, with techniques based on neural networks showcasing outstanding achievements across various sub-domains, such as semantic segmentation [19, 156, 187, 30, 24, 179, 76, 4, 183, 145, 28, 92, 190, 170, 60, 27], instance segmentation [51, 73, 23, 191, 17, 67, 144, 155, 60, 25], and panoptic segmentation [129, 68, 32, 21, 29, 100, 80].

2.1.1 CNN-Based Methods

Convolution neural networks (CNN) are the first to drive advancements in image segmentation during the deep learning era. A notable example of CNN-based segmentation methods is the Deeplab series [20, 18, 19], which employs atrous convolution to increase the size of the receptive field, enabling richer semantic information to be captured and preserving the high-resolution of feature maps to avoid boundary blurring. Similarly, PSPNet [179] employs a pyramid pooling module to aggregate global and local features at different scales, enhancing segmentation accuracy especially in complex scenes. UNet [117] is a another classical CNN architecture for the image segmentation task, particularly in biomedical imaging. It employs an effective and efficient encoder-decoder structure with the "U" shape, where the encoder captures context by downsampling the input image, while the decoder reconstructs the image with fine-grained details through upsampling. Skip connections link the encoder and decoder layers, allowing the model to retain important spatial information. STLNet [187] proposes the first module to extract texture features for enhancing segmentation performance. Some CNN-based methods focus on reducing the computational cost of segmentation models to improve efficiency. For example, BiseNet [166] and its extension [40] propose efficient semantic segmentation models designed to balance accuracy and speed. It employs a dual-branch architecture, combining a spatial path for high-resolution spatial details and a context path for rich semantic information, enabling real-time segmentation while maintaining strong performance.

2.1.2 Transformer-Based Methods

More recently, transformer-based approaches [156, 31, 30, 59, 184, 111] have pushed the boundaries of segmentation performance even further. For instance, SegFormer

[156] introduces an innovative pipeline that combines a hierarchical transformer encoder with an MLP-based decoder. MaskFormer [30] propose a transformer-based model designed for unified image segmentation, capable of handling instance, semantic, and panoptic segmentation within a single framework. Instead of predicting pixel labels directly, MaskFormer employs a novel paradigm that generates a set of mask embeddings to represent different regions in the image and achieves enhanced performance. Mask2Former [31] further improves MaskFormer, employing masked attention to achieve faster convergence by constraining cross-attention within predicted mask regions. The segment anything model (SAM) [69] introduces a general method that is trained on a large and diverse dataset, allowing users to segment objects in an image by providing different types of prompts, such as points, boxes, or masks. SAM2 [114] further extends the general segmentation capabilities of SAM from static images to video data. It leverages temporal information to achieve consistent and accurate segmentation across video frames, significantly enhancing SAM's applicability to dynamic scenes. Despite its effectiveness, SAM has been observed to perform suboptimally in certain domains, such as medical and remote sensing images. To address this issue, SAM-adapter [26, 27] leverages transformer-based adapter techniques to efficiently finetune SAM for specific tasks or specific image domains.

2.1.3 Challenge

Existing segmentation methods typically require training on large datasets and often suffer from catastrophic forgetting during multi-stage training, where previously learned information is often lost after new knowledge is acquired. To address these issues, in this thesis, unlike the previously discussed conventional methods, we focus on the tasks of few-shot segmentation and continual semantic segmentation, which enable the segmentation of a query image for a novel class using only a very small number of annotated support images, and build the capability to retain performance on old classes after training on new data and classes. This research avoids the high costs associated with extensive data collection and annotation, allowing for high-performance segmentation models trained with only a small amount of data. In addition, it mitigates the issue of catastrophic forgetting, enabling the model to retain its ability to handle data and classes from all stages after multi-stage training.

2.2 Few-shot Segmentation

2.2.1 Conventional Few-shot Segmentation Methods

Few-shot segmentation (FSS) [140, 163, 85, 64, 89, 10, 72, 106, 71, 33, 131, 94, 182, 159] has gained significant attention in recent years due to its ability to work well with only limited data, which is highly practical in real-world applications. [119] proposes the pioneering method in this field, where a feature is extracted from the labeled support images to generate a head that is then used to segment unlabeled query images. Building upon this framework, many existing methods [41, 72, 163, 103, 74, 57, 33, 169, 146] adopt a prototype-guided strategy. These techniques employ masked average pooling (MAP) to derive global or local average prototypes from support image features, which then guide the segmentation of query images through various approaches such as feature fusion [74, 72, 90], distance measurement [98, 52], or attention-based mechanisms [109]. To avoid information loss caused by the prototype generation process, some more recent methods [175, 174, 137, 56, 157, 148] do not compress features into prototypes but instead retain the complete feature maps for per-pixel processing. For example, [160] proposes a self-calibrated cross-attention network with pixel-wise correlation extraction to solve the background mismatch and foreground-background entanglement issues. Other approaches [98, 121, 52, 63] try to extract the relationship between support and query image features in even greater detail by capturing 4D correlations. For instance, [98] proposes a hypercorrelation squeeze network that leverages efficient 4D convolutions to extract multi-level feature correlations.

While these methods have achieved some success, they still have two significant issues. First, the previous methods typically suffer from misalignment of foreground features caused by the different backgrounds. In this thesis, we propose the first method to address this problem by extracting background context to modulate support foreground features. Some other few-shot segmentation methods [41, 163, 91] also take background into consideration. However, they either employ background prototypes to eliminate background regions in query predictions [91], or segment interfering objects belonging to other categories in the background [163], without further considering how background context affects foreground features and the resultant issue of misaligned features. In few-shot segmentation, we design the first method to address this problem. Second, the most existing methods can only leverage a limited amount of information extracted from a very small number of support images. Such a constraint may lead to suboptimal performance and decreased robustness. In this thesis, we are the first to employ large language models (LLMs) to achieve few-shot segmentation by using our carefully designed instructions, which offer a more comprehensive and effective multimodal guidance system. This novel method introduces a brand-new paradigm for few-shot segmentation, exploring new possibilities for using LLMs in this research domain.

2.2.2 Few-Shot Segmentation with Foundation Models

Some recent studies explore the use of foundation models to enhance the performance of few-shot segmentation. Several approaches leverage the powerful general segmentation capabilities of the Segment Anything Model (SAM) to improve few-shot segmentation tasks. For example, VRP-SAM [127] extends SAM [69] to few-shot segmentation by introducing a visual reference prompt encoder, allowing annotated reference images to serve as prompts for segmentation. Additionally, [171] applies graph analysis to strengthen SAM's few-shot segmentation performance by dynamically selecting point prompts through a positive-negative alignment module and incorporating a point-mask clustering module to align mask and point granularities, further boosting segmentation accuracy. Other methods explore to use language models for few-shot segmentation. For example, [164] employs word embeddings from Word2Vec as a more comprehensive and general source of class information to aid in segmentation. [139] leverages the semantic alignment capabilities of the CLIP model [112] to generate more accurate prior information for few-shot segmentation tasks. However, despite some improvements, these approaches remain limited by the relatively weak capabilities of small language models and lacks an in-depth exploration of how to combine textual information with support image information more effectively for the improved guidance. In contrast, this thesis pioneers the exploration of large language models (LLMs) in the task of few-shot segmentation, which is more effective than the previously used small language models such as CLIP. Moreover, we propose a better approach to integrating textual information from language models with visual information from few-shot annotated images, which brings significant performance enhancement.

2.3 Large Language Models and their Applications in Image Segmentation

2.3.1 Large Language Models

The advent of large language models (LLMs) such as GPT [11] and Llama [134] has marked the beginning of a new era in artificial intelligence. Thanks to their significantly increased model parameters and training data, these LLMs contain rich prior knowledge and can be efficiently finetuned for specific tasks or application requirements through methods such as prompts [86], adapters [54] and LoRA [55]. Recently, researchers have started exploring visual large language models [75, 83, 141, 126, 5, 181, 62, 185] to establish a unified framework for multimodal data processing, aiming to override the restriction of LLMs being solely applicable to language data. For example, BLIP-2 [75] proposes a large vision-language model designed to bridge the gap between image and text understanding. It employs a Q-former to condense image features into visual tokens, and leverages a two-stage training process that aligns visual and textual representations, enabling efficient cross-modal learning. LLaVA [83] introduces an alternative, more streamlined framework for large vision-language models, where the Q-Former in BLIP-2 is omitted, allowing visual features to be directly fed into the LLM without the need for complex transformer networks for further processing. This simplified architecture improves efficiency and facilitates joint image-text understanding. MiniGPT-4 [181] proposes a more data-efficient training approach for large vision-language models, in which only 3500 image-text pairs are used to perform instruction tuning on the LLM, enabling significantly enhanced multimodal understanding capability with only very minimal data requirements and computational cost. However, despite significant success, most of these methods can only handle image-level understanding tasks, such as image captioning, visual question answering, etc., but are incapable of performing the more fine-grained segmentation tasks.

2.3.2 LLMs for Image Segmentation

Some more recent research [70, 116, 178, 141, 168, 113] has begun exploring how to extend the capabilities of LLMs to the field of image segmentation. For example, LISA [70] proposes a large language instructed segmentation assistant to produce segmentation masks by incorporating an additional segmentation token into the existing vocabulary. GSVA [152] addresses the shortcomings of LISA by employing multiple [SEG] tokens for multi-target segmentation and a [REJ] token to reject empty targets. [116] proposes an LLM-based segmentation model with a lightweight pixel decoder and a comprehensive segmentation codebook. [178] introduces a novel framework to handle unified segmentation by first generating mask proposals and then using LLMs to classify them. In our approach, we follow the strategy proposed by VisionLLM [141], which empowers LLMs to produce segmentation masks by generating the vertices of enclosing polygons. While all of the above methods are capable of performing image segmentation, our method differs from them significantly since none of them are designed specifically for few-shot segmentation. In contrast, in this thesis, we propose a novel framework namely LLaFS++, which is the first LLM-based few-shot segmentation framework with several novel and task-tailored designs including: *(a) LLM's inputs.* Two novel instructions serving as LLM's inputs are proposed to extract rich information from the annotated support images in few-shot scenarios. *(b) Training data.* A novel method for synthesizing pseudo samples is proposed to solve the insufficient training data issue in few-shot segmentation. *(c) Optimization approaches.* A curriculum learning strategy is implemented to overcome slow convergence challenges. Incorporating these novel designs, LLaFS++ constitutes a brand-new framework that can effectively leverage information from both the annotated images and language priors to achieve high-quality few-shot segmentation.

2.4 Continual Semantic Segmentation

2.4.1 Continual Semantic Segmentation Methods

Continual semantic segmentation overcomes the limitations of traditional segmentation algorithms by allowing different classes to be learned at different stages. To address the biggest challenge in this task—catastrophic forgetting—some methods [15, 96, 22, 110, 105, 138, 167] adopt knowledge distillation techniques that enable the model to review knowledge from previous stages during the learning process of new stages. For example, MIB [15] addresses the challenge of semantic shift in the background class by proposing a novel weight initialization method and distillation loss. SDR [96] proposes a novel distillation-based preservation technique based on prototype matching, contrastive learning, and feature sparsity. CSW-KD [110] selectively revises the knowledge of old classes that are likely to be forgotten through distillation by focusing on those that are visually similar to the new classes. Other methods [37, 177, 82, 133] address the forgetting problem by annotating the new stage's data with pseudo-labels for the old stage's classes. For example, PLOP [37] employs a pseudo-labeling mechanism based on prediction confidence to improve continual learning performance. CoinSeg [177] introduces a contrastive-learning-based distillation to enhance both inter-class and intra-class representations with the guidance of pseudo-labels. Some methods [172, 153, 34] use weight fusion to combine model parameters from both new and old stages to retain previous knowledge. For example, EWF [153] fuses models from different stages by taking the weighted average of their corresponding parameters. Cs2K [34] further improves the model fusion method by selectively integrating different model parameters through weight-guided selective consolidation. Another effective approach to addressing catastrophic forgetting is to employ a portion of past data for replay. For example, some methods [16, 162] use a memory buffer to store replay exemplars, but the samples are selected either randomly or based on heuristic rules. [95] derives richer replay exemplars through a generative adversarial network, but this approach comes with high computational costs and requires additional web-crawled images. In this thesis, we propose a new method that also adopts a memory replay mechanism but with a different and entirely new pipeline, in which we introduce a novel automatic sample selection mechanism and propose a new, effective replay training method to utilize memory samples better. Some more recent works [120, 107, 14, 46] are designed based on transformer networks. We have also conducted experiments to compare with these methods and achieved superior results.

2.4.2 Memory Sample Selection

As an important approach to addressing catastrophic forgetting in continual learning, the effectiveness of replay-based methods relies on the careful selection of memory samples. Most of the previous selection methods are designed based on hand-crafted heuristic rules. For example, some methods [1, 165, 65, 7] consider sample diversity as a key factor in determining replay effectiveness and select samples accordingly. [122] uses adversarial Shapley value for sample selection to preserve latent decision boundaries for previously observed classes. [115] proposes selecting a fixed number of representative samples that best capture the feature distribution of each class. Despite some success, such hand-crafted methods are difficult to be optimal due to the complex interplay between different factors that can affect selection performance. In contrast, our method proposed in this thesis explores a novel direction by enabling the selection policy to be automatically learned through a carefully designed RL mechanism, which achieves better performance than previous selection methods as demonstrated by the experimental results presented in this thesis.

2.4.3 Reinforcement Learning

Reinforcement learning (RL) is a technique that has achieved remarkable success in many decision-making tasks, such as game intelligence [124], robot control [125], and large language models [6]. It has also been applied to computer vision in various areas such as active learning [45], pose estimation [49], model compression [2], and person re-identification [151]. In this thesis, we introduce a novel and effective automatic paradigm for replay sample selection in continual semantic segmentation through reinforcement learning. A previous work related to our method is [88], which uses RL for exemplar length management but operates with a completely different mechanism from ours. Instead of employing RL to control class-level memory length while still relying on a random selection process as in [88], our method is end-to-end and can directly select specific samples in a fully automated, single step. In addition to the selection method, we also propose a novel approach to better utilize the samples selected by the RL policy. By incorporating these designs, our method in this thesis introduces a brand-new RL-based replay pipeline for continual semantic segmentation.

Chapter 3

Addressing Background Context Bias in Few-Shot Segmentation through Iterative Modulation

In this chapter, we propose a novel network to address the problem of background bias in few-shot segmentation as illustrated in Chapter 1. Specifically, existing few-shot segmentation methods usually extract foreground prototypes from support images to guide query image segmentation. However, different background contexts of support and query images can cause their foreground features to be misaligned. This phenomenon, known as background context bias, can hinder the effectiveness of support prototypes in guiding query image segmentation. In this work, we propose a novel framework with an iterative structure to address this problem. In each iteration of the framework, we first generate a query prediction based on a support foreground feature. Next, we extract background context from the query image to modulate the support foreground feature, thus eliminating the foreground feature misalignment caused by the different backgrounds. After that, we design a confidence-biased attention to eliminate noise and cleanse information. By integrating these components through an iterative structure, we create a novel network that can leverage the synergies between different modules to improve their performance in a mutually reinforcing manner. Through these carefully designed components and structures, our network can effectively eliminate background context bias in few-shot segmentation, thus achieving outstanding performance. We conduct extensive experiments on the PASCAL-5^i and COCO-20^i datasets and achieve state-of-the-art (SOTA) results, which demonstrate the effectiveness of our approach.

3.1 Introduction

Image segmentation [20, 18, 42, 187, 19, 179, 193, 31, 156, 145, 183, 43, 26] is a crucial task in computer vision. In recent years, significant progresses have been made in this field, which are primarily attributed to the development of deep learning models [93, 187, 19, 179] trained on large-scale datasets [35, 180]. However, obtaining sufficient labeled data to train a segmentation model is time-consuming and labor-intensive, since it usually takes more than 10 minutes to annotate only one image for getting its ground truth label. To address this issue, Few-Shot Segmentation (FSS) has been proposed as

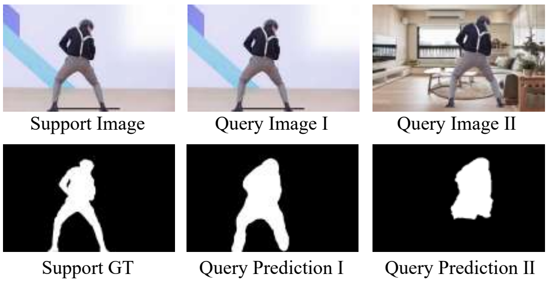


FIGURE 3.1: An example of background context bias in few-shot segmentation. When the query image shares the same background as the support image (Query Image I), the segmentation is high-quality; but when the query image has a different background (Query Image II), the segmentation is undesirable.

an alternative solution, which aims to segment a class using a very small number of annotated images, thus reducing the need for costly data labeling.

Currently, the prevailing methods for few-shot segmentation [10, 123, 140, 85, 89, 163, 188] typically use the meta-learning and episodic training strategies, in which the model is trained to segment a query image based on a few support images and their ground truth maps in the target class. To extract useful information that can represent the general properties of a class, these methods usually extract one [72, 41, 103] or a few [74, 163] prototypes from the foreground region of the support images. These prototypes are then used to segment query images based on feature concatenation [72, 74] or distance calculation [98, 52]. For this paradigm to be successful, it is necessary to assume that support and query images posses the similar or aligned foreground features. Only when this assumption holds true, the support prototypes can capture the query foreground properties accurately, thus enabling them to guide the segmentation process effectively.

However, we contend that this assumption may not always be true, since different backgrounds between support and query images may cause misalignment of their foreground features. Specifically, the prototypes for few-shot segmentation are typically derived from CNN or transformer features, which have a large receptive field that allows background context to be transmitted into foreground. As a result, different backgrounds can affect support and query foreground features differently, leading to feature misalignment that limits support prototypes' ability to guide query image segmentation.

Figure 3.1 shows an example of this problem. In our experiment, even though the support and query images contain the identical foreground object, the query image's segmentation results become undesired when the object is placed in different environments in support and query images. A similar problem of background context bias has been reported in other domains such as person ReID [128], but to the best of our knowledge, it has not been explicitly emphasized and addressed by researchers in the task of few-shot segmentation. Consequently, it remains a critical yet unresolved challenge that requires attention.

To mitigate the research gap, in this chapter, we propose a novel network for few-shot segmentation, which can effectively alleviate background context bias and demonstrate improved performance. Specifically, in our method, we employ query context to modulate support features. Through this modulation, query background information that influences the query foreground can be incorporated into the support prototypes, aligning them more closely with the query foreground features and therefore improving their ability to guide query image segmentation. To ensure the effectiveness of this modulation, we further investigate how to extract background context with stronger representation ability. Concretely, to ensure that the extracted context can adequately capture the influence of the background on the foreground, we model the input-to-output evolution of query foreground features within a deep network, and utilize this evolution as a basis for extracting context that is used in the modulation process. We also propose an information cleansing method to prevent noise from accumulating during the modulation process. By integrating these components through an iterative structure, we create a novel network that can leverage the synergies between different modules to improve their performance in a mutually reinforcing manner. Through these carefully designed components and structures, our network can effectively eliminate background context bias in few-shot segmentation, thus achieving outstanding performance as demonstrated by experiments.

We perform extensive experiments on common datasets including PASCAL-5^i and COCO-20^i and report state-of-the-art (SOTA) performance. Our contributions can be summarized as follows: (1) Firstly, we investigate the background context bias problem in FSS, which we find is a critical yet unresolved issue. (2) Secondly, we introduce an iterative approach to address context bias. Each iteration of our method involves a query prediction step for query segmentation, a support modulation step to enhance the guidance effectiveness of support features, and an information cleansing step to prevent the accumulation of noisy information. (3) Thirdly, our proposed approach achieves state-of-the-art (SOTA) performance on few-shot segmentation.

3.2 Task Definition

FSS seeks to perform segmentation given only a small number of annotated images. The target is to train an FSS model on the training set $\mathcal{D}_{train}$ and evaluate it on the test set $\mathcal{D}_{test}$, where the two datasets are disjoint with respect to object classes. To achieve this, we follow previous works [41, 74] under the meta-learning setting and execute episodic training to optimize our FSS model. Specifically, each set is partitioned into multiple episodes, where each episode consists of a support set $\{I_s^k, G_s^k\}_{k=1}^K$ and a query set $\{I_q, G_q\}$, with $I^* \in \mathbb{R}^{H\times W\times 3}$ and $G^* \in \mathbb{R}^{H\times W}$ representing the image and the corresponding ground truth respectively. G^* is a binary mask indicating pixels within the target class, which, for convenience, are denoted as **foreground** in the subsequent sections, and other pixels outside the target class are denoted as **background**. During training, we iteratively sample an episodic from $\mathcal{D}_{train}$ to train a model that predicts G_q based on $\{I_s^k, G_s^k\}_{k=1}^K$ and I_q. Once the training is completed, the model is evaluated on $\mathcal{D}_{test}$. For the convenience of introduction, in the following sections, we describe our method under the 1-shot setting where only one support image is available (K=1). In Section 3.3.6, we elaborate on how to extend the method to the K-shot setting.

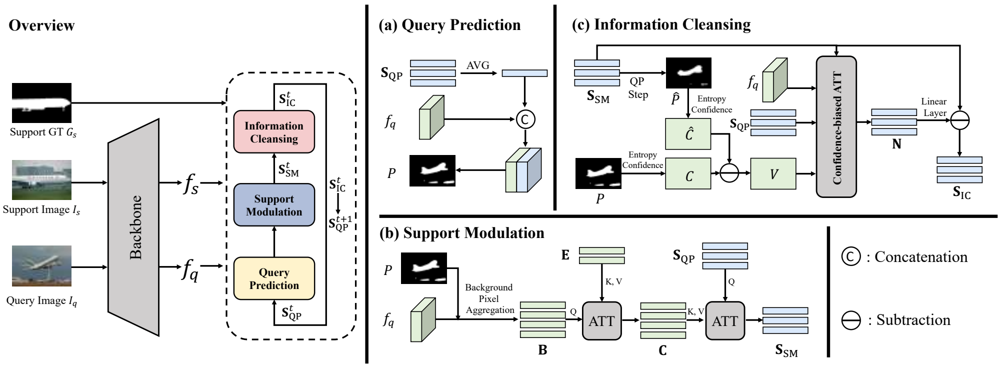

Overview
Support GT G_s
Support Image I_s
Query Image I_q
Backbone
f_s
f_q
$\mathbf{S}_{\mathrm{IC}}^t$
Information Cleansing
$\mathbf{S}_{\mathrm{SM}}^t$
Support Modulation
Query Prediction
$\mathbf{S}_{\mathrm{QP}}^t$
$\mathbf{S}_{\mathrm{IC}}^t$
$\mathbf{S}_{\mathrm{QP}}^{t+1}$
(a) Query Prediction
$\mathbf{S}_{\mathrm{QP}}$
AVG
f_q
P
(c) Information Cleansing
$\mathbf{S}_{\mathrm{SM}}$
QP Step
$\hat{P}$
Entropy Confidence
$\hat{C}$
f_q
$\mathbf{S}_{\mathrm{QP}}$
Confidence-biased ATT
Linear Layer
N
$\mathbf{S}_{\mathrm{IC}}$
P
Entropy Confidence
C
V
(b) Support Modulation
P
f_q
Background Pixel Aggregation
E
K, V
Q
ATT
B
C
$\mathbf{S}_{\mathrm{QP}}$
Q
K, V
ATT
$\mathbf{S}_{\mathrm{SM}}$
: Concatenation
: Subtraction

FIGURE 3.2: Overall structure and different components of our network. First, the backbone generates f_s and f_q for the support image and query image respectively. Next, an iterative structure is designed to fully utilize the features for segmentation, which consists of T iterations, with each iteration containing three successive steps: (a) Query Prediction, (b) Support Modulation, and (c) Information Cleansing. The output from the Information Cleansing step is input into the Query Prediction step of the subsequent iteration. Segmentation result from the Query Prediction step in the last iteration serves as the final prediction at the inference stage. The structure of the confidence-biased ATT in (c) is shown in Figure 3.4.

3.3 Method

3.3.1 Overview

Figure 3.2 provides an overview of our method. First, a backbone network produces features f_s and f_q for support and query images respectively. Next, an iterative structure is designed to fully utilize the features for segmentation, which consists of T iterations, with each iteration containing three successive steps: Query Prediction (QP), Support Modulation (SM), and Information Cleansing (IC). Considering the t-th iteration of the structure, the operation is performed as follows. Firstly, in the QP step, query images are segmented under the guidance of a support foreground feature $\mathbf{S}_{\mathrm{QP}}^t$. Then, in the SM step, query context is extracted and used to enhance $\mathbf{S}_{\mathrm{QP}}^t$'s effectiveness for guiding query segmentation. The enhanced $\mathbf{S}_{\mathrm{SM}}^t$ is obtained in this step. After that, in the IC step, a biased attention adjusts $\mathbf{S}_{\mathrm{SM}}^t$ from the SM step to cleanse information, getting the processed $\mathbf{S}_{\mathrm{IC}}^t$. The three steps are performed iteratively, where the updated $\mathbf{S}_{\mathrm{IC}}^t$ from the previous iteration guides the QP step in the subsequent iteration ($\mathbf{S}_{\mathrm{QP}}^{t+1} \leftarrow \mathbf{S}_{\mathrm{IC}}^t$). Query segmentation result from the last iteration serves as the final prediction at the inference stage.

The motivation to design an iterative manner is based on our observation that each of the three steps can have an influence on one another, so by successively updating the feature at each step, the network can be forced to refine itself towards an optimal solution in a recurrent optimization manner. Specifically, in every iteration, an improved support foreground feature $\mathbf{S}_{\mathrm{QP}}^t$ ($\mathbf{S}_{\mathrm{IC}}^{t-1}$), which is modulated from the SM and IC steps of the previous iteration, can guide the generation of a more accurate query prediction in the QP step. By using this query prediction with higher accuracy, the subsequent

SM and IC steps can perform better, thus resulting in an improved $\mathbf{S}_{\mathrm{IC}}^{t}$. $\mathbf{S}_{\mathrm{IC}}^{t}$ is further utilized for query prediction in the following iteration ($\mathbf{S}_{\mathrm{QP}}^{t+1} \leftarrow \mathbf{S}_{\mathrm{IC}}^{t}$). In this way, a recurrent optimization scheme is created that can utilize the iterative structure to continuously refine the query prediction results. In the subsequent sections, we describe each step of our method in detail.

3.3.2 Query Prediction Step

As shown in Figure 3.2 (a), in this step, query images are segmented under the guidance of a support foreground feature $\mathbf{S}_{\mathrm{QP}}$. In the first iteration, $\mathbf{S}_{\mathrm{QP}}$ is initialized using the support backbone features f_s. Specifically, each foreground pixel's feature in f_s is treated as a token, and these tokens are aggregated to obtain $\mathbf{S}_{\mathrm{QP}}$. As a combination of features from all foreground pixels, $\mathbf{S}_{\mathrm{QP}}$ can reflect the general properties of support foreground, making it possible to guide the segmentation of query foreground belonging to the same category. For the remaining iterations, $\mathbf{S}_{\mathrm{QP}}$ is updated as the output from the IC step of the previous iteration. Given $\mathbf{S}_{\mathrm{QP}}$ and the query backbone features f_q, query segmentation is carried out with the process as follows:

$$P = \phi_p \left(\mathrm{CAT}\left(\mathrm{AVG}\left(\mathbf{S}_{\mathrm{QP}}\right), f_q\right)\right), \tag{3.1}$$

where AVG denotes the average over all tokens in $\mathbf{S}_{\mathrm{QP}}$, CAT refers to the channel-wise concatenation, ϕ_p is a two-layered 1×1 convolutions that generates the prediction P.

3.3.3 Support Modulation Step

As a result of background context bias between support and query images, their foreground features are misaligned, which makes the QP step alone insufficient to ensure accurate query image segmentation. To address this problem, we propose a support modulation (SM) step, which uses background context from the query image to modulate $\mathbf{S}_{\mathrm{QP}}$, thereby minimizing the feature gap and increasing its effectiveness in guiding query image segmentation. Specifically, we denote the number of foreground pixels and background pixels in the query image by N_f and N_b, and as shown in Fig. 4.1 (b), we implement the SM step as follows. Firstly, we extract an evolution feature $\mathbf{E} \in \mathbb{R}^{N_f \times C}$ that reflects how the query foreground representation changes from input to output layers within the backbone network (details are illustrated below). Subsequently, we concatenate the features of all background pixels[1] in f_q to generate a background representation $\mathbf{B} \in \mathbb{R}^{N_b \times C}$. By correlating $\mathbf{E}$ with $\mathbf{B}$, we then capture background context $\mathbf{C}$ by:

$$\mathbf{C} = \mathrm{ATT}\left(\mathbf{Q}_{\mathbf{B}}, \mathbf{K}_{\mathbf{E}}, \mathbf{V}_{\mathbf{E}}\right), \tag{3.2}$$

where ATT is a cross-attention module in a QKV manner. Finally, we use query context to modulate $\mathbf{S}_{\mathrm{QP}}$ through another cross-attention:

$$\mathbf{S}_{\mathrm{SM}} = \mathbf{S}_{\mathrm{QP}} + \mathrm{ATT}\left(\mathbf{Q}_{\mathbf{S}_{\mathrm{QP}}}, \mathbf{K}_{\mathbf{C}}, \mathbf{V}_{\mathbf{C}}\right). \tag{3.3}$$

[1] To generate $\mathbf{B}$, the background region of the query image is estimated from the prediction P of the QP step.

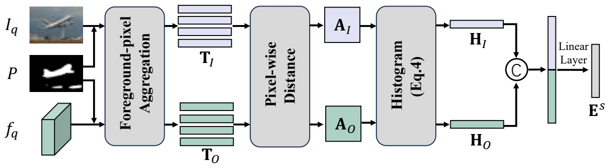


FIGURE 3.3: Generation of the structure-wise evolution feature $\mathbf{E}^s$.

The resulted $\mathbf{S}_{\text{SM}}$ is the output of the SM step. This step includes an innovative design that extracts background context via the evolution feature $\mathbf{E}$, which has not been used previously but has demonstrated success in our experiments. $\mathbf{E}$ is a feature that describes how foreground features change from the input layer to the output layer. Using it is motivated by the analysis of network inputs and outputs. Specifically, in a deep neural network, the input image is context-independent, so its foreground contains no background information; on the other hand, the output features are heavily influenced by long-range context due to the network's high-receptive-field property, so its foreground contains substantial background context. Consequently, by modeling the change from input to output, $\mathbf{E}$ can capture the influence of the background on the foreground within the network, thus enabling it to perform modulation effectively. To capture rich and multi-level information for the effective context extraction, we sum two types of evolution features to obtain $\mathbf{E}$, namely pixel-wise evolution feature $\mathbf{E}^p$ and structure-wise evolution feature $\mathbf{E}^s$ as follow:

Pixel-wise Evolution Feature

First, we extract $\mathbf{E}^p$, which provides the evolution information for each pixel in the query foreground. For this, we pass the input query image and its backbone output features through two individual 1×1 convolutions. Through this process, we get a context-independent input representation F_I and a context-influenced output representation F_O, respectively. After that, by identifying the query foreground region using P generated by the QP step, we generate foreground tokens $\mathbf{F}_I \in \mathbb{R}^{N_f \times C}$ and $\mathbf{F}_O \in \mathbb{R}^{N_f \times C}$, which aggregate the features of all foreground pixels in F_I and F_O, respectively. $\mathbf{F}_I$ and $\mathbf{F}_O$ are concatenated along the channel dimension and are passed through a two-layered MLP to produce a feature matrix $\mathbf{E}^p \in \mathbb{R}^{N_f \times C}$. In this way, by modeling the interaction between the context-independent input image and the context-influenced output feature, $\mathbf{E}^p$ extracts pixel-wise evolution features that can be used to model query context.

Structure-wise Evolution Feature

As shown in Figure 3.3, in addition to $\mathbf{E}^p$, motivated by the demonstrated importance of structural information in the segmentation task [61, 81], we further introduce a Structure-wise Evolution Feature denoted by $\mathbf{E}^s$. Firstly, We aggregate all foreground pixels from the query image I_q and its backbone output f_q, getting input tokens $\mathbf{T}_I \in \mathbb{R}^{N_f \times 3}$ and output tokens $\mathbf{T}_O \in \mathbb{R}^{N_f \times C}$ respectively. Next, we calculate pixel-wise distances in $\mathbf{T}_I$ and $\mathbf{T}_O$ to get affinity maps $\mathbf{A}_I \in \mathbb{R}^{N_f \times N_f}$ and $\mathbf{A}_O \in \mathbb{R}^{N_f \times N_f}$. Specifically, each item $\mathbf{A}_I^{i,j}$ on $\mathbf{A}_I$ is computed as the cosine similarity between $\mathbf{T}_I^i$ and $\mathbf{T}_I^j$, where $\mathbf{T}_I^i$ refers to the i-th pixel on $\mathbf{T}_I$. $\mathbf{A}_O$ is produced from $\mathbf{T}_O$ similarly. We flatten $\mathbf{A}_I$ and $\mathbf{A}_O$ to the shape $\mathbb{R}^{N_f^2}$, and then introduce histogram features $\mathbf{H}_I \in \mathbb{R}^L$ and $\mathbf{H}_O \in \mathbb{R}^L$ to capture statistical information from them. For this, the continuous range [0,1] is subdivided into L discrete bins $\{I_l\}_{l=1}^L$ with $I_l = [(l-1)/L, l/L]$, then each dimension $\mathbf{H}_I^l$ on $\mathbf{H}_I$ is calculated as the total number of dimensions in $\mathbf{A}_I$ with values falling into the interval I_l. Formally,

$$\mathbf{H}_I^l = \sum_{i=1}^{N_f^2} \mathbb{1}\left(\frac{l-1}{L} < \mathbf{A}_I^i < \frac{l}{L}\right),\ l \in [0, L-1]. \tag{3.4}$$

For $\mathbb{1}$ to be differentiable, we use a spire-shaped judge function. Specifically, to facilitate expression, we use a to represent $\frac{l-1}{L}$ and b to represent $\frac{l}{L}$. In this way, this spire-shape function can be formulated as:

$$\mathbb{1}\left(a < \mathbf{A}_l^i < b\right) = \begin{cases} 1 - \left|\mathbf{A}_I^i - \dfrac{a+b}{2}\right| & if \quad a < \mathbf{A}_I^i < b \\ 0 \quad else \end{cases} \tag{3.5}$$

By doing so, the gradients can be propagated successfully, allowing the network to be end-to-end trained in optimization. Using the same approach, we can get $\mathbf{H}_O$ from $\mathbf{A}_O$. Finally, we normalize and concatenate $\mathbf{H}_I$ and $\mathbf{H}_O$, followed by a two-layered MLP to get the Structure-wise Evolution Feature $\mathbf{E}^s \in \mathbb{R}^C$.

In the end, $\mathbf{E} \in \mathbb{R}^{N_f \times C}$ is generated by adding $\mathbf{E}^s \in \mathbb{R}^C$ to each pixel of the pixel-wise evolution feature $\mathbf{E}^p \in \mathbb{R}^{N_f \times C}$, which can then be used to modulate $\mathbf{S}$ through Eq. 3.2 and Eq. 3.3.

Discussion: Why to Use Affinity Maps and Histograms

To generate $\mathbf{E}^s$, we employ affinity maps followed by histograms to extract structural information. Using affinity maps is motivated by their strong ability to represent image structures [87, 72]. Histograms are used for two reasons. Firstly, histograms can capture structural information like contrast and smoothness [47], so they are helpful for representing the structure features of the foreground region at a particular network layer. Secondly, histograms facilitate the extraction of features from affinity maps $\mathbf{A}_I \in \mathbb{R}^{N_f \times N_f}$ and $\mathbf{A}_O \in \mathbb{R}^{N_f \times N_f}$, which are unfixed in size due to the variable number of foreground pixels (N_f) in different images. Using histograms, $\mathbf{A}_I$ and $\mathbf{A}_O$ can be converted into fixed-shaped representations, thus making it possible to extract evolution features from them through linear layers that require a fixed number of input and output channels.

3.3.4 Information Cleansing Step

To extract context, in the SM step, we use the prediction P from the QP step as a foreground-background indicator to create features like $\mathbf{B}$ in Eq.3.2. However, P is a coarse prediction with some incorrectly segmented pixels, so using it directly can introduce noise into these features. Consequently, through the processes of the SM step, this noise can be propagated to the output $\mathbf{S}_{\text{SM}}$, finally impeding its effectiveness

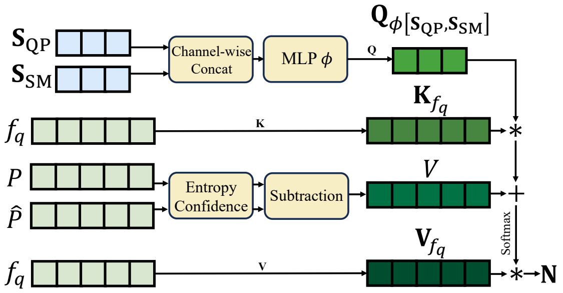


FIGURE 3.4: Illustration figure of the confidence-biased attention used in the Information Cleansing Step. P, $\hat{P}$ and f_q are flattened along the spatial dimension before the attention.

in guiding query prediction. To overcome this issue, we propose an additional step called Information Cleansing (IC), which removes the noise from $\mathbf{S}_{\text{SM}}$, thus producing a cleaner $\mathbf{S}_{\text{IC}}$ that can guide query image segmentation more effectively. The structure of this step is shown in Fig. 3.2 (c). Firstly, we introduce a confidence-biased attention as shown in Figure 3.4 to capture the accumulated noisy information, which is inspired by recent semi-supervised learning study [149] demonstrating that noisy information can lead to lower prediction certainty. Specifically, we replace $\mathbf{S}_{\text{QP}}$ in Eq. 3.1 with $\mathbf{S}_{\text{SM}}$ and perform the QP step. By doing so, in addition to prediction P from $\mathbf{S}_{\text{QP}}$, we get an intermediate prediction $\hat{P}$ from $\mathbf{S}_{\text{SM}}$. Next, we calculate the entropy confidence for each pixel on P and $\hat{P}$, resulting in confidence maps C and $\hat{C}$ with the same shape as P and $\hat{P}$. A confidence variance map V is then calculated as the difference between C and $\hat{C}$, i.e., $V = \hat{C} - C$. We use V as a bias term, adding it to the softmax matrix of a vanilla attention to derive a modified attention that is formulated as:

$$\mathbf{N} = \text{softmax}\left(\mathbf{Q}_{\phi[\mathbf{S}_{\text{QP}},\mathbf{S}_{\text{SM}}]}\mathbf{K}_{f_q} + V\right)\mathbf{V}_{f_q}, \tag{3.6}$$

where $\phi[\mathbf{S}_{\text{QP}}, \mathbf{S}_{\text{SM}}]$ denotes passing the concatenation of $\mathbf{S}_{\text{QP}}$ and $\mathbf{S}_{\text{SM}}$ through a two-layered MLP. The feature produced by this operation reflects the evolution from $\mathbf{S}_{\text{QP}}$ to $\mathbf{S}_{\text{SM}}$ through the SM step. V and f_q are flatten along the spatial dimension before the attention. For a pixel (i, j) on V, a higher value of $V^{i,j}$ indicates a sharper decline in its prediction confidence, and vice versa. By adding V to the softmax matrix, the attention is encouraged to focus on pixels with reduced confidences. In this way, the attention can capture noisy information $\mathbf{N}$ accumulated by the SM step. Eventually, we remove the noisy information from $\mathbf{S}_{\text{SM}}$ by:

$$\mathbf{S}_{\text{IC}} = \mathbf{S}_{\text{SM}} - \phi\left(\mathbf{N}\right), \tag{3.7}$$

where ϕ is a linear layer. The generated $\mathbf{S}_{\text{IC}}$ is the output of the IC step, and also the input for the QP step in the next iteration ($\mathbf{S}^{t+1}_{\text{QP}} \leftarrow \mathbf{S}^{t}_{\text{IC}}$).

3.3.5 Optimization

We optimize the model with the following loss function:

$$\mathcal{L} = \sum_{t=1}^{T} L_{CE}\left(P_t, G_q\right) + \lambda \sum_{t=1}^{T-1} L_{CE}\left(\hat{P}_t, G_q\right), \tag{3.8}$$

where T refers to the number of iterations. G_q is the ground truth of the query image. P_t denotes the query prediction from the QP step in the t-th iteration. $\hat{P}_t$ denotes the intermediate prediction from the IC step in the t-th iteration. L_{CE} refers to the cross-entropy loss function. λ is a hyper-parameter.

3.3.6 Extension to K-shot Setting

In the aforementioned sections, we describe our method under the 1-shot setting where only one support image is available ($K = 1$). Our method can be easily extended to K-shot setting by solely modifying the initialization approach of $\mathbf{S}_{\mathrm{QP}}^1$ for the first iteration in the iterative framework. Specifically in the K-shot setting, for each support image $I_s^k, k \in [1, K]$, we first extract a feature f_s^k from the backbone network. Then, to initialize $\mathbf{S}_{\mathrm{QP}}^1$ for the QP step of the first iteration, we concatenate the features of all foreground pixels across $\{f_s^k\}_{k=1}^K$ from all K support images. The subsequent steps remain the same as in the 1-shot setting.

3.4 Experiments

3.4.1 Datasets

We evaluate our method on two widely-used datasets: PASCAL-5^i and COCO-20^i. The PASCAL-5^i dataset contains images from PASCAL VOC 2012, with annotations extended from the SDS to include 20 categories. The COCO-20^i dataset is based on the MSCOCO dataset and includes 80 categories. To be consistent with previous works, we divided the overall categories into four folds and conduct experiments in a cross-validation manner, i.e., to use three folds for training and the remaining one for testing.

3.4.2 Implementation Details

We employ ResNet50 and ResNet101 pretrained on ImageNet as the network backbone with the structure in [98] to enhance feature effectiveness, followed a correlation modules in [109] to fuse information. L discrete bins is used to generate histograms for the structure-wise evolution feature, where L is set to 16. T indicating the number of iterations for the iterative structure is 3. The model is trained for 250 epochs on PASCAL-5^i and 70 epochs on COCO-20^i. λ in Eq. 3.8 is set to 0.2. We use SGD as the optimizer with momentum and weight decay set to 0.9 and 0.0001, respectively. We set the initial learning rate to 0.002 and batch size to 16 for PASCAL-5^i, and 0.005 with batch size 8 for COCO-20^i. We adopt 'poly' policy as the learning rate decay strategy, where the learning rate for each iteration equals to initial rate multiplied by $\left(1 - \frac{iter}{max\_iter}\right)^{0.9}$.

Backbone	Method	Conference	1-shot					5-shot				
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
ResNet50	NTRENet	CVPR2022	65.4	72.3	59.4	59.8	63.2	66.2	72.8	61.7	62.2	65.7
	BAM	CVPR2022	69.0	73.6	67.5	61.1	67.8	70.6	75.1	70.8	67.2	70.9
	AAFormer	ECCV2022	69.1	73.3	59.1	59.2	65.2	72.5	74.7	62.0	61.3	67.6
	SSP	ECCV2022	60.5	67.8	66.4	51.0	61.4	67.5	72.3	**75.2**	62.1	69.3
	IPMT	NeurIPS2022	72.8	73.7	59.2	61.6	66.8	73.1	74.7	61.6	63.4	68.2
	ABCNet	CVPR2023	68.8	73.4	62.3	59.5	66.0	71.7	74.2	65.4	67.0	69.6
	HDMNet	CVPR2023	71.0	75.4	68.9	62.1	69.4	71.3	76.2	71.3	68.5	71.8
	MIANet	CVPR2023	68.5	75.8	67.5	63.2	68.7	70.2	77.4	70.0	**68.8**	71.7
	MSI	ICCV2023	71.0	72.5	63.8	**65.9**	68.5	73.0	74.2	70.5	66.6	71.1
	SCCAN	ICCV2023	68.3	72.5	66.8	59.8	66.8	72.3	74.1	69.1	65.6	70.3
	ABCB (Ours)	CVPR2024	**72.9**	**76.0**	**69.5**	64.0	**70.6**	**74.4**	**78.0**	73.9	68.3	**73.6**
ResNet101	NTRENet	CVPR2022	65.5	71.8	59.1	58.3	63.7	67.9	73.2	60.1	66.8	67.0
	DCAMA	ECCV2022	62.5	70.8	64.5	56.4	63.5	70.0	73.8	66.8	65.0	68.9
	VAT	ECCV2022	68.1	71.7	64.8	63.3	67.0	72.6	74.1	69.5	69.5	71.4
	ABCNet	CVPR2023	65.3	72.9	65.0	59.3	65.6	71.4	75.0	68.2	63.1	69.4
	MSI	ICCV2023	**73.1**	73.9	64.7	68.8	70.1	73.6	76.1	68.0	71.3	72.2
	SCCAN	ICCV2023	70.9	73.9	66.8	61.7	68.3	73.1	76.4	70.3	66.1	71.5
	ABCB (Ours)	CVPR2024	73.0	**76.0**	**69.7**	**69.2**	**72.0**	**74.8**	**78.5**	**73.6**	**72.6**	**74.9**

TABLE 3.1: Performance comparison with other methods on PASCAL-5^i.

For the input images, we employ random scaling and horizontal flipping for data augmentation, and then crop images to the size 473×473 for PASCAL-5^i and 641×641 for COCO-20^i to get the training samples. The experiments are implemented using Pytorch on NVIDIA Tesla V100 GPUs.

3.4.3 Comparison to State-of-the-art

To evaluate the effectiveness of our method, we compare it with other state-of-the-art methods under different backbones, including ResNet50 and ResNet101, and on a variety of few-shot settings, including 1-shot and 5-shot. For each setting, we report the results of using different folds as the test set as well as their mean result. All the compared methods are published in the past 2 years.

The results on PASCAL-5^i are shown in Table 4.1. Under both 1-shot and 5-shot settings, our method can significantly outperform existing methods for both ResNet50 and ResNet101 backbones. To be specific, for the ResNet101 backbone, our method achieves 72.0% and 74.9% mIoUs on the 1-shot and 5-shot settings, outperforming the second-place method by 1.9% and 2.7%, respectively. The results demonstrate that our method can work well and achieve outstanding performance for both 1-shot and multi-shot segmentation. It is worth noting that some of the compared methods [41, 72, 160] also make use of image background information. Our method differs from them by addressing the problem of background context bias for the first time, thus achieving the best performance.

The results on COCO-20^i are shown in Table 3.2. Based on ResNet101, our approach performs better than the second-place method by 1.7% and 1.8%, reaching 51.5% and 58.8% mIoUs on the 1-shot and 5-shot settings respectively. Compared to PASCAL-5^i, COCO-20^i is more challenging due to its more complicated backgrounds. Despite these challenges, our method still shows significant advantages benefiting from the context-based modulation, as demonstrated by the excellent results in various settings.

Backbone	Method	Conference	1-shot					5-shot				
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
ResNet50	NTRENet	CVPR2022	36.8	42.6	39.9	37.9	39.3	38.2	44.1	40.4	38.4	40.3
	BAM	CVPR2022	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
	SSP	ECCV2022	35.5	39.6	37.9	36.7	47.4	40.6	47.0	45.1	43.9	44.1
	MM-Former	NeurIPS2022	40.5	47.7	45.2	43.3	44.2	44.0	52.4	47.4	50.0	48.4
	ABCNet	CVPR2023	42.3	46.2	46.0	42.0	44.1	45.5	51.7	52.6	46.4	49.1
	MIANet	CVPR2023	42.5	53.0	47.8	47.4	47.7	45.8	58.2	51.3	51.9	51.7
	MSI	ICCV2023	42.4	49.2	49.4	46.1	46.8	47.1	54.9	54.1	51.9	52.0
	SCCAN	ICCV2023	40.4	49.7	49.6	45.6	46.3	47.2	57.2	**59.2**	52.1	53.9
	ABCB (Ours)	CVPR2024	**44.2**	**54.0**	**52.1**	**49.8**	**50.0**	**50.5**	**59.1**	57.0	**53.6**	**55.1**
ResNet101	NTRENet	CVPR2022	38.3	40.4	39.5	38.1	39.1	42.3	44.4	44.2	41.7	43.2
	SSP	ECCV2022	39.1	45.1	42.7	41.2	42.0	47.4	54.5	50.4	49.6	50.2
	IPMT	NeurIPS2022	40.5	45.7	44.8	39.3	42.6	45.1	50.3	49.3	46.8	47.9
	ABCNet	CVPR2023	36.5	35.7	34.7	31.4	34.6	40.1	40.1	39.0	35.9	38.8
	MSI	ICCV2023	44.8	54.2	52.3	48.0	49.8	49.3	58.0	56.1	52.7	54.0
	SCCAN	ICCV2023	42.6	51.4	50.0	48.8	48.2	49.4	61.7	61.9	55.0	57.0
	ABCB (Ours)	CVPR2024	**46.0**	**56.3**	**54.3**	**51.3**	**51.5**	**51.6**	**63.5**	**62.8**	**57.2**	**58.8**

TABLE 3.2: Performance comparison with other methods on COCO-20^i.

3.4.4 Ablation Study

We conduct several ablation studies to verify the effectiveness of our designs. The experiments in this section are performed on PASCAL-5^i fold-0 with the ResNet50 backbone.

Ablation of Different Components

Our method is structured as an iterative process with three successive steps: Query Prediction (QP), Support Modulation (SM), and Information Cleansing (IC). We conduct experiments to evaluate the effectiveness of each component, and the results are shown in Table 3.3. A network only comprising a backbone and a QP step is used as the baseline, which achieves 61.48% mIoU. Incorporating the SM step on the baseline results in a performance boost, achieving a mIoU of 68.20%, 6.72% higher than the baseline. The further usage of IC step improves mIoU to 72.92%. The results demonstrate that each component can contribute to performance improvement in our method.

Method	mIoU
Baseline (Backbone + QP)	61.48
Baseline + SM	68.20
Baseline + SM + IC	72.92

TABLE 3.3: Ablation of different components in our method.

Ablation of Support Modulation step

To improve the effectiveness of support foreground features, we use context information extracted from query images for modulation. To capture a comprehensive context, we generate two query evolution features: pixel-wise evolution feature $\mathbf{E}^p$ and structure-wise evolution feature $\mathbf{E}^s$, which capture the evolution of pixels and global structures respectively. To verify the necessity of these representations, we conduct experiments and present the results in Table 3.4. By removing $\mathbf{E}^p$ and $\mathbf{E}^s$, performance is decreased by 4.07% and 2.63% respectively. These results suggest that both features are important to capture a comprehensive evolution feature, which is crucial for ensuring an effective context extraction.

Method	mIoU
Ours	72.92
Ours w/o $\mathbf{E}^p$	68.85
Ours w/o $\mathbf{E}^s$	70.29

TABLE 3.4: Ablation of $\mathbf{E}^p$ and $\mathbf{E}^s$ used in support modulation step

Ablation of Number of Iterations

An iterative structure is employed in our method with T iterations. We perform experiments to determine the best choice for the hyperparameter T and present the results in Table 3.5, which shows the mIoUs and MACs for different settings. As shown in the table, the mIoU improves from 61.48% to 72.92% as T increases from 1 to 3. When T is higher than 3, further increasing T does not significantly improve performance, but rather increases computation burden. Therefore, we choose $T = 3$ as the optimal number of iterations.

Number of Iterations (T)	mIoU	MACs (G)
1	61.48	225.8
2	69.82	241.3
3	72.92	257.0
4	72.99	272.7
5	73.05	288.2

TABLE 3.5: Ablation for the number of iterations.

Ablation of Bin Number

The generation of histograms used for structure-wise evolution feature $\mathbf{E}^s$ involves a step that equally subdivides the continuous range [0, 1] into L bins, with the l-th one denoting the interval $[(l-1)/L, l/L]$ (refer to Sec 4.3 of the main paper for details). In Figure 3.5, we present the validation results of using different numbers of bins. The experiments are conducted on PASCAL-5^i fold-0 with the ResNet50 backbone. It is observed that when L is greater than 12 and less than 24, the mIoU remains stable. When L is too small, the histogram is coarse, resulting in less effective structure information extraction and lower validation accuracy. Conversely, when L is too large, overfitting may occur to hinder the model's effectiveness. Based on experimental results, we choose 16 as the setting of L.

Ablation of Information Cleansing

In the information cleansing (IC) step, we propose a confidence-biased attention to extract the accumulated noisy information, in which the softmax matrix in an attention

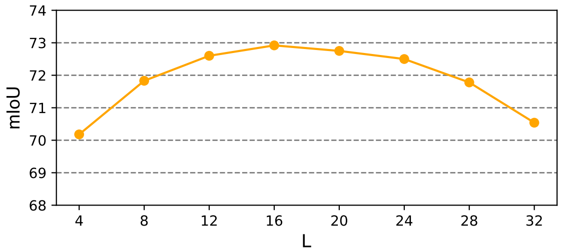


FIGURE 3.5: mIoUs when setting L to different values. We choose L=16 as our final setting.

is summed with the confidence variance V. To validate the effectiveness of this key design, we remove the bias term V in Eq.5, transferring the confidence-biased attention into a normal one. The results are shown in Table 3.6. This modification decreases mIoU from 72.92% to 68.53%. Compared to the method without the IC step, which yields 68.20% mIoU, using IC step without the bias term V almost brings no improvement. This demonstrates that the improvement of IC step is not from the increase in the number of parameters, but due to the removement of noisy information that is extracted by our proposed novel attention mechanism.

Method	mIoU
Ours	72.92
Ours w/o using V in Eq.5 of main paper	68.53
Ours w/o IC step	68.20

TABLE 3.6: Ablation study of information cleansing step. Removing V in Eq.5 significantly decreases performance, demonstrating the importance of the proposed confidence-biased attention.

3.4.5 Predictions at Different Iterations

To gain a deeper evaluation of our iterative structure, we analyze the segmentation results from different iterations. The total number of iterations T is set to 3. The results are presented in Table 3.7, which shows the mIoUs of predictions generated at different iterations (t=1,2,3). As t increases, the segmentation results continue to improve. In the first iteration, we obtain suboptimal predictions because the feature misalignment is caused by the different backgrounds, which results in the low mIoU scores. By using the iterative structure, the performance is improved significantly, resulting in a 9.84% increase in mIoU from the first to the final iteration.

Iteration (t)	mIoU
1	63.08
2	70.76
3	72.92

TABLE 3.7: The mIoUs of predictions from different iterations.

3.4.6 Mitigation of the Feature Misalignment

To demonstrate that our method can eliminate foreground feature misalignment caused by background context bias, for all episodes of the test set, we compute the cosine distance between foreground average features of support and query images. This evaluation is conducted on both the baseline and our method for comparison. The distribution of these distance values is shown in Figure 3.6, where the horizontal axis represents different distance intervals, and the vertical axis indicates the proportion of episodes falling into each interval. Compared to the baseline, when our method is used, there are more episodes with lower foreground feature distances. This shows that our method can effectively narrow foreground feature distances between support and query images, thereby enhancing the effectiveness of support foreground features in guiding query segmentation.

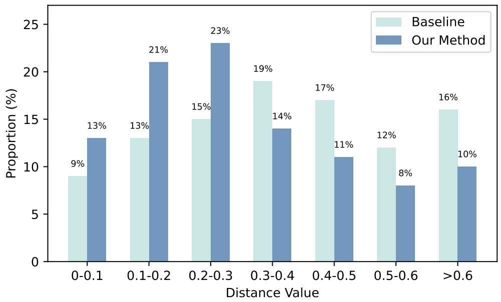


FIGURE 3.6: Statistical distribution of distances between the support foreground features and query foreground features across all episodes. Baseline refers to backbone with a single QP step.

3.4.7 Visualizations

In order to illustrate the effectiveness and advantages of our method, in Figure 3.7, Figure 3.8 and Figure 3.9, we present the following three types of visualizations:

Visualization Comparisons with SOTA methods

In Figure 3.7, we provide prediction visualizations of different methods when the support and query images have significantly different backgrounds. The compared methods include CANet [173], SSP [41], and IPMT [90]. Under this challenging scenario, all the compared methods encounter the background context bias issue, which results in unsatisfactory predictions. In contrast, our method effectively addresses this issue and produces significantly better predictions. These results demonstrate that our method can achieve robust results when support and query images have different backgrounds and outperforms the other methods.

Prediction Visualizations of Using Features After Each Step

In Figure 3.8, we present the prediction results by using the support features $\mathbf{S}_{\mathrm{QP}}$, $\mathbf{S}_{\mathrm{SM}}$ and $\mathbf{S}_{\mathrm{IC}}$ after each of the QP, SM, and IC steps in an iteration as the guidance feature

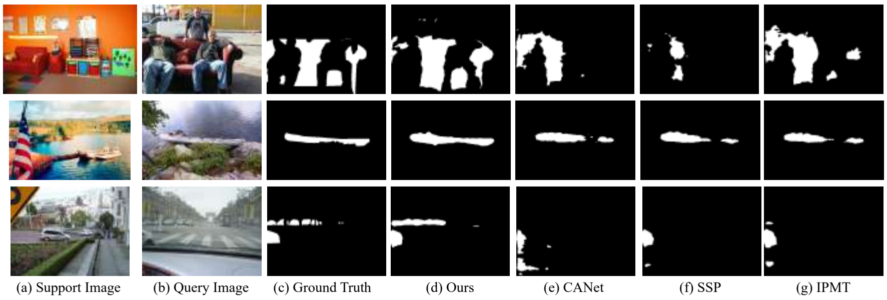


FIGURE 3.7: Prediction visualizations of different methods when the support and query images have significantly different backgrounds.

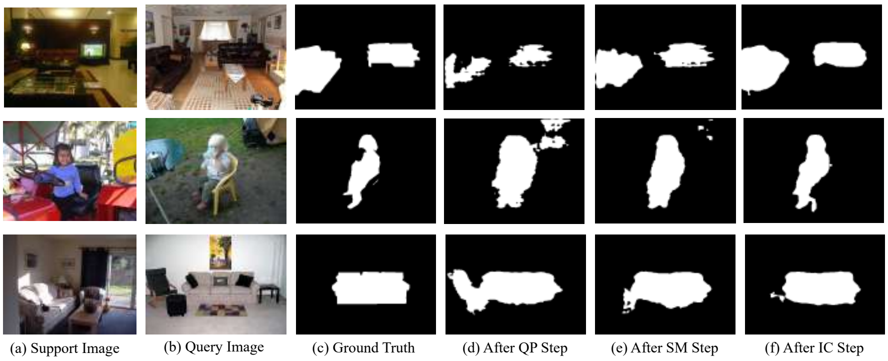


FIGURE 3.8: Prediction visualizations by using the output features after each of the QP, SM, and IC steps in an iteration for query guidance.

used in Eq.3.1. It can be observed that the prediction gradually refines after each step, which demonstrates the benefit of using each step proposed in our method.

visualization from Different Iterations

We further present more visualization results of our method in Figure 3.9, which shows the segmentation predictions generated from different iterations in our framework ($t = 1, 2, 3$). Due to the feature misalignment resulting from the different background contexts, the predictions from the first iteration are suboptimal. Our iterative structure remarkably improves the performance, resulting in gradually refined segmentation results from the first to the final iteration.

3.4.8 Computation Cost and Parameter Number

Our method is both effective and efficient. Compared to the baseline, which comprises a backbone network followed by a single QP step, our methods only increase computation and memory usage slightly. Specifically, on the PASCAL-5^i dataset with the ResNet50 backbone, the baseline consumes 210.3G MACs of computation, requiring 27.6M parameters. Our method consumes 257.0G MACs of computation, requiring 33.5M parameters. Compared to the baseline, our method significantly improves mIoU (+16.50%) while only increasing computation and memory by 22.2% and 21.4%,

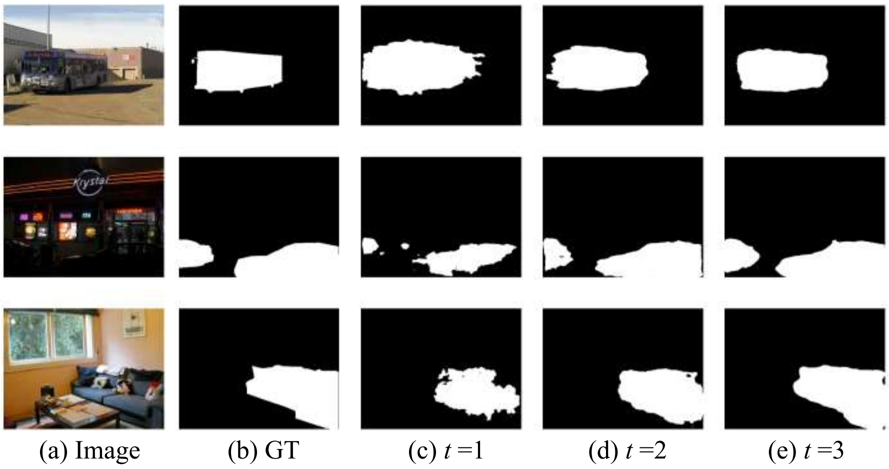


FIGURE 3.9: Visualizations of predictions from different iterations t of our iterative structure.

respectively. We also calculate the parameter usage of each step in our method. Specifically, the QP step, SM step, and IC step require 0.6M, 3.9M, and 2.0M parameters, respectively. It is worth noting that different iterations share the same parameters, thus avoiding the huge memory cost caused by the iterative structure.

3.5 Conclusion

This chapter presents a novel method to address background context bias in few-shot segmentation. In this method, we employ an iterative structure involving three successive steps: Query Prediction, Support Modulation, and Information Cleansing. This structure can address the misalignment of foreground features and reduce the noise accumulation, creating a recurrent optimization scheme that can continuously refine the segmentation results. Experiments on PASCAL-5^i and COCO-20^i demonstrate the high effectiveness of our method in achieving SOTA segmentation results. We believe that our research provides valuable insights and advancements in the field of few-shot segmentation, which can contribute to the development of segmentation algorithms with higher accuracy and robustness.

Chapter 4

LLaFS++: Few-Shot Image Segmentation with Large Language Models

In this chapter, we also focus on the problem of few-shot segmentation (FSS) as the previous Chapter 3. Despite the rapid advancements in FSS, most of existing methods in this domain, such the one we introduced in Chapter 3, are hampered by their reliance on the limited and biased information from only a small number of labeled samples. This limitation inherently restricts their capability to achieve sufficiently high levels of performance. To address this issue, in this chapter, we propose a pioneering framework named LLaFS++, which, for the first time, applies large language models (LLMs) into FSS and achieves notable success. LLaFS++ leverages the extensive prior knowledge embedded by LLMs to guide the segmentation process, effectively compensating for the limited information contained in the few-shot labeled samples and thereby achieving superior results. To enhance the effectiveness of the text-based LLMs in FSS scenarios, we present several innovative and task-specific designs within the LLaFS++ framework. Specifically, we introduce an input instruction that allows the LLM to directly produce segmentation results represented as polygons, and propose a region-attribute corresponding table to simulate the human visual system and provide multi-modal guidance. We also synthesize pseudo samples and use curriculum learning for pretraining to augment data and achieve better optimization, and propose a novel inference method to mitigate potential oversegmentation hallucinations caused by the regional guidance information. Incorporating these designs, LLaFS++ constitutes an effective framework that achieves state-of-the-art results on multiple datasets including PASCAL-5^i, COCO-20^i, and FSS-1000. Our superior performance showcases the remarkable potential of applying LLMs to process few-shot vision tasks.

4.1 Introduction

Image segmentation is a fundamental task in computer vision with broad applications. The advent of deep learning algorithms trained by expansive datasets has brought significant progress to this domain. However, annotating pixel-level segmentation labels on a large scale is extremely resource-intensive. Consequently, few-shot segmentation, a more source-efficient learning paradigm that requires fewer annotated samples, has garnered increasing interest within the academic community and holds immense practical value.

In few-shot segmentation (FSS), a model is required to recognize and segment a novel class based on very few annotated examples, known as the support images. Motivated by the success of few-shot classification, the majority of FSS approaches typically adopt a support-feature-guided mechanism. This mechanism involves extracting representative features from the support images to assist in segmenting an unlabeled image, referred to as the query image. Several methods have been proposed to boost the effectiveness of this mechanism, focusing on enhancing the extraction method of support features [74, 106, 98] or improving how these features assist in segmenting the query images [57, 147, 160]. Although these methods have made some incremental improvements, their segmentation performance is still far from satisfactory. A critical factor contributing to this underperformance issue is their heavy dependence on a very limited set of support images, which can only provide a narrow, incomplete, and possibly biased set of information. Consequently, frameworks that depend exclusively on such restricted data are inherently limited by informational constraints, thus incapable of achieving sufficiently high accuracy. In light of these limitations, we believe that the further advancement of FSS requires the development of an entirely new framework. Such a framework should be capable of utilizing a richer and more comprehensive set of information, thus breaking through the bottlenecks of the existing paradigms to enable superior performance.

In the chapter, we delved into the large language models (LLMs) and found them to be an effective foundation for developing such a brand-new framework with more information gotten involved. Specifically, we identified two properties of LLMs that motivated us to leverage them as a few-shot segmenter: (1) LLMs' extensive pretraining on broad textual corpora equips them with a vast amount of prior knowledge, which can effectively supplement the insufficient information in support images, thus providing enhanced guidance. (2) Furthermore, LLMs have been demonstrated to be effective few-shot learners in NLP [11]. This success naturally inspires us to further extend their capabilities to few-shot tasks in other modalities. Drawing on these motivations, in [188], we introduced LLaFS, an innovative framework that pioneered applying LLMs to FSS and achieved SOTA results. Unlike some previous FSS methods that also use language models (LMs) but only for auxiliary purposes, such as utilizing LMs to extract text features [164, 139], our LLaFS is the first to leverage the more powerful large language models and directly employs LMs to generate segmentation results. This approach elevates LMs from a supportive position to a central role, making them no longer work as only auxiliary tools but unlocking their complete potential to perform complex vision tasks in an end-to-end manner. In this way, we provide a pioneering exploration towards creating a generalized framework that empowers LLMs to address few-shot learning challenges in other modalities beyond NLP.

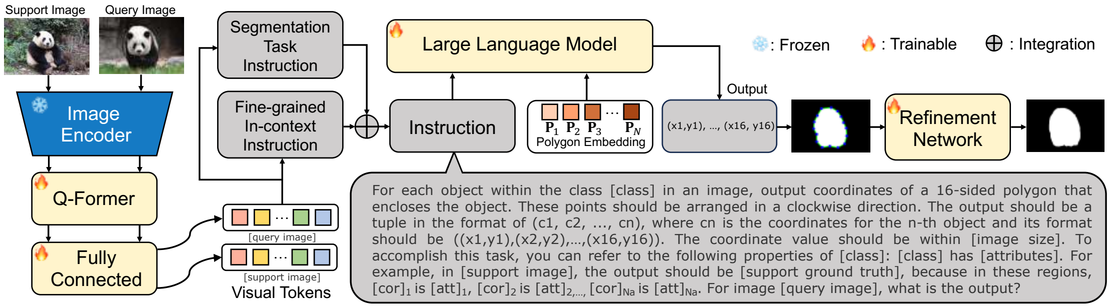
In the research, we find that utilizing LLMs for FSS presents a lot of significant, non-trivial challenges that must be overcome. A primary issue is how to adapt LLMs, which are designed to produce text-based outputs, to the requirements of image segmentation that demands to output pixel-level binary masks. Drawing inspiration from previous work [141], we tackle this challenge by representing segmentation results as the vertices of 16-sided polygons and crafting an instruction within the LLM's input to explicitly define this format. This approach provides LLM with a clear hint about the task's definition and requirements, thereby prompting it to handle image segmentation more effectively and robustly. Another crucial challenge lies in how to effectively combine the visual information from support images with the textual information from LLMs to guide the segmentation of query images. Leveraging the LLMs' strong capacity for in-context learning, we treat support images as demonstration exemplars and introduce a region-attribute corresponding table as a more fine-grained multi-modal guidance. This table details specific attributes of the target class alongside their corresponding regions on the support image, thereby instructing the LLM to execute segmentation in a more fine-grained and human-like manner. Moreover, we notice the issue of training difficulty caused by the limited data and propose a pseudo-sample generation strategy to tackle it with a curriculum learning mechanism to facilitate optimization. By incorporating these innovative designs, our proposed LLaFS framework presents excellent effectiveness in handling few-shot segmentation.

While the LLaFS framework has demonstrated impressive performance, we still identified some limitations, which we also aim to address in this chapter. A primary concern arises from the proposed region-attribute corresponding table, which is designed to align local regions in the support image with specific class attributes to form a fine-grained guidance. The existing approach for aligning region-attributes in LLaFS, which requires cropping images from local regions before extracting their CLIP features, could potentially diminish the model's effectiveness, since the act of image cropping may eliminate crucial global context information, possibly leading to imprecise alignment and, consequently, undermining the effectiveness of the corresponding table. To overcome this challenge, we introduce a simpler yet more effective technique to construct the region-attribute corresponding table. This method not only yields superior results but also reduces the computational cost required by the alignment process. Another issue of the region-attribute table is that it contains many features describing local areas of a class, such as the black eyes of a panda. These locally-focused features could potentially mislead the LLM into focusing on segmenting only local regions rather than the entire object of the target class, thus resulting in the reduced segmentation performance. To mitigate this issue, we introduce a novel inference method within this paper, employing a contrastive prediction strategy designed to exclude incorrectly predicted local regions. By incorporating these two enhancements into the existing framework, We propose LLaFS++, a more robust and effective FSS framework with higher performance.

We conduct extensive experiments across multiple datasets, and the results validate the outstanding performance of LLaFS and LLaFS++. On PASCAL-5^i, COCO-20^i, and FSS-1000, LLaFS++ achieves improvements of 7.0%, 8.3%, and 3.1% respectively compared to the previously reported best results. We also carry out comprehensive ablation studies to demonstrate the effectiveness and rationality of each module and design within our LLaFS++ framework. In summary, the main contributions of this chapter are as follows:

- We propose LLaFS (and its extension LLaFS++), the first framework to address few-shot segmentation using large language models.

- We propose various innovative designs to make better use of LLMs in few-shot



For each object within the class [class] in an image, output coordinates of a 16-sided polygon that encloses the object. These points should be arranged in a clockwise direction. The output should be a tuple in the format of (c1, c2, ..., cn), where cn is the coordinates for the n-th object and its format should be ((x1,y1),(x2,y2),...,(x16,y16)). The coordinate value should be within [image size]. To accomplish this task, you can refer to the following properties of [class]: [class] has [attributes]. For example, in [support image], the output should be [support ground truth], because in these regions, $[\text{cor}]_1$ is $[\text{att}]_1$, $[\text{cor}]_2$ is $[\text{att}]_{2,\ldots,}$ $[\text{cor}]_{\text{Na}}$ is $[\text{att}]_{\text{Na}}$. For image [query image], what is the output?

FIGURE 4.1: Overview of LLaFS++. The image encoder and Q-former extract image features and generate a set of visual tokens. Subsequently, a segmentation task instruction and fine-grained in-context introduction are introduced to provide detailed and comprehensive information. These two instructions are integrated and fed into the LLM along with a set of polygon embeddings $\{\mathbf{P}_n\}_{n=1}^{N}$ to produce the vertices coordinates of polygons that enclose the target object. The segmentation mask represented by this polygon is processed by a refinement network to get the final result.

segmentation, including a task-tailored instruction, a fine-grained in-context instruction serving as multi-modal guidance, a pseudo-sample-based curriculum pretraining mechanism, and a novel inference method to mitigate prediction mistakes.

- Our approach achieves state-of-the-art performance on multiple datasets, with extensive experiments demonstrating the effectiveness of our designs.

4.2 Method

4.2.1 Overview

In this chapter, we aim to construct an LLM-based framework for few-shot segmentation, i.e., to segment a query image I_q based on N_s support images $\{I_s^n\}_{n=1}^{N_s}$ and their ground truth maps $\{G_s^n\}_{n=1}^{N_s}$.[1] As shown in Figure 4.1, the overall framework of LLaFS++ can be divided into three key components: (1) a feature extractor that extracts image features and generates visual tokens; (2) a task-tailored instruction that combines visual tokens, target categories, and task requirements to provide task-related information and support guidance; and (3) an LLM that predicts segmentation masks based on the input instruction and segmentation embeddings, followed by a refinement network to optimize the results. For the feature extractor, we adopt the approach in Blip2 [75] by using an image encoder followed by a Q-former and a fully-connected layer to generate a set of visual tokens. We use ResNet as the image encoder and keep it frozen during training. For the instruction, we carefully design it as the combination of two parts: segmentation task instruction (Sec.4.2.2) and fine-grained in-context instruction (Sec.4.2.3) to provide comprehensive and detailed guidance. The instruction

[1] For simplify of illustration, we introduce LLaFS++ under the one-shot setting. Appendix presents how to extend LLaFS++ to the multi-shot setting.

is concatenated with a set of learnable segmentation embeddings $\{\mathbf{P}_n\}_{n=1}^{N}$ (Sec.4.2.4) for inputting into the LLM. For the LLM, we employ CodeLlama [118] with 7 billion parameters that have been finetuned through instruction tuning. Note that compared to vanilla Llama, we empirically find that CodeLlama finetuned with code generation datasets exhibits higher accuracy and stability in generating structured information like the segmentation result in our task. We equip CodeLlama with LoRA for finetuning. All these components work together within the LLaFS++ framework to achieve high-quality few-shot segmentation.

As the input of LLM, the instruction is the most crucial component in our framework that makes LLM possible to handle few-shot segmentation. To provide comprehensive information, we design two instructions, namely segmentation task instruction and fine-grained in-context instruction, to respectively provide the LLM with detailed task definitions and fine-grained multi-modal guidance. These two instructions are integrated to formulate the complete instruction as shown in Figure 4.1. In the following Sec.4.2.2 and Sec.4.2.3, we introduce these two instructions in detail.

4.2.2 Segmentation Task Instruction

The LLMs trained on massive text contents have gained strong reasoning capabilities and a vast amount of world knowledge. Language instructions have shown to be a powerful tool for leveraging this knowledge and capability to handle complex tasks [108]. To achieve better results, the instructions need to be sufficiently clear and detailed, whereas those using only simple terminologies such as 'performing image segmentation' are too abstract for LLMs to comprehend. Thus, we design a structured instruction to explicitly provide more task details such as the expected input and output formats of few-shot segmentation. Specifically, in our instruction, we follow [141] by representing the pixel-wise segmentation output as a set of 16-sided polygons that enclose the target objects [84]. Note that it is hard for LLMs to directly generate pixel-wise segmentation masks due to LLM's limited number of output tokens. Our alternative solution of generating polygon vertices provides a token-efficient method for using LLMs to achieve pixel-level segmentation.

Furthermore, the language-focused design of LLMs poses a challenge for their precise interpretation of visual information. This issue is particularly severe in few-shot image segmentation, where the availability of training images is extremely limited. To address this problem, inspired by the success of in-context learning in NLP [99, 44], we propose a novel strategy that encodes the support image along with its ground truth as a visual demonstration example. This example is then incorporated into the instruction, providing the LLM with a clear and intuitive reference that instructs the LLM on how to accurately segment a specific class within an image.

By incorporating these designs, we write our segmentation task instruction as: *"For each object within the class* [class] *in an image, output coordinates of a 16-sided polygon that encloses the object. These points should be arranged in a clockwise direction. The output should be a tuple in the format of (c1, c2, ..., cn), where cn is the coordinates for the n-th object and its format should be ((x1,y1),(x2,y2),...,(x16,y16)). The coordinate value should be within* [image size]. *For example, for image* [support image], *the output should be* [support ground truth]". Here, [support image] is the visual tokens from the support image, [support ground truth] denotes the vertex coordinates of 16-sided polygons that enclose the support foreground regions.

4.2.3 Fine-grained In-context Instruction

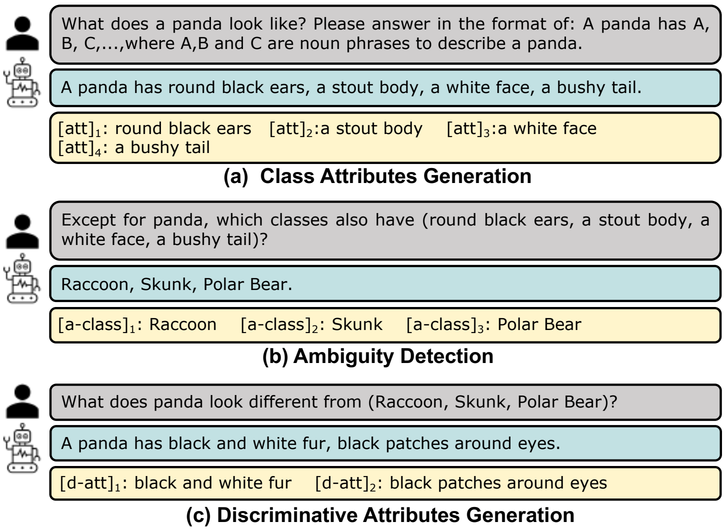
Motivation

The above task instruction makes segmenting a class possible by leveraging LLM's knowledge of the class. In the instruction, the class to be segmented is indicated by the [class] token, which is typically a single noun. However, considering that LLMs are language-based models mainly trained on text corpus, it is challenging for them to directly align this abstract noun with an image region that may possess a complex internal structure. To address this issue, we drew inspiration from human brains and found that when classifying an unseen new class, the human cognitive system follows a mechanism of '*from general to detailed, from abstract to concrete*' [150, 102]. Specifically, given an unseen class represented by a *general* noun, the human brain first decomposes it into *detailed* attributes based on the acquired knowledge. For example, in the case of an unseen class 'panda', a person can first gather information from references to learn about the panda's attributes such as 'black and white fur' and 'black ears'. Subsequently, it can search the image for *concrete* regions that match these *abstract* attributes to determine the presence of the class.

Motivated from the above discussion, we propose a fine-grained in-context instruction that leverages support images to simulate such a human cognitive mechanism. Specifically, we first instruct the LLM to extract detailed attributes of the target class (Sec.4.2.3). Subsequently, we locate regions within support images that match these attributes and create a corresponding table accordingly (Sec.4.2.3). This table, together with the extracted attributes, constitutes an in-context instruction (Sec.4.2.3), which is then fed into the LLM to serve as a demonstration example that guides the LLM on how to recognize image classes in a more human-like and fine-grained manner. This approach effectively mitigates the limitations of LLMs in performing segmentation tasks based solely on generic class names. Furthermore, we also present an LLM-checking framework to refine the produced instructions (Sec.4.2.3). In the following sections, we introduce the method for generating and refining the instruction in detail.

Attributes Extraction

We first simulate the step of '*from general to detailed*' to extract class attributes. Specifically, as shown in Figure 4.2(a), we construct a prompt '*What does a* [class] *look like? Please answer in the format of: A* [class] *has A, B, C,..., where A, B, and C are noun phrases to describe a* [class].', and instruct the LLM in LLaFS++ to extract phrases-based attributes that describe the fine-grained details of this class. These attributes are denoted as [attributes] = $\{[\text{att}]_i\}_{i=1}^{N_a}$. For each $[\text{att}]_i$, we utilize '*A photo of* $[\text{att}]_i$' as a prompt to extract an embedding t_i from the CLIP's text encoder. In this way, we get $\{t_i\}_{i=1}^{N_a}$ from $\{[\text{att}]_i\}_{i=1}^{N_a}$.



What does a panda look like? Please answer in the format of: A panda has A, B, C,...,where A,B and C are noun phrases to describe a panda.

A panda has round black ears, a stout body, a white face, a bushy tail.

[att]$_1$: round black ears [att]$_2$:a stout body [att]$_3$:a white face [att]$_4$: a bushy tail

(a) Class Attributes Generation

Except for panda, which classes also have (round black ears, a stout body, a white face, a bushy tail)?

Raccoon, Skunk, Polar Bear.

[a-class]$_1$: Raccoon [a-class]$_2$: Skunk [a-class]$_3$: Polar Bear

(b) Ambiguity Detection

What does panda look different from (Raccoon, Skunk, Polar Bear)?

A panda has black and white fur, black patches around eyes.

[d-att]$_1$: black and white fur [d-att]$_2$: black patches around eyes

(c) Discriminative Attributes Generation

FIGURE 4.2: Examples of using LLM for (a) class attributes generation, (b) ambiguity detection and (c) discriminative attributes generation.

Region-attribute Corresponding Table

Considering that many attributes describe the locally regional characteristics of a category, for example, 'black ear' for 'panda', to obtain a more refined support guidance, we further simulate the second step of '*from abstract to concrete*' by identifying specific local regions within the support image that can be aligned with these class attributes, and then employing the alignments to construct a fine-grained in-context demonstration example. To implement this alignment, we introduce a simple yet effective method. Specifically, we first feed the support image into an enhanced CLIP image encoder proposed by [101] to produce a feature map $f \in \mathbb{R}^{H \times W \times C}$, where H, W and C represent the height, width, and channel number of f, respectively. Benefiting from its patch-level contrastive pretraining [101], this enhanced CLIP encoder excels in aligning text with specific local image regions. We then compute the cosine similarity between each pixel f^j within f and each attribute embedding t_i to produce a similarity map $M_i \in \mathbb{R}^{H \times W}$. As shown in Figure 4.3(a), it is encouraging to observe that this similarity map, although derived through a simple and straightforward method without complex post-processing, already exhibits a good level of attribute awareness, with regions corresponding to the attribute typically exhibiting higher similarities than the other areas. We further observe that some attributes, such as 'black and white fur' for 'panda', describe the wide-level properties of a class rather than specific details in local regions. In this case, M_i can still capture the presence of such attributes effectively, with a wide range of pixels across the entire foreground showing a high degree of similarity.

The next challenge involves how to encode the attribute-corresponding region captured by M_i into a format that the LLM can receive as input. To tackle this issue, we introduce a lightweight region encoding network (REN) designed to convert M_i into an implicit feature. As shown in Figure 4.3(b), the REN is structured with three serial transformer layers, with the input being the concatenation of the similarity map M_i

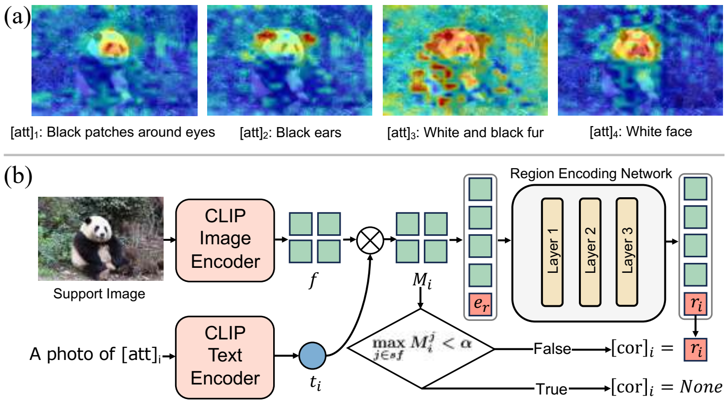


FIGURE 4.3: (a) Examples of similarity maps M_i computed from the support image and class attributes. (b) Illustration of how to construct the region-attribute corresponding table for the i-th attribute [att]$_i$. sf refers to all pixels in support foreground. Note that the spatial shape of f and M_i shown in this figure is 2×2. This is only for the simplification of illustration but not the actual size $H \times W$ used in practice.

and a learnable region embedding e_r. e_r's hidden state r_i at the output of the transformer is utilized as a feature to represent [att]$_i$'s corresponding region in the support image. Note that in the transformer, we employ masked attention [31] rather than the vanilla self-attention to focus REN on the support foreground area that belongs to the target class. During the training process, REN is optimized end-to-end in sync with the LLM. We also notice that not every attribute extracted through Sec.4.2.3 can find a corresponding region on the support foreground, mainly due to the variations in camera angles and instances of occlusion. To prevent introducing misleading information, we use a simple thresholding approach to filter out feature r calculated from these support-non-corresponding attributes. In this way, we establish region-attribute correspondence [cor]$_i$ for each attribute [att]$_i$ by:

$$[\mathrm{cor}]_i = None \text{ if } \max_{j \in sf} M_i^j < \alpha \text{ else } r_i, \tag{4.1}$$

where M_i^j denotes the j-th pixel on M_i and sf refers to all pixels in support foreground. α is a pre-defined threshold. The obtained [cor]$_i$ represents regions in support image that align with the i-th attribute. In this way, we get $\{[\mathrm{cor}]_i\}_{i=1}^{N_a}$ from $\{[\mathrm{att}]_i\}_{i=1}^{N_a}$, which serves as a region-attribute corresponding table that can provide fine-grained multimodal reference.

It is crucial to highlight that the aforementioned technique for creating the region-attribute corresponding table is simpler yet more effective than the method utilized in the conference version LLaFS. As shown in Figure 4.4, in the method proposed in LLaFS, we first divide the support foreground into multiple local regions. Specifically, for each object in the support image within the target class, we employ the method in selective search [135] to generate a set of superpixels $\{s_i\}_{i=1}^{N_r}$ with different scales in an unsupervised manner. Each s_i aggregates pixels that are close in position and similar in features, so it can represent a local region with a specific semantic meaning. Based on

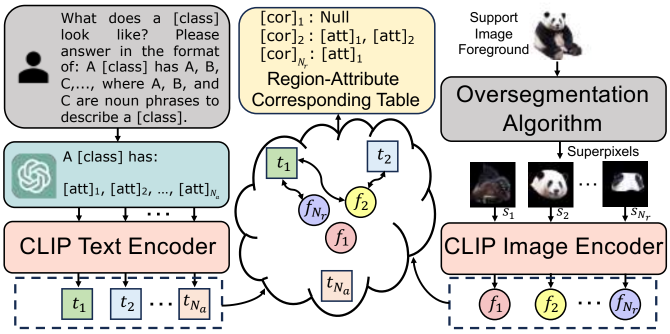

What does a [class] look like? Please answer in the format of: A [class] has A, B, C,..., where A, B, and C are noun phrases to describe a [class].
A [class] has:
[att]$_1$, [att]$_2$, ..., [att]$_{N_a}$
CLIP Text Encoder
t_1
t_2
t_{N_a}
[cor]$_1$: Null
[cor]$_2$: [att]$_1$, [att]$_2$
[cor]$_{N_r}$: [att]$_1$
Region-Attribute Corresponding Table
f_{N_r}
f_2
f_1
Support Image Foreground
Oversegmentation Algorithm
Superpixels
s_1
s_2
s_{N_r}
CLIP Image Encoder

FIGURE 4.4: Illustration of how to construct the region-attribute corresponding table used in the fine-grained in-context instruction in LLaFS.

each s_i, a masked image is generated and passed through CLIP's image encoder to produce a feature f_i. We calculate the cosine similarity between f_i and the embedding t_j for each attribute [att]$_j$, and utilize a thresholding process to establish region-attribute correspondence. This process is formulated as:

$$[\mathrm{cor}]_i = \Big[[\mathrm{att}]_j \;\; \text{for} \;\; j \in [1, N_a] \;\; \text{if} \;\; \cos(f_i, t_j) > \alpha\Big] \tag{4.2}$$

where $\cos$ refers to cosine similarity followed by a softmax operation among the region's similarities with all attributes, α is a pre-defined threshold. The obtained [cor]$_i$ contains attributes that align with s_i. In this way, we get $\{[\mathrm{cor}]_i\}_{i=1}^{N_r}$ from $\{s_i\}_{i=1}^{N_r}$, which serves as an attribute-region corresponding table that can provide the fine-grained multi-modal reference. As mentioned in Sec.4.1, this approach requires the extraction of CLIP features from cropped images, which compromises the region-attribute alignment accuracy due to the loss of context information. In contrast, the improved approach presented in this chapter eliminates this need for image cropping and achieves better performance as demonstrated by the experimental results shown in Table 4.6.

Instruction Construction

We integrate the class attributes $\{[\mathrm{att}]_i\}_{i=1}^{N_a}$ and corresponding table $\{[\mathrm{cor}]_i\}_{i=1}^{N_a}$, and write the fine-grained in-context instruction as: *"To accomplish this task, you can refer to the following properties of* [class]*: The* [class] *has* [attributes]. *For example, in* [support image]*, the output should be* [support ground truth]*, because in these regions,* [cor]$_1$ *is* [att]$_1$, [cor]$_2$ *is* [att]$_2$, ..., [cor]$_{N_a}$ *is* [att]$_{N_a}$*"*. Note that to prevent introducing misleading information, only non-empty [cor]$_i$ will be included in this instruction. By using the instruction as input, we provide the LLM with a detailed reference regarding the attributes of the target class and their corresponding regions in the support image. This creates a demonstration example that simulates how the human cognitive mechanism recognizes the support foreground as the target class. With such an example as reference, the LLM can be taught how to understand and segment an image class in a fine-grained and human-like manner.

Instruction Refinement

The above instruction, which is constructed by the extracted attributes $\{[\text{att}]_i\}_{i=1}^{N_a}$ and table $\{[\text{cor}]_i\}_{i=1}^{N_a}$, can be directly fed into LLM for guidance. However, we have identified potential issues that directly combining the attributes derived from Sec.4.2.3 may introduce class ambiguities due to the shared attributes across different classes. For example, the combination of attributes 'wheels, windows, doors' might be extracted for the 'train' class but could also refer to other classes such as 'bus' and 'car'. Furthermore, since attributes not corresponding to the support image have been filtered out through Eq.4.1, the generated table $\{[\text{cor}]_i\}_{i=1}^{N_a}$ may represent regions for only a subset of attributes within $\{[\text{att}]_i\}_{i=1}^{N_a}$. The combination of these partial attributes is consequently more susceptible to class ambiguities, and thus making the resultant instruction to be confusing and misleading.

To alleviate the aforementioned issue, we propose an LLM-checking framework to refine the instruction. This framework identifies potential ambiguous classes for the existing attributes, and subsequently extracts additional attributes with higher class discrimination ability to mitigate the ambiguity problem. Specifically, the instruction refinement is implemented through the following three steps: ***1) Ambiguity Detection.*** As shown in Figure 4.2(b), we instruct the LLM to identify potential ambiguous classes in the obtained table $\{[\text{cor}]_i\}_{i=1}^{N_a}$. Specifically, we denote the set of all attributes with a non-empty $[\text{cor}]_i$ as [valid-att] and ask the LLM *'Except for* [class]*, which classes also have* [valid-att]*?'* In this way, we obtain a set of ambiguous classes denoted as [a-classes]=$\{[\text{a-class}]_i\}_{i=1}^{N_{ac}}$ from LLM's feedback. ***2) Discriminative Attributes Generation.*** As shown in Figure 4.2(c), to avoid being misled by these ambiguous classes, we use *'What does* [class] *look different from* [a-classes]*?'* as a text prompt, enabling the LLM to generate attributes that are more discriminative from the ambiguous classes. The obtained attributes $\{[\text{d-att}]_i\}_{i=1}^{N_d}$ are added to [attributes] for updating. ***3) Table and Instruction Refinement.*** Finally, using the updated attributes, we generate a refined table by reperforming Eq.4.1. The updated attributes and table are reassembled through the way in Sec.4.2.3 to obtain a refined instruction.

We found that a single execution of the three steps already resolves ambiguities in over 92% of the instructions. While for the residual 8%, the class ambiguities remain, resulting in a still-ambiguous instruction after refinement. To address this problem, we apply the three steps iteratively until the ambiguity is completely eliminated. To achieve this goal, from the second iteration onwards, we replace the text prompt in the discriminative attributes generation step with *'Apart from* [all-d-att]*, tell me more differences in appearance between* [class] *and* [a-classes]', where [all-d-att] refers to the discriminative attributes $[\text{d-att}]_i$ obtained from all previous iterations. This modification enables our iterative framework to continuously discover more discriminative attributes and refine the instruction accordingly. We end the iteration process when either of two conditions is met: the LLM cannot find any ambiguous class, or the number of iterations reaches our predefined maximum. For efficiency, we set this maximum to 3, in which we found 98% of the ambiguities have been entirely eradicated.

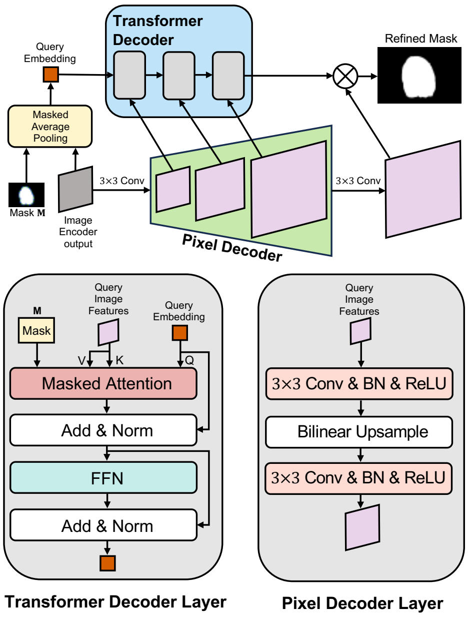


FIGURE 4.5: Structure of the refinement network. This network is lightweight, comprising only 6 convolution layers and 3 attention layers.

4.2.4 Segmentation Prediction

We integrate segmentation task instruction and fine-grained in-context instruction to formulate the complete instruction as shown in Figure 4.1. With this instruction as input, the LLM can predict the vertex coordinates of 16-sided polygons that surround the target objects. Specifically, as shown in Figure 4.1, inspired by MaskFormer, we introduce N sets of polygon embeddings $\{\mathbf{P}_n\}_{n=1}^{N}$, which are concatenated with the instruction and fed into the LLM. Each $\mathbf{P}_n$ consists of 33 learnable embeddings $(\mathbf{x}_n^1, \mathbf{y}_n^1, \mathbf{x}_n^2, \mathbf{y}_n^2,$ $..., \mathbf{x}_n^{16}, \mathbf{y}_n^{16}, \mathbf{v}_n)$, where the LLM's outputs of the first 32 embeddings determine the x- and y-coordinates of the polygon's 16 vertices, and the final embedding $\mathbf{v}_n$, after being processed by the LLM, is fed into a fully-connected layer f_v to yield a validity score v_n. This score reflects the likelihood that the polygon produced by $\mathbf{P}_n$ can accurately represent the target object in the query image. Consequently, we filter out the polygons with $v_n < 0$ and regard the regions enclosed by the remaining polygons as the segmentation prediction result. We use 1 to fill the area enclosed by the valid polygons and 0 for the area outside these polygons, resulting in a binary segmentation mask denoted as $\mathbf{M}$.

Moreover, to rectify the imprecision caused by the polygon representation of object edges, we introduce a refinement network that comprises a pixel decoder and a mask transformer to generate a refined segmentation mask by using these polygons as the initial masks. As shown in Figure 4.5, this refinement network has a structure similar to Mask2Former [31], comprising a pixel decoder that progressively increases the sizes of query image feature maps and a masked transformer decoder for refining the queries. $\mathbf{M}$ is used as the mask for masked attention in the transformer decoder. Readers can refer to Sec.3.21 of [31] for more details of masked attention. Note that, compared to the vanilla Mask2Former, our method does not employ a heavy transformer to construct the pixel decoder; instead, we use a simple structure composed of a small number of convolution and bilinear upsampling layers. Moreover, our approach uses the transformer decoder only once, rather than applying it iteratively. These modifications reduce the computational complexity, resulting in a highly lightweight refinement network with just six convolution layers and three attention layers. Given that the output from the LLM is already quite effective, this lightweight network is entirely adequate for the refinement purpose. Also note that this refinement network is only an optional component that can further improve performance. Excluding it from LLaFS++ and directly using the LLM-generated polygons as the final segmentation mask is completely acceptable, and it can still achieve the SOTA performance.

4.2.5 Curriculum Pretraining with Pseudo Samples

Motivation

After carefully designing the model structure and instruction format, the next challenge is how to train LLaFS effectively to achieve high-quality segmentation results. Previous work [83] has highlighted that the success of LLMs typically relies on the training on extensive data. However, due to the challenge of acquiring pixel-annotated labels, the datasets for training in segmentation often have a limited number of images. To mitigate this limitation, we propose an innovative solution that generates pseudo support-query pairs for pretraining the LLM. The LLM's ability to handle few-shot segmentation can thus be enhanced by seeing more visual samples with segmentation annotations.

Pseudo Sample Generation

Specifically, we propose a method to generate pseudo support-query pairs with the following steps:

Step 1: Support foreground-background partition. We first generate a random contour within a black image. The area surrounded by this contour is considered as the foreground within the target class, while the regions outside the contour are treated as the background. To mimic the variety of backgrounds found in real-world images, we divide the background into multiple subregions using random contours. The number of these subregions is randomly determined, ranging from one to five.

Step 2: Support noise filling. We randomly generate an array $m_{sf} \in \mathbb{R}^3$ within the value range [0, 255] and use it as the RGB mean to create Gaussian noise that fills the support foreground region. For each subregion of the support background, we generate another array $m_{sb} \in \mathbb{R}^3$ as the mean for producing Gaussian noise to fill that subregion. The random generation space of m_{sb} is constrained so that the distance $||m_{sb} - m_{sf}|| \in [a, b]$, where a, b are two adjustable parameters. By modifying a and b, we can control the difference between the foreground and background in each synthetic image. The following section will illustrate how to adjust these parameters in different pretraining steps.

Step3: Query foreground-background partition by adjusting from support. To ensure that the support and query foregrounds share similar shapes and thus represent the same category, the contour used to generate the query foreground is adjusted based on that used for generating the support foreground. Specifically, we first add standard Gaussian noise to the ten control points that define the support foreground contour. This results in a noised contour, which is then further adjusted by random rotation and scaling between [0.5, 1.5]. Subsequently, we randomly place the resulting contour within a black image, and the area it encloses is regarded as the query foreground. Finally, we employ the same method used for the support background to divide the query background into subregions.

Step4: Query noise filling by adjusting from support. Using the same method as for the support generation, we randomly generate arrays $m_{qf} \in \mathbb{R}^3$ and $m_{qb} \in \mathbb{R}^3$ as RGB means to produce Gaussian noises, which fill the query foreground and each subregion of the query background. To ensure that the support and query foregrounds share similar internal features and thus reflecting the same category, we constrain the random generation space of m_{qf} so that the distance satisfies $||m_{qf} - m_{sf}|| \in [c, d]$, where c and d are adjustable parameters. For the background's m_{qb}, we apply two constraints to determine its random generation space: (1) similar to the support background, we maintain the difference between the query background and query foreground within the range [a, b]; (2) to ensure the query foreground is the most similar region to the support foreground in the query image, we also enforce that the difference between the query background and the support foreground is greater than the difference between the query foreground and the support foreground. This is expressed as $||m_{qb} - m_{sf}|| > ||m_{qf} - m_{sf}||$. Under these constraints, m_{qf} and m_{qb} are randomly generated and used as the means of Gaussian noises that fill the query foreground and background, respectively, creating the pseudo query image.

Curriculum Pretraining

The synthetic support-query pairs can be directly used for pretraining. However, this straightforward method is observed to yield a slow rate of convergence. One potential explanation for this issue is that the LLM, given its language-based nature, may face difficulties in optimizing for a complex image processing task. To address this issue, we propose a progressive pretraining approach inspired by the success of curriculum learning [143], in which we initiate the model's pretraining with a simple task and gradually increase the task's difficulty until it ultimately reaches the requirements of segmentation. Experimental results show that this curriculum learning approach allows the model to converge better and achieve higher results.

Specifically, as show in Figure 4.6, during pretraining, we incrementally raise the task's difficulty from the following two aspects:

(1) Image understanding. During pretraining, by controlling the difference between mean values of different filled noise, we gradually increase the difference in foreground

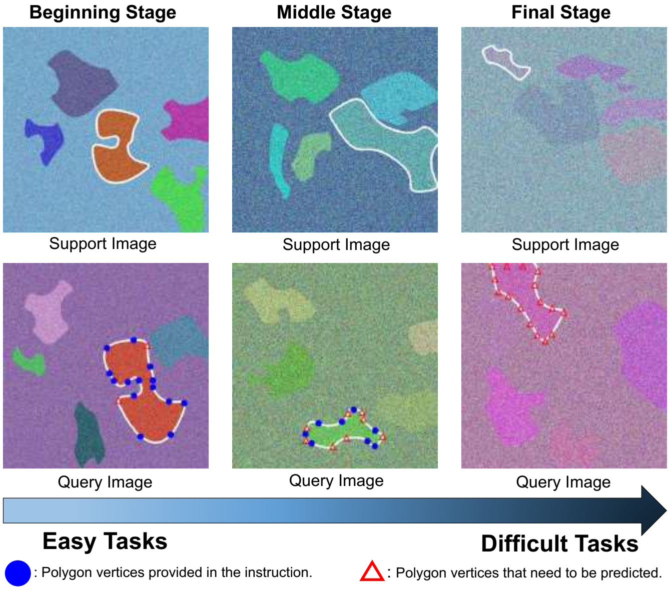


FIGURE 4.6: Examples of pseudo samples generated at different pretraining stages. Foreground regions are marked by white contours. As pretraining progresses, pseudo images have reduced intra-image foreground-background differences and greater support-query foreground differences. Meanwhile, the number of polygon vertex coordinates provided in the instruction decreases, while the predicted vertex count increases. These changes gradually increase the pretraining difficulty. (Best viewed in color)

between the synthetic support and query, while reducing the internal difference between foreground and background within each image. This strategy incrementally increases the challenge for the LLM to execute few-shot guidance and distinguish between foreground and background areas as pretraining progresses.

Specifically, the interval $[a, b]$ controls the difference between the foreground and background in an image. During the pretraining process, to decrease this difference, we gradually reduce the values of and a, b, with a eventually decreasing to 0. Denoting the total number of pretraining steps as N_p ($N_p = 60K$ in our experiments), the values of a_n and b_n at step n are formulated as:

$$\begin{aligned} a_n &= a_0 - \frac{n.a_0}{N_p}, \\ b_n &= a_n + b_0 - a_0, \end{aligned} \tag{4.3}$$

where a_0 and b_0 are the hyper-parameters that define the initial values of a and b in the first step of pretraining.

The interval $[c, d]$ regulates the difference between the support foreground and query foreground. During the pretraining process, to enlarge this difference, we gradually increase the values of c and d, making c to be increased from 0 to c_{N_p} as the step progresses from 0 to N_p. In this way, the values of c_n and d_n at step n are formulated as:

$$\begin{aligned} c_n &= \frac{n.c_{N_p}}{N_p}, \\ d_n &= c_n + d_{N_p} - c_{N_p}, \end{aligned} \tag{4.4}$$

where c_{N_p} and d_{N_p} are the hyper-parameters that define the final values of a and b in the last step of pretraining.

In this approach, a_0, b_0, c_{N_p} and d_{N_p} are predefined hyper-parameters, which are respectively set to 100, 150, 50, and 100 in our framework. Note that, our experiments demonstrate that ***the performance of LLaFS++ is NOT sensitive to these hyper-parameters***. See Sec.4.3.4 and Table.4.11 for details.

(2) Polygon generation. Generating a polygon represented by a combination of vertex coordinates is observed to be another challenge for the LLM. Therefore, we also apply a progressive strategy to this aspect. Specifically, during the pretraining stage, we randomly provide the coordinates of K vertices and task the LLM with predicting the coordinates of the remaining $16 - K$ vertices. K is decreased by 1 every $N_p/30$ steps over the first half of the pretraining process ($N_p/2$ steps). This gradual reduction of K from 15 to 0 means the model receives fewer hints and is required to predict more vertex coordinates as pretraining progresses. Consequently, the pretraining difficulty gradually increases, ultimately reaching the task of predicting all 16 vertices for segmentation. In the second half of the pretraining process ($N_p/2$ steps), we maintain $K = 0$.

Ultimately, the model is trained on the realistic few-shot segmentation dataset after completing the aforementioned pretraining process. We will illustrate the detailed training procedures in the following section.

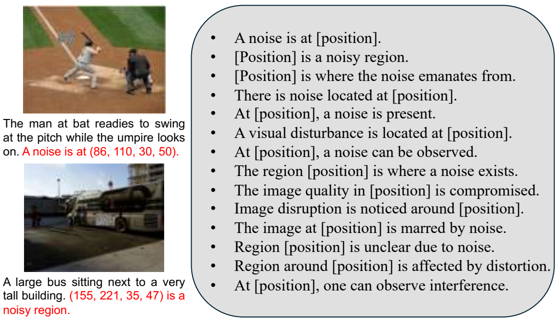
4.2.6 Training and Inference

After introducing our innovative designs within the LLaFS++ framework, in this section, we further elaborate on the complete process of model training and inference methods. Specifically, we follow previous works by using a multi-stage training strategy, and propose a novel inference method to address potential hallucination issues when executing the LLaFS++ framework.

Training

Following the method of Blip2 [75], which commences by training the Q-former independently before jointly training it with the LLM, we train the LLaFS++ framework using three stages, with distinct components targeted at each stage. In the first stage, we freeze the LLM, pretrain the Q-former and fully-connected layers for 100K steps using the image captioning datasets[2] and methods in Blip2 [75], with the aim of enabling the LLM to acquire the capability to process visual images. Note that incorporating image captioning datasets into the training process is a strategy widely adopted by a lot of LLM-based segmentation methods such as LISA [70] and VisionLLM [141]. Thus, we employ the same method as well. Another important point worth mentioning is that

[2] COCO is excluded from the pretraining set to avoid test data leakage.



The man at bat readies to swing at the pitch while the umpire looks on. A noise is at (86, 110, 30, 50).

A large bus sitting next to a very tall building. (155, 221, 35, 47) is a noisy region.

- A noise is at [position].
- [Position] is a noisy region.
- [Position] is where the noise emanates from.
- There is noise located at [position].
- At [position], a noise is present.
- A visual disturbance is located at [position].
- At [position], a noise can be observed.
- The region [position] is where a noise exists.
- The image quality in [position] is compromised.
- Image disruption is noticed around [position].
- The image at [position] is marred by noise.
- Region [position] is unclear due to noise.
- Region around [position] is affected by distortion.
- At [position], one can observe interference.

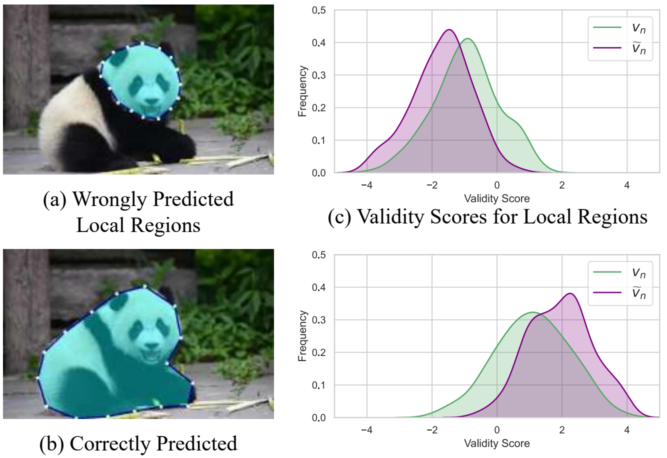
FIGURE 4.7: Examples of augmented image captioning data and templates to extend captions.

we found models trained directly with the original image captioning dataset exhibit poor spatial locality awareness, which is detrimental to image segmentation. To address this issue, we employ a simple data augmentation method based on noise filling. Specifically, as shown in Figure 4.7, we inject Gaussian noise into a randomly chosen rectangular area of each training image. Note that to maintain the integrity of the image's overall content, the dimensions of this rectangle are constrained to no more than 1/8 of the image's size. We then update the image caption to reflect the location of this noisy region. Concretely, to enrich the diversity of the augmented captions, we employ ChatGPT to reformulate the sentence "A noise is at [position]" to 100 variant templates, where [position] is a tuple containing the center coordinates, height, and width of the noisy rectangle. Some of these templates are shown in the right part of Figure 4.7. For each noise-augmented image, one of these templates is randomly selected and appended after its original caption. Our experimental results, detailed in Sec.4.3.4 and Table 4.12, demonstrate that pretraining with such noise-augmented data helps to enhance the model's segmentation performance.

In the second stage, we freeze the Q-former, equip the LLM with LoRA, and pretrain the fully-connected layers, LLM and refinement network using the pseudo-sample-based curriculum learning method (Sec.4.2.5) for 60k steps. In the third stage, we train the fully-connected layers, LLM and refinement network on the realistic few-shot segmentation dataset (25 epochs for PASCAL-5^i and FSS-1000, 3 epochs for COCO-20^i). Note that using the vertex coordinates generated by the LLM to construct a binary mask as input for the refinement network is a non-differentiable process. Therefore, we employ two separate loss functions to train the refinement network and the remaining components of LLaFS++, respectively. In this way, the overall loss can be written as:

$$\mathcal{L} = \mathcal{L}_{llm} + \mathcal{L}_{ref}, \tag{4.5}$$

where $\mathcal{L}_{llm}$ denotes the loss for training fully-connected layers and LLM, $\mathcal{L}_{ref}$ denotes the loss for training the refinement network. For $\mathcal{L}_{llm}$, we first use bipartite matching to align the LLM-predicted polygons with each object in the ground truth, then we use cross-entropy loss to compute $\mathcal{L}_{llm}$. This process is just similar to the loss functions adopted in DETR [13] and MaskFormer [31], with the validity score for each polygon serving as the role of classification score used in [13, 31] for bipartite matching. For $\mathcal{L}_{ref}$, we use cross-entropy loss with online hard examples mining (OHEM) strategy to



(a) Wrongly Predicted Local Regions

(c) Validity Scores for Local Regions

(b) Correctly Predicted Complete Objects

(d) Validity Scores for Complete Objects

FIGURE 4.8: Illustration of the oversegmentation hallucination problems (a) and the distribution of v_n (green lines) and $\widetilde{v}_n$ (purple lines) for local regions (c) and complete objects (d). Best viewed in color.

compute it. Note that to compute the loss $\mathcal{L}_{llm}$, it is essential to obtain instance-wise ground truth objects for bipartite matching. There exist two methods to acquire this object ground truth: (1) Utilizing the instance segmentation annotations provided by the Pascal VOC and COCO datasets directly, or (2) Considering each connected component within the semantic segmentation ground truth mask as a separate instance object. Given the constraints of few-shot segmentation, where only semantic-level ground truth is accessible while instance-level annotations are not, we adopt the latter approach. Although this method of deriving object ground truth may introduce some inaccuracies, it is observed to already yield sufficiently satisfactory training results. If we can use the more accurate instance-level annotations for training, LLaFS++'s 1-shot performance on PASCAL-5^i and COCO-20^i could be further improved by 1.3% and 0.8%, respectively.

Inference with Hallucination Mitigation

During the inference stage, we treat the annotated image of a new class as the support image and the test image to be segmented as the query image. These images are fed into the LLaFS++ framework as shown in Figure 4.1 to generate the segmentation results. While this straightforward method typically produces satisfactory outcomes, as shown in Figure 4.8(a), we occasionally face a hallucination issue where the model incorrectly segments incomplete local regions of the target object rather than capturing its entirety. This problem may arise from the utilization of the fine-grained in-context instruction as described in Sec.4.2.3, where some locally regional characteristics of the target class and their corresponding local regions within the support image are input into the LLM for guidance, which could potentially mislead the LLM to focus on segmenting only local regions of the target class, thus resulting in the hallucination issue. To address this problem, we introduce a contrastive prediction approach for model inference. Specifically, consider the LLM ϕ with L layers; denote the first $L-1$ layers of the LLM as $\phi_{[1:L-1]}$ and the final layer as $\phi_{[L]}$. Within $\phi_{[L]}$, we use self-attention masks to block all hidden states of [class] tokens and [support ground truth] tokens, thereby preventing other tokens from seeing these tokens describing the target class's global information. In this way, we create a locally-biased layer denoted as $\hat{\phi}_{[L]}$. For the n-th polygon embedding $\mathbf{P}_n$, we denote its validity score (refer to Sec.4.2.4 for details) computed from $[\phi_{[1:L-1]}, \phi_{[L]}]$ as v_n and that from $[\phi_{[1:L-1]}, \hat{\phi}_{[L]}]$ as $\hat{v}_n$. In the training stage, as illustrated in Sec.4.2.4, we directly use v_n to verify the validity of the polygon; whereas in the inference stage, we calculate another score $\widetilde{v}_n$ for this verification by:

$$\widetilde{v}_n = v_n + (v_n - \hat{v}_n) = 2v_n - \hat{v}_n. \tag{4.6}$$

Subsequently, polygons with $\widetilde{v}_n < 0$ are excluded. This procedure incorporates a contrastive element $(v_n - \hat{v}_n)$ during the inference phase, a strategy drawing inspiration from the success of contrastive decoding [78] in NLP. Specifically, within the LLM's input instructions, the [class] tokens denote the class name, while the [support ground truth] tokens describe the entire area belonging to the target class within the support image. These tokens collectively represent the holistic or global information of the target class. When such global tokens are masked out within $\hat{\phi}_{[L]}$, the remaining information is primarily related to the localized attributes of the target class. Relying solely on such locally-focused information inherently increases the validity scores of the predicted local regions, while reducing those for polygons that enclose the entire target object. Therefore, a higher contrastive value $(v_n - \hat{v}_n)$ can reflect a greater possibility that the n-th polygon represents the entire object. By combining this contrastive element with v_n and using the combined score $\widetilde{v}_n$ to assess the polygons, we can filter out those that represent just local regions of the target object, thus mitigating the aforementioned hallucination problem. The score distributions shown in Figure 4.8 demonstrate the effectiveness of $\widetilde{v}_n$, which shows that compared to v_n, a greater number of $\widetilde{v}_n$ for the incorrectly predicted local regions fall below 0. Conversely, for the correctly predicted complete objects, the distribution exhibits an opposite trend. Note that our inference method does not require adding any extra parameters; it only necessitates a slight 3% increase in computational cost for an additional run of the LLM's last layer. Therefore, our approach can enhance performance at almost zero cost, as demonstrated by the experimental results presented in Table 4.6.

4.2.7 Extension to Multi-shot Setting

In the above sections, we introduce LLaFS++ under the one-shot setting. The method for the multi-shot setting can be easily extended from the one-shot method. Specifically, for each support image, we extract a set of visual tokens and a region-attribute corresponding table using the method illustrated in the main paper. These pieces of information from all N_s support images are then incorporated into the following instruction for feeding into the LLM: *For each object within the class* [class] *in an image, output coordinates of a 16-sided polygon that encloses the object. These points should be arranged in a clockwise direction. The output should be a tuple in the format of (c1, c2, ..., cn), where cn is the coordinates for the n-th object and its format should be ((x1,y1),(x2,y2),...,(x16,y16)). The coordinate value should be within* [image size]*. To accomplish this task, you can refer to the following properties of* [class]*:* [class] *has* [attributes]*. For example, for image* [support image 1]*, the output should be* [support ground truth 1]*, because in these regions,* $[\text{cor}]_1$ *is* $[\text{att}]_1$*,*

[cor]$_2$ *is* [att]$_2$, *...* , [cor]$_{N_a}$ *is* [att]$_{N_a}$*;...; for image* [support image N_s], *the output should be* [support ground truth N_s], *because in these regions,* [cor]$_1$ *is* [att]$_1$, [cor]$_2$ *is* [att]$_2$, *...,* [cor]$_{N_a}$ *is* [att]$_{N_a}$. *For image* [query image], *what is the output?*

4.3 Experiments

4.3.1 Datasets and Metrics

We evaluate our method on three commonly used datasets: PASCAL-5^i [119], COCO-20^i [103], and FSS-1000 [77]. The PASCAL-5^i dataset is comprised of images sourced from the PASCAL VOC 2012 dataset with annotations extended by the SDS dataset. COCO-20^i is proposed in [103] and built based on MSCOCO. Following previous work [188, 160, 94, 33], we employ a cross-validation strategy for our experiments. Specifically, we divide the total classes of each dataset into four equal subsets, using three subsets for training and the remaining one subset for testing in each experiment. In this way, for PASCAL-5^i, we have 15 classes for training and 5 classes for testing, and for COCO-20^i, we have 60 classes for training and 20 classes for testing in each experiment. This approach results in four sets of experimental results along with their mean result for each dataset. The FSS-1000 dataset contains images of 1000 classes, of which 486 classes are new classes not present in previous benchmarks. The overall classes in FSS-1000 are divided into 520, 240, and 240 classes for training, validation, and testing, respectively. Following previous methods [94], we report the results on the test set. We use two widely-adopted metrics for evaluation, including mean intersection-over-union (mIoU) and foreground-background IoU (FB-IoU).

4.3.2 Implementation Details

We set the threshold α in Eq.4.1 to 0.22, and the number N of polygon embedding $\mathbf{P}_n$ to 15. The ground truth of polygon vertices is obtained in polar coordinates [154]. Specifically, starting from the object center, 16 rays are uniformly emitted at equal angular intervals $\triangle\theta = 22.5°$. The points of intersection between these rays and the object contour are taken as the ground truth of the polygon vertices. AdamW is used as the optimizer with the cosine annealing schedule and an initial learning rate of 0.0002. The model is trained on A100 GPUs. ResNet50 and ResNet101 from the CLIP are used as the image encoder. In ResNet, the output features from stage 3 and stage 4 are resized to 1/8 of the input size and concatenated with the output features from stage 2. This combined feature is used as the input for the Q-former and the pixel decoder in the refinement network. The Q-former has 8 layers with the dimension of 384. The number of queries in the Q-former is set to 100. The input for the text transformer in the Q-former is ‘a photo of [class].’. Feature dimension in the region encoding network and refinement network is 128. The batch size is 32 and the input image size is (384, 384).

Backbone	Method	Venue	1-shot						5-shot					
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU	Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU
ResNet50	NTRENet	CVPR'22	65.4	72.3	59.4	59.8	63.2	77.0	66.2	72.8	61.7	62.2	65.7	78.4
	BAM	CVPR'22	69.0	73.6	67.5	61.1	67.8	79.7	70.6	75.1	70.8	67.2	70.9	82.2
	AAFormer	ECCV'22	69.1	73.3	59.1	59.2	65.2	73.8	72.5	74.7	62.0	61.3	67.6	76.2
	SSP	ECCV'22	60.5	67.8	66.4	51.0	61.4	-	67.5	72.3	75.2	62.1	69.3	-
	IPMT	NeurIPS'22	72.8	73.7	59.2	61.6	66.8	77.1	73.1	74.7	61.6	63.4	68.2	81.4
	ABCNet	CVPR'23	68.8	73.4	62.3	59.5	66.0	76.0	71.7	74.2	65.4	67.0	69.6	80.0
	HDMNet	CVPR'23	71.0	75.4	68.9	62.1	69.4	-	71.3	76.2	71.3	68.5	71.8	-
	MIANet	CVPR'23	68.5	75.8	67.5	63.2	68.7	79.5	70.2	77.4	70.0	68.8	71.7	82.2
	MSI	ICCV'23	71.0	72.5	63.8	65.9	68.3	79.1	73.0	74.2	66.6	70.5	71.1	81.2
	SCCAN	ICCV'23	68.3	72.5	66.8	59.8	66.8	77.7	72.3	74.1	69.1	65.6	70.3	81.8
	AMFormer	NeurIPS'23	71.1	75.9	69.7	63.7	70.1	-	73.2	77.8	73.2	68.7	73.2	-
	HPA	T-PAMI'23	67.5	72.4	65.2	56.7	65.4	76.4	71.2	73.9	68.8	63.8	69.4	81.1
	BAM-final	T-PAMI'23	69.2	74.7	67.8	61.7	68.3	80.3	71.8	75.7	72.0	67.5	71.8	83.1
	PFENet++	T-PAMI'24	63.3	71.0	65.9	59.6	64.9	76.8	66.1	75.0	74.1	64.3	69.9	81.1
	LLaFS	CVPR'24	74.2	78.8	72.3	68.5	73.5	84.8	75.9	80.1	75.8	70.7	75.6	85.3
	LLaFS++	-	**77.8**	**82.1**	**75.8**	**72.9**	**77.2**	**86.7**	**79.7**	**83.6**	**77.9**	**73.8**	**78.8**	**87.7**
ResNet101	NTRENet	CVPR'22	65.5	71.8	59.1	58.3	63.7	75.3	67.9	73.2	60.1	66.8	67.0	78.2
	DCAMA	ECCV'22	65.4	71.4	63.2	58.3	64.6	77.6	70.7	73.7	66.8	61.9	68.3	80.8
	VAT	ECCV'22	70.0	72.5	64.8	64.2	67.9	79.6	75.0	68.4	69.5	72.0	83.2	
	ABCNet	CVPR'23	65.3	72.9	65.0	59.3	65.6	78.5	71.4	75.0	68.2	63.1	69.4	80.8
	MSI	ICCV'23	73.1	73.9	64.7	68.8	70.1	82.3	73.6	76.1	68.0	71.3	72.2	82.3
	SCCAN	ICCV'23	70.9	73.9	66.8	61.7	68.3	78.5	73.1	76.4	70.3	66.1	71.5	82.1
	AMFormer	NeurIPS'23	71.3	76.7	70.7	63.9	70.7	-	74.4	78.5	74.3	67.2	73.6	-
	HPA	T-PAMI'23	67.2	73.1	64.3	59.8	66.1	76.6	68.3	75.2	66.4	67.8	69.4	80.4
	BAM-final	T-PAMI'23	69.9	75.4	67.1	62.1	68.6	80.2	72.6	77.1	70.7	69.8	72.5	84.1
	PFENet++	T-PAMI'24	63.1	72.4	63.4	62.2	65.3	75.5	67.2	76.1	75.5	67.2	71.5	82.7
	LLaFS	CVPR'24	75.0	79.3	72.9	69.4	74.1	85.1	77.0	81.1	76.5	72.1	76.7	85.8
	LLaFS++	-	**78.8**	**82.4**	**76.2**	**73.2**	**77.7**	**87.2**	**80.5**	**84.4**	**78.7**	**74.8**	**79.6**	**88.4**

TABLE 4.1: Performance comparison with other methods on PASCAL-5^i.

4.3.3 Main Results

Comparison with Few-shot Segmentation methods

In this section, we compare our method with existing state-of-the-art few-shot segmentation techniques on three datasets: PASCAL-5^i, COCO-20^i, and FSS-1000, with the results presented in Table 4.1, Table 4.2, and Table 4.3, respectively. To evaluate the generalization capabilities of our approach, we report comparative results utilizing two different backbone scales: ResNet50 and ResNet101. All the compared methods are advanced approaches published at top conferences (CVPR, ICCV, NeurIPS, etc.) or in top journals (T-PAMI) within the recent two years. Our method displays superior performance across all datasets and experimental settings, consistently outperforming the existing approaches and showing a significant enhancement over the previous state-of-the-art results. For instance, using the ResNet50 backbone on the PASCAL-5^i dataset, our conference version, LLaFS [188], achieves an mIoU of 73.5% and an FB-IoU of 84.8%, significantly surpassing the runner-up by 3.4% and 4.5%, respectively. In this chapter, we introduce LLaFS++, which incorporates additional improvements and thus achieving an even higher mIoU of 77.1% and FB-IoU of 86.7% in the 1-shot scenario, which surpasses LLaFS by 3.6% and 1.9%, and outperforms the previous best results by 7.0% and 6.4%, respectively. Considering the more demanding COCO-20^i dataset, which presents a bigger challenge due to a greater number of classes and more diverse images, our method shows even higher advantages, outperforming the previous state-of-the-art techniques by 8.3% in mIoU and 9.3% in FB-IoU for the 1-shot scenario using the ResNet101 backbone. It is worth noting that MIANet [164], a method we

Backbone	Method	Venue	1-shot						5-shot					
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU	Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU
ResNet50	NTRENet	CVPR'22	36.8	42.6	39.9	37.9	39.3	68.5	38.2	44.1	40.4	38.4	40.3	69.2
	BAM	CVPR'22	43.4	50.6	47.5	43.4	46.2	67.4	49.3	54.2	51.6	49.6	51.2	71.9
	SSP	ECCV'22	35.5	39.6	37.9	36.7	47.4	-	40.6	47.0	45.1	43.9	44.1	-
	AAFormer	ECCV'22	39.8	44.6	40.6	41.4	41.6	67.7	42.9	50.1	45.5	49.2	46.9	68.2
	MM-Former	NeurIPS'22	40.5	47.7	45.2	43.3	44.2	-	44.0	52.4	47.4	50.0	48.4	-
	IPMT	NeurIPS'22	41.4	45.1	45.6	40.0	43.0	-	43.5	49.7	48.7	47.9	47.5	-
	ABCNet	CVPR'23	42.3	46.2	46.0	42.0	44.1	69.9	45.5	51.7	52.6	46.4	49.1	72.7
	HDMNet	CVPR'23	43.8	55.3	51.6	49.4	50.0	72.2	50.6	61.6	55.7	56.0	56.0	77.7
	MIANet	CVPR'23	42.5	53.0	47.8	47.4	47.7	71.5	45.8	58.2	51.3	51.9	51.7	73.1
	MSI	ICCV'23	42.4	49.2	49.4	46.1	46.8	-	47.1	54.9	54.1	51.9	52.0	-
	SCCAN	ICCV'23	40.4	49.7	49.6	45.6	46.3	69.9	47.2	57.2	59.2	52.1	53.9	74.2
	AMFormer	NeurIPS'23	44.9	55.8	52.7	50.6	51.0	72.9	52.0	61.9	57.4	57.9	57.3	78.8
	HPA	T-PAMI'23	41.0	46.9	44.3	43.2	43.8	68.3	46.2	56.2	49.2	50.4	50.5	71.4
	BAM-final	T-PAMI'23	43.9	51.4	47.9	44.5	46.9	72.3	49.8	55.4	52.3	50.2	51.9	74.7
	PFENet++	T-PAMI'24	40.9	46.0	42.3	40.1	42.3	65.7	47.5	53.3	47.3	46.4	48.6	70.3
	LLaFS	CVPR'24	47.5	58.8	56.2	53.0	53.9	75.2	53.2	63.8	63.1	60.0	60.0	79.5
	LLaFS++	-	**50.8**	**62.7**	**60.2**	**56.4**	**57.5**	**78.8**	**53.9**	**64.9**	**63.8**	**61.1**	**60.9**	**79.9**
ResNet101	NTRENet	CVPR'22	38.3	40.4	39.5	38.1	39.1	67.5	42.3	44.4	44.2	41.7	43.2	69.6
	SSP	ECCV'22	39.1	45.1	42.7	41.2	42.0	-	47.4	54.5	50.4	49.6	50.2	-
	IPMT	NeurIPS'22	40.5	45.7	44.8	39.3	42.6	-	45.1	50.3	49.3	46.8	47.9	-
	ABCNet	CVPR'23	36.5	35.7	34.7	31.4	34.6	59.2	40.1	40.1	39.0	35.9	38.8	62.8
	MSI	ICCV'23	44.8	54.2	52.3	48.0	49.8	-	49.3	58.0	56.1	52.7	54.0	-
	SCCAN	ICCV'23	42.6	51.4	50.0	48.8	48.2	69.7	49.4	61.7	61.9	55.0	57.0	74.8
	AMFormer	NeurIPS'23	40.5	45.7	44.8	39.3	42.6	-	45.1	50.3	49.3	46.8	47.9	-
	HPA	T-PAMI'23	43.2	50.5	45.5	46.2	46.3	68.8	49.4	58.4	52.5	50.9	52.8	74.4
	BAM-final	T-PAMI'23	45.2	55.1	48.7	45.0	48.5	69.9	48.3	58.4	52.7	51.4	52.7	74.1
	PFENet++	T-PAMI'24	42.0	44.1	41.0	39.4	41.6	65.4	47.3	55.1	50.1	50.1	50.7	70.9
	LLaFS	CVPR'24	48.1	59.3	56.5	53.6	54.4	75.6	53.2	64.1	63.3	60.2	60.2	79.6
	LLaFS++	-	**51.1**	**63.0**	**61.4**	**56.9**	**58.1**	**79.2**	**54.2**	**65.2**	**63.9**	**61.3**	**61.1**	**80.0**

TABLE 4.2: Performance comparison with other methods on COCO-20^i.

compare against, also employs a language model (word2vec) to facilitate few-shot segmentation. Our method, different from MIANet, leverages the more powerful large language model (LLM) complemented by several innovative and task-specific designs that enhance the LLM's capability in addressing the few-shot segmentation problem. Particularly, our fine-grained in-context instruction delves deeper into how to better integrate textual and visual information from language models and annotated images for getting a better multimodal guidance. With these novel designs, our method significantly outperforms MIANet by 8.5% mIoU on PASCAL-5^i. These results demonstrate the excellent performance of our LLaFS++ and highlight the huge potentiality of using LLMs to tackle few-shot segmentation.

Backbone	Method	Venue	1-shot	5-shot
ResNet50	MSI	ICCV'23	90.0	90.6
	PFENet++	T-PAMI'24	88.6	89.1
	LLaFS	CVPR'24	92.3	92.8
	LLaFS++	-	**93.2**	**93.5**
ResNet101	MSI	ICCV'23	90.6	91.0
	PFENet++	T-PAMI'24	88.6	89.2
	LLaFS	CVPR'24	92.7	93.0
	LLaFS++	-	**93.4**	**93.8**

TABLE 4.3: Performance comparison on FSS-1000.

Comparison with LLM-based Segmentation Methods

We further compare our method with other LLM-based segmentation techniques to highlight the superior advantages of our LLaFS++ framework. We choose two recently published advanced algorithms from CVPR 2024 for this comparison: LiSA [70] and

PixelLM [116] and the comparative results are shown in Table 4.4. Given that these methods are not originally designed for few-shot segmentation tasks, we perform some minor adjustments to their model structures to better suit the task. Specifically, on top of their existing textual input, we include support image features and support ground truth features as additional inputs for the language models. The support image features are obtained directly from the CLIP encoder, while the support ground truth features are acquired through SAM's [69] prompt encoder. After incorporating these changes, we retrain the altered models on few-shot segmentation datasets, allowing us to fairly evaluate their performance against our LLaFS++ framework. As shown in Table 4.4, we note that although LiSA and PixelLM also employ LLMs with a 7B parameter size, their performance on all three datasets is significantly worse than that of LLaFS++. This is because our LLaFS++ contains several task-tailored designs such as the novel instructions and pseudo-sample-based pretraining mechanisms, which enable the LLM to handle few-shot segmentation more effectively. The results demonstrate the excellent performance of LLaFS++ in comparison to other LLM-based segmentation methods. It also suggests that the superior performance of LLaFS++ is NOT attributable only to the use of an LLM, but is also a result of our carefully designed, innovative, and task-tailored methods that enhance the LLM's ability to process few-shot segmentation.

Method	Venue	PASCAL-5^i		COCO-20^i		FSS-1000	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
LISA	CVPR'24	70.2	73.7	51.9	58.0	90.5	91.1
PixelLM	CVPR'24	69.5	73.2	51.0	57.2	90.0	90.5
LLaFS	CVPR'24	74.1	76.7	54.4	60.2	92.7	93.0
LLaFS++	-	**77.7**	**79.6**	**58.1**	**61.1**	**93.4**	**93.8**

TABLE 4.4: Comparison with LLM-based segmentation methods.

4.3.4 Ablation Study

In this section, we perform several ablation studies to verify the effectiveness of the proposed designs and components in our LLaFS++. The experiments are conducted on PASCAL-5^i Fold-0 with the ResNet50 backbone and 1-shot scenario.

Effectiveness of Key Components

To enhance the LLM's capability in handling few-shot segmentation, we propose several novel designs in this work including (1) the segmentation task instruction, (2) the fine-grained in-context instruction, (3) the refinement network, and (4) the pseudo-sample-based curriculum pretraining. These innovative designs work together within our LLaFS++ framework to achieve high-performance few-shot segmentation. To evaluate the contribution of each component, we conduct a series of ablation experiments with the results presented in Table 4.5. We observe that replacing the detailed segmentation task instruction with an abstract summary 'perform image segmentation' decreases the mIoU by 6.5%. Not using the other components can also lead to a significant drop in performance, demonstrating their importance and effectiveness. It is important to note that even without the refinement network, directly using the polygons outputted by the LLM as the final segmentation results still yields quite good performance (74.3% mIoU) that outperforms previous SOTA (71.1% mIoU) significantly.

Method	mIoU	FB-IoU
LLaFS++	77.8	87.1
LLaFS++ w/o segmentation task instruction w/ abstract summary	73.9	84.9
LLaFS++ w/o fine-grained in-context instruction	70.2	80.6
LLaFS++ w/o refinement network	74.3	84.8
LLaFS++ w/o pseudo-sample-based curriculum pretraining	67.5	77.2

TABLE 4.5: Effectiveness of different components in the LLaFS framework.

Effectiveness of Extension Compared to LLaFS

Our LLaFS++ proposed in this chapter extends its conference version of LLaFS in two significant aspects: (1) a better method for aligning image regions with class attributes to build the region-attribute corresponding table, and (2) a contrastive prediction method for inference to mitigate hallucinations. As presented in Table 4.6, discarding these enhancements and reverting to the original methods used in LLaFS results in a significant decrease in performance, which validates the high effectiveness of our extended approaches. Also note that the region-attribute alignment method used in LLaFS++ (including the process of region encoding network REN) reduces computational load by over 85% compared to LLaFS, since it no longer requires extracting a CLIP feature for every cropped image. Furthermore, we conduct a more detailed evaluation. Specifically, we randomly select 500 images from the PASCAL-5^i test set, and use the methods in LLaFS and LLaFS++ to extract the alignment results between image regions (cropped image in LLaFS and similarity map M in LLaFS++) and class attributes. It is challenging to directly assess these results' quality since we do not have ground truth for such alignment. Thus, we instead conduct a user study by inviting six volunteers, who are completely unrelated to this research, to rate each test image's alignment result as 'bad', 'medium', or 'good'. The average results of the six raters are presented in Figure 4.9, which indicates the significant advantages of LLaFS++ proposed in this paper. Additionally, we also calculate the frequency of oversegmentation hallucination occurring in all the test images of the PASCAL-5^i dataset in both LLaFS and LLaFS++. Oversegmentation hallucination refers to the issue where the model incorrectly segments multiple local regions of the target object instead of capturing it as a whole (See Sec.4.2.6 for details). As shown in Table 4.6, using the inference method in LLaFS++ reduces this frequency (FH) from 11.8% to 3.9%, demonstrating the effectiveness of our approach.

Effectiveness of Large Language Models

With extensive prior knowledge and powerful few-shot capabilities, the large language model (LLM) contributes significantly to the high effectiveness of our LLaFS++ framework. To validate the importance of the LLM within our model, we conduct an experiment by excluding the LLM from LLaFS++ and evaluate the performance of a modified model that is composed of the remaining parts of LLaFS++. More specifically, in

Method	mIoU	GFLOPs
RAA of LLaFS++	77.8	66
RAA of LLaFS	75.8	485

Method	mIoU	FH
IN of LLaFS++	77.8	3.9%
IN of LLaFS	75.0	11.8%

TABLE 4.6: Effectiveness of extension compared to LLaFS. 'RAA': region-attribute alignment method; GFLOPs represent the computations for the RAA process; 'IN': inference method; 'FH': the frequency of over-segmentation hallucinations occurring in all test images.

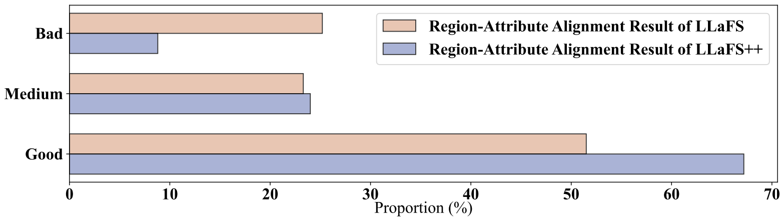


FIGURE 4.9: Six volunteers' average scores regarding the quality of the region-attribute alignment results for 500 randomly sampled images.

constructing this model 'LLaFS++ w/o LLM', we perform the following alterations to ensure that few-shot segmentation could be executed solely with the remaining components: (1) In the process of utilizing the Q-former to extract visual tokens from the support image, we replace the vanilla cross-attention that interacts the learned queries with support image features by the masked attention as in [31]. This modification guarantees that the derived support image tokens are only associated with the foreground area where the target category is located. (2) Subsequent to the Q-former, we introduce an additional cross-attention step to enable interactions between support image tokens and query image tokens, thereby allowing the query image tokens to incorporate reference information from the support foreground. The query image tokens resultant from this step are then leveraged as input query embeddings for the transformer decoder in the refinement network, which produces the segmentation result. As shown in Table 4.7, removing LLM significantly decreases mIoU by 15.5% compared to the complete LLaFS++, demonstrating the crucial role of the LLM in ensuring the high performance of our framework. In our method, we employ CodeLlama instead of the vanilla Llama as the large language model. This is because CodeLlama finetuned with code generation datasets is more killed in generating structured information like the segmentation result in our task. This is demonstrated by the result presented in Table 4.7, which shows that the performance of using CodeLlama is 4.4% better than Llama.

Method	mIoU	FB-IoU
LLaFS++	77.8	87.1
LLaFS++ w/o LLM	62.3	73.7
LLaFS++ (CodeLlama)	77.8	87.1
LLaFS++ (Llama2)	74.3	84.5

TABLE 4.7: Effectiveness of large language model.

Effectiveness of Support Images

Based on the task setting of few-shot segmentation, we leverage a small number of annotated images, called support images, to provide visual reference information for guiding the segmentation process. In fact, LLM-based segmentation models can also perform segmentation in an open-vocabulary manner, that is, to segment a category by solely utilizing its class name but without the need to apply any annotated support image. To evaluate the impact of support images on enhancing the model's effectiveness, we conduct experiments to compare with such an open-vocabulary and support-image-free method, with the results presented in Table 4.8. When we keep the training schema unchanged yet removing the support image and its ground truth mask from the input during inference, there is an observed decrease in the model's mIoU by 14.3%. If these elements are also excluded from the training phase, this gap can be reduced to 12.2% since the training and inference inputs become aligned, but the result is still significantly worse than the original LLaFS++. These results demonstrate that our LLaFS++ benefits not solely from LLM's prior knowledge in an open-vocabulary manner but indeed gains further improvement from the provided few-shot samples. Moreover, we investigate a scenario within an in-vocabulary setting, where the categories in testing are the same as the categories in training. Concretely, we employ all 20 classes in PASCAL-5^i for training and also apply all these classes for testing. In this scenario, LLaFS++ still significantly outperforms the methods that do not leverage support images, indicating that incorporating a small number of annotated samples during evaluation can effectively enhance model performance, even for categories that have already been well trained with extensive training data. Such results demonstrate the crucial role of support images within our few-shot segmentation framework.

Tr w/ SI	In w/ SI	In-vocab	mIoU	FB-IoU
✓	✓	✗	77.8	87.1
✓	✗	✗	63.5	75.0
✗	✗	✗	65.6	77.3
✓	✓	✓	91.2	94.6
✗	✗	✓	89.1	93.0

TABLE 4.8: Effectiveness of support images. 'Tr': training; 'In': inference; 'SI': support images. 'In-vocab': the scenario where the categories in testing are the same as the categories in training.

Ablation of Fine-grained In-context Instruction

The fine-grained in-context instruction constitutes a crucial component of our LLaFS++ framework, which combines visual information from support images with textual cues from the LLM's pretrained knowledge to form a comprehensive reference that can guide the segmentation of query images effectively. We conduct a thorough evaluation of various components and designs within this instruction and present the results in Table 4.9. As illustrated in Sec.4.2.3, the fine-grained in-context instruction is primarily made up of two parts: attributes of the target class and a region-attribute corresponding table derived from the support image. Table 4.9 shows that excluding these components can respectively decrease the mIoU by 3.6% and 5.5%, demonstrating the importance of these reference information in guiding the segmentation of query images. We also evaluate the detailed designs within this instruction, including (1) the thresholding procedure (Eq.4.1) to exclude support-non-matching attributes, (2) the instruction refinement framework to resolve class ambiguities (Sec.4.2.3), and (3) the iterative execution of this refinement. Table 4.9 shows that removing any of these designs will cause a significant reduction in performance, thus demonstrating their important contributions to enhancing model effectiveness.

Method	mIoU	FB-IoU
LLaFS++	77.8	87.1
LLaFS++ w/o class attributes	74.2	84.8
LLaFS++ w/o region-attribute corresponding table	72.3	82.9
LLaFS++ w/o thresholding procedure in Eq.4.1	73.9	84.7
LLaFS++ w/o instruction refinement	74.5	85.0
LLaFS++ w/o iterative refinement	75.8	85.7

TABLE 4.9: Ablation study of fine-grained in-context instruction.

Ablation of Pseudo-sample-based Curriculum Pretraining

In Sec.4.2.5, we present a method for creating pseudo support-query pairs to expand the training dataset for few-shot segmentation. Additionally, we propose a curriculum learning-based strategy to address difficulties in model training convergence. To validate the effectiveness of these methods, we conduct several ablation study experiments and present the results in Table 4.10. For pseudo sample synthesis, we investigate two crucial aspects: (1) When the pseudo-sample-based pretraining is excluded, we observe an mIoU drop of 10.3%. (2) When generating pseudo support-query samples, to ensure that the support and query can reflect the same category, the contour and the mean value of foreground noise used to generate the query image are adjusted based on those used for generating the support image. When this strategy is not employed and random generation is used instead, the mIoU decreases by 9.0%. We also evaluate the proposed curriculum pretraining strategy that progressively increases the pretraining tasks' difficulty in the following aspects: (1) image understanding, (2) polygon generation, in which the difficulty increase of image understanding is implemented by (a) increasing the difference between support foreground and query foreground, and (b) reducing the difference between foreground and background within each image. Excluding either of these methodologies would cause a significant performance decline, demonstrating their importance and necessity in our framework. Beyond applying curriculum-based polygon generation to synthetic images during the pretraining stage, we also examine its further application to realistic data during the training phase. We observe that such an extension does not significantly improve performance. A possible explanation is that the model has already acquired sufficient ability to generate 16-vertex coordinates through curriculum pretraining with pseudo samples, so it no longer requires the continued application of this curriculum method in the subsequent training stage. Therefore, we only use this strategy during pretraining.

Method	mIoU	FB-IoU
LLaFS++	77.8	87.1
LLaFS++ w/o pseudo samples	67.5	77.2
LLaFS++ w/ random pseudo query generation	67.9	77.2
LLaFS++ w/o curriculum strategy	71.6	82.0
LLaFS++ w/o curriculum strategy in image understanding	75.0	85.2
LLaFS++ w/o curriculum strategy in polygon generation	73.2	83.5
LLaFS++ w/o increasing SF-QF difference	75.9	85.6
LLaFS++ w/o reducing F-B difference	75.6	85.8
LLaFS++ + curriculum polygon generation in training	78.0	87.1

TABLE 4.10: Ablation study of pseudo-sample-based curriculum pretraining. 'SF', 'QF', 'F', 'B' respectively refer to support foreground, query foreground, foreground, background.

Settings of Hyper-parameter α

As illustrated in Sec.4.2.3, it is observed that not every attribute extracted for the target class can find a corresponding region in the support foreground. To prevent the introduction of misleading information due to this issue, we use a thresholding method to exclude the regional features calculated from attributes that do not correspond with the support. As illustrated in Eq.4.1, this process is made possible by a predefined threshold α. We experiment with different values for α to find the optimal choice and present the results in Figure 4.10. It is observed that both excessively small and large values for α can decrease the mIoU. This might be due to the fact that an excessively small value of α could lead to a false positive problem, where non-matching attributes may be erroneously classified as matching; while an excessively large value of α could lead to a false negative issue, where matching attributes are incorrectly deemed non-matching. Both conditions can adversely affect the quality of the generated region-attribute corresponding table. Based on the results shown in Figure 4.10, we choose $\alpha = 0.22$ as the threshold setting in our framework.

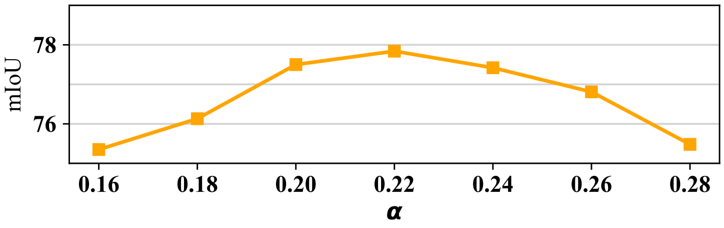


FIGURE 4.10: Performance of using different values for the threshold α in Eq.1

Number of Polygon's Sides

In our framework, we represent the segmentation output mask as a region enclosed by a polygon with 16 sides. We find that the number of polygon's sides, denoted as P, can have an impact on the model's performance. To determine the optimal setting for the number of polygon sides, we evaluate the relationship between mIoU and P

and present the results in Figure 4.11. We observe that when P is small, the model's performance is suboptimal. This is because polygons with a too small number of sides cannot accurately describe object edges so the enclosed region is not precise enough. As P increases from 8 to 16, the mIoU gradually improves from 72.1% to 77.8%. However, when P exceeds 16, we observe a slight decrease in performance when P continues to increase. This could be because a larger P increases the task's complexity for LLM to tackle. Based on the results, we chose $P = 16$ as our setting.

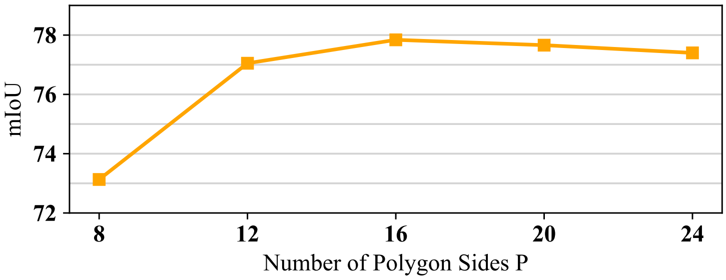


FIGURE 4.11: Ablation for the number of polygons' sides.

Number of Polygon Embeddings

We vary the number N of polygon embeddings $\mathbf{P}_n$ from 5 to 25 to investigate its impact on segmentation performance. As presented in Figure 4.12, the model consistently achieves stable and excellent performance when $N > 15$. The results demonstrate the high robustness of our method.

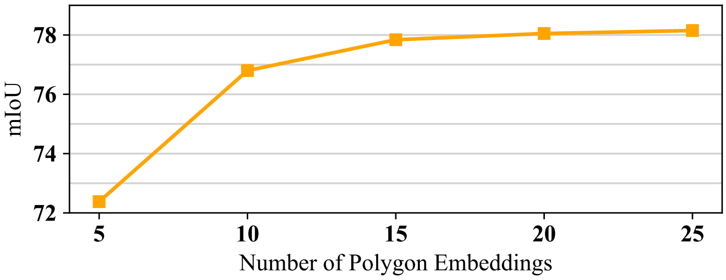


FIGURE 4.12: Ablation for the number of polygon embeddings.

Hyper-parameter Settings for Pseudo-sample-based Curriculum Pretraining

As discussed in detail in Sec.4.2.5, our proposed pseudo-sample-based curriculum pretraining involves four hyper-parameters $(a_0, b_0, c_{N_p}, d_{N_p})$. The results for different combinations of these hyper-parameters are presented in Table 4.11. It can be observed that our method can consistently achieve excellent and similar results across different $(a_0, b_0, c_{N_p}, d_{N_p})$ settings. These results demonstrate that the performance of LLaFS++ is NOT sensitive to these hyper-parameters, showing the high robustness of our method.

$(a_0, b_0, c_{N_p}, d_{N_p})$	mIoU
(100, 150, 50, 100)	77.8
(75, 125, 75, 125)	77.5
(125, 175, 25, 75)	78.0
(100, 150, 75, 125)	77.7
(75, 125, 50, 100)	77.5

TABLE 4.11: Different settings for hyper-parameters $(a_0, b_0, c_{N_p}, d_{N_p})$

Effectiveness of Noise-enabled Augmentation

As discussed in the method sections, we adopt a noise-enabled augmentation method to enhance the pretraining effectiveness. As presented in Table 4.7, compared to using the original image captioning dataset, training with data augmented by this method results in an increase of 3.8% in mIoU, demonstrating its high effectiveness in enhancing segmentation performance.

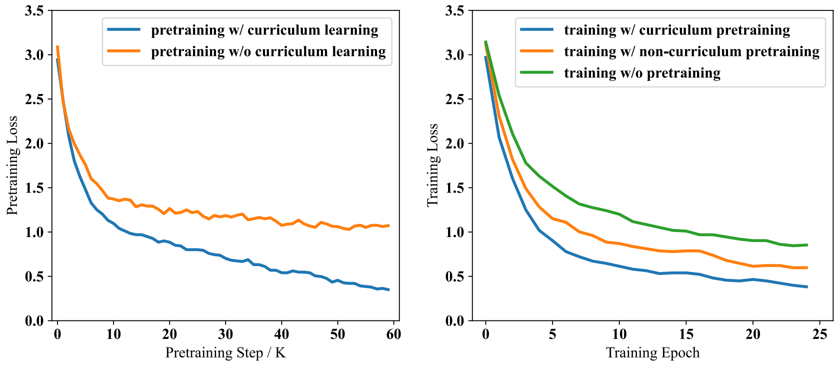
Method	mIoU
w/o NA	74.0
w/ NA	77.8

TABLE 4.12: Effectiveness of noise-enabled augmentation Method. 'NA': the noise-enabled augmentation method.

4.3.5 Loss Curves

In Sec.4.2.5, we introduce a curriculum-learning-based method to accelerate the optimization convergence of our model. To evaluate the effectiveness of this approach, we compare the loss curves for models pretrained with and without this curriculum learning strategy. The results presented in Figure 4.13(a) indicate that without the use of curriculum learning, the pretraining task becomes excessively challenging, which causes the model optimization to quickly reach a bottleneck with difficulties in the further convergence. After utilizing curriculum learning, this issue is significantly alleviated and the model can continuously converge. In Figure 4.13(b), we further present a comparison of the loss reduction conditions during the training phase after using different pretraining methods: pretraining with curriculum learning, pretraining without curriculum learning, and no pretraining at all. The model that has not undergone any pretraining is observed to have the lowest convergence rate, while the model pretrained with the curriculum learning strategy shows the swiftest convergence in the training phase, which demonstrates the effectiveness of our proposed curriculum-based pretraining method. In Sec.4.2.5, we introduce a curriculum-learning-based method to accelerate the optimization convergence of our model. To evaluate the effectiveness of this approach, we compare the loss curves for models pretrained with and without this curriculum learning strategy. The results presented in Figure 4.13(a) indicate that without the use of curriculum learning, the pretraining task becomes excessively challenging, which causes the model optimization to quickly reach a bottleneck with

difficulties in the further convergence. After utilizing curriculum learning, this issue is significantly alleviated and the model can continuously converge. In Figure 4.13(b), we further present a comparison of the loss reduction conditions during the training phase after using different pretraining methods: pretraining with curriculum learning, pretraining without curriculum learning, and no pretraining at all. The model that has not undergone any pretraining is observed to have the lowest convergence rate, while the model pretrained with the curriculum learning strategy shows the swiftest convergence in the training phase, which demonstrates the effectiveness of our proposed curriculum-based pretraining method.



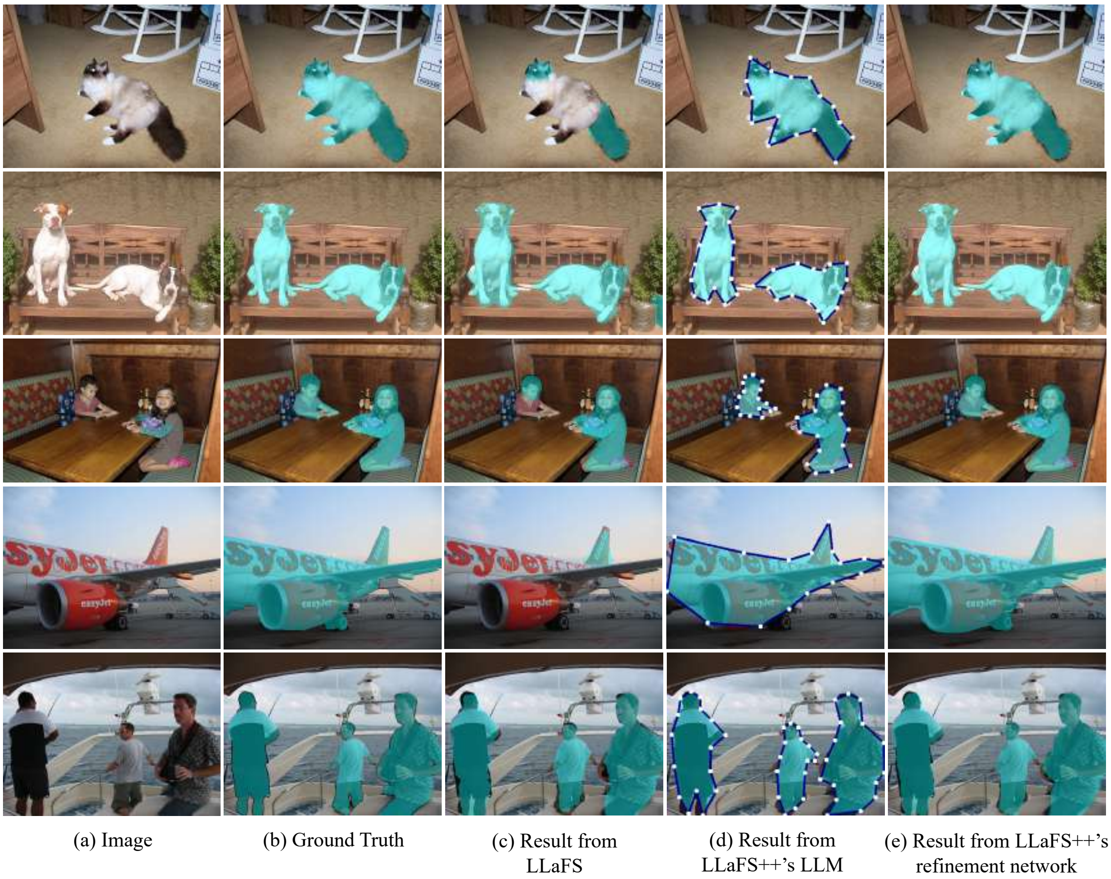
(a) Pretraining Loss Curve

(b) Training Loss Curve

FIGURE 4.13: Pretraining (a) and training (b) loss curves in different settings. Curriculum pretraining results in the best convergence in both pretraining and training stages. (Best viewed in color)

4.3.6 Visualizations of Segmentation Results

To provide an intuitive demonstration of our method's high performance and to illustrate the progress we have made in this extended work, we present visualizations of segmentation results generated by LLaFS++ and compare them with those from the conference version of our method, LLaFS [188]. These visualization results are presented in Figure 4.14, with each row from left to right showcasing the query image, the query ground truth, the segmentation result from LLaFS, the segmentation result from LLaFS++'s LLM, and the segmentation result from LLaFS++'s refinement network, respectively. A frequent error observed in the original LLaFS is its tendency toward oversegmentation hallucination, which refers to the model's mistake to segment only partial regions rather than the entirety of the target object in the query image (illustrated in detail in Sec.4.2.6). LLaFS++, the extended version in this chapter with several improvements and new designs, shows stronger segmentation capabilities with the more accurate segmentation outputs compared to LLaFS. Furthermore, the issue of oversegmentation hallucination is also significantly mitigated benefiting from our newly proposed inference method. It is noteworthy that the polygons produced by the LLaFS++'s LLM already exhibit strong segmentation results, while the results output from the refinement network are further refined and more precise, particularly at the object edges. Another important observation is that in cases where an image consists of


(a) Image

(b) Ground Truth

(c) Result from LLaFS

(d) Result from LLaFS++'s LLM

(e) Result from LLaFS++'s refinement network

FIGURE 4.14: Visualization of segmentation results for LLaFS and LLaFS++.

multiple objects, our method still demonstrates robust performance by accurately predicting multiple polygons to enclose different objects. These results demonstrate the excellent performance of our LLaFS++, showcasing its high effectiveness in handling the task of few-shot segmentation.

4.3.7 More Visualizations of Region-attribute Similarity Maps

In Figure 4.15, we showcase additional examples of the similarity map M_i generated based on the support image and each class attribute (see Sec.4.2.3 for details). It is promising to note that this similarity map demonstrates a notable attribute sensitivity, where regions associated with a given attribute often display higher similarities compared to other areas. This demonstrates the map's high effectiveness in aligning image regions with class attributes—a fundamental step in the formulation of the region-attribute correspondence table within our LLaFS++ framework.

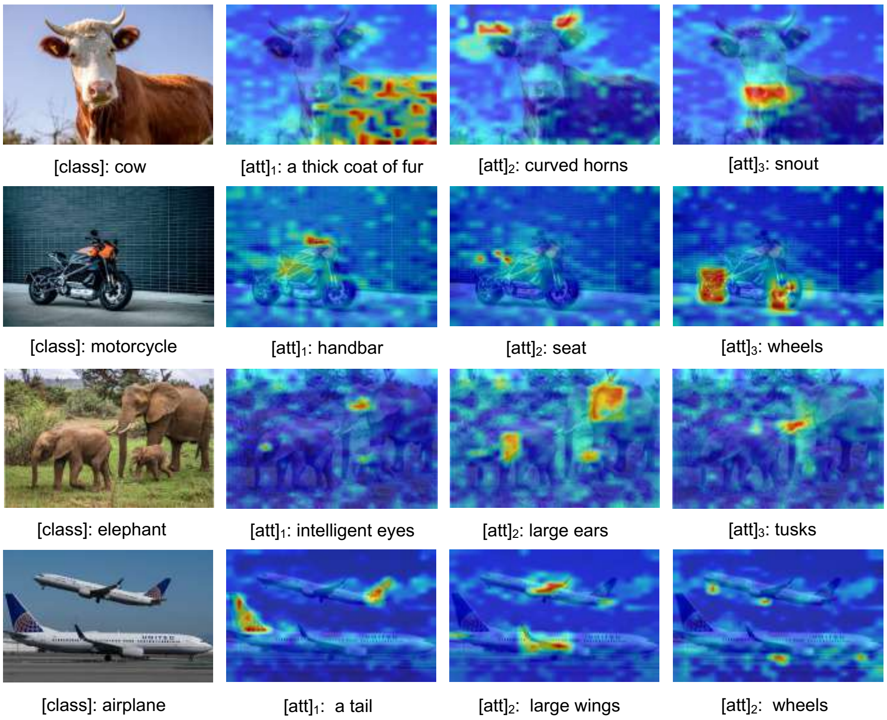


FIGURE 4.15: More examples of similarity maps M_i computed from the support image and class attributes.

4.3.8 Extended Experiments

In this section, we conduct a series of extended experiments to explore the applicability of our approach to several other tasks, including generalized few-shot segmentation, cross-domain few-shot segmentation, weak-label few-shot segmentation, and few-shot object detection. To ensure a fair comparison with previous methods, we use the ResNet50 backbone for generalized few-shot segmentation, cross-domain few-shot segmentation and weak-label few-shot segmentation; and for few-shot object detection, we use the ResNet101 backbone.

Generalized Few-Shot Segmentation

Compared to the setup adopted in this work, generalized few-shot segmentation is a more challenging task with greater real-world application value. It requires the trained model to not only segment new classes with annotated samples but also segment all the base classes that have been seen during the training phase. To adapt our LLaFS++ for this task, we introduce some minor modifications. Specifically, in its original form, LLaFS++ utilizes a set of learnable polygon embeddings as LLM's inputs to produce segmentation results. To address the generalized few-shot segmentation task, we split these polygon embeddings into two groups, denoted as $\{\hat{\mathbf{P}}_n\}$ and $\{\widetilde{\mathbf{P}}_n\}$. $\{\hat{\mathbf{P}}_n\}$ is tasked with producing segmentation results for the particular class corresponding to the support image, whereas $\{\widetilde{\mathbf{P}}_n\}$ is responsible for segmenting all classes that can be seen during the training phase. At the testing stage, we accomplish generalized few-shot segmentation by using $\{\hat{\mathbf{P}}_n\}$ for segmenting the novel class and $\{\widetilde{\mathbf{P}}_n\}$ for segmenting all seen base classes. This modified model is retrained on the PASCAL-5^i dataset and the results are presented in Table 4.13. We compare our method with other advanced cross-domain few-shot segmentation methods, including CAPL [130], DIaM [50] and VP [53]. Following previous works, we utilize three metrics—$\mathrm{mIoU_b}$, $\mathrm{mIoU_n}$, and $\mathrm{mIoU_m}$—to quantify the mIoU scores for base classes, novel classes, and the mean of $\mathrm{mIoU_b}$ and $\mathrm{mIoU_n}$, respectively. The comparative results show that LLaFS++ achieves the best performance on all three metrics. Another important finding is that in comparison with other approaches, the issue of bias towards the base classes is less pronounced in our method, as indicated by the narrower margins between metrics $\mathrm{mIoU_n}$ and $\mathrm{mIoU_b}$. This mitigation of bias may be due to our employment of a large number of class-agnostic synthetic pseudo samples for pretraining, which allows the model to learn a more general segmentation capability rather than overfitting to the trained categories.

Method	Venue	1-shot			5-shot		
		$\mathrm{mIoU_b}$	$\mathrm{mIoU_n}$	$\mathrm{mIoU_m}$	$\mathrm{mIoU_b}$	$\mathrm{mIoU_n}$	$\mathrm{mIoU_m}$
CAPL	CVPR'22	65.5	18.9	42.2	66.1	22.4	44.3
DIaM	CVPR'23	70.9	35.1	53.0	70.9	55.3	63.1
VP	CVPR'24	76.4	39.8	58.1	76.4	56.1	66.3
PixelLM	CVPR'24	78.6	53.5	66.1	79.1	62.6	70.9
LLaFS++	-	**79.8**	**62.7**	**71.3**	**80.2**	**71.4**	**75.8**

TABLE 4.13: Experimental results on generalized few-shot segmentation

Cross-Domain Few-Shot Segmentation

The recently proposed task of cross-domain few-shot segmentation focuses on addressing the challenging domain shift problem within few-shot segmentation. This task not only needs to segment new, previously unseen classes as the normal few-shot segmentation, but also requires the model to be able to process testing images in the different domains from the training images. Following previous works, we conduct experiments in the COCO-to-PASCAL setting, where the model is trained using the COCO-20^i dataset and tested on the PASCAL-5^i dataset. Results presented in Table 4.14 indicate that LLaFS++ achieves the best performance. Note that we do not make any changes to the model structure or the training method of LLaFS++ in this experiment, yet it still outperforms all comparative methods, including some designed specifically for cross-domain few-shot segmentation [142, 104]. This could be attributed to the rich and general pretrained knowledge of LLM, which enables our framework to handle different domains effectively. These results indicate that LLaFS++ can achieve excellent performance even in the presence of domain shifts, thus demonstrating its high robustness and effectiveness.

Method	Venue	1-shot	5-shot
Meta-Memory	CVPR'22	65.6	70.1
BAM-final	T-PAMI'23	69.0	71.7
IFA	CVPR'24	71.0	80.9
LLaFS++	-	**79.6**	**84.3**

TABLE 4.14: Results on cross-domain few-shot segmentation.

Weak-Label Few-shot Segmentation

Annotating pixel-level segmentation ground truth is time-consuming and labor-intensive, whereas a weaker annotation, such as the bounding box, is much easier to obtain. To this end, we evaluate the effectiveness of utilizing bounding boxes as support images' labels in our framework. Specifically, we consider the space inside the bounding box as the foreground region, generating a corresponding binary mask which is then input into the LLaFS++ framework to serve as the support ground truth mask. The results in Table 4.15 indicate that the original LLaFS++ model, which is trained on the pixel-level ground truth mask, can still maintain good performance when evaluated using bounding-box-based support ground truth, with only a minor drop in mIoU by 3.5% compared to testing using the pixel-level support ground truth. Moreover, this performance gap can be further reduced to 1.3% when LLaFS++ is retrained using bounding boxes as the support ground truth. These results demonstrate the robustness of LLaFS++ against annotation noise, indicating that LLaFS++ can work well even when provided with only bounding-box-level annotations. Such adaptability is valuable for real-world applications, offering a practical solution to reduce the burden of detailed annotation.

Training	Testing	mIoU
PM	PM	77.8
PM	BB	75.3
BB	BB	76.5

TABLE 4.15: Experimental results on weak labels. 'PM': pixel-level mask; 'BB': bounding box.

Few-Shot Object Detection

To validate the generalizability of LLaFS++, we expand our experiments to include tasks beyond segmentation, specifically, few-shot object detection. Similar to few-shot segmentation, few-shot object detection aims to detect the objects within a new category using only a small number of annotated images for this class. To tailor LLaFS++ for this task, we modify the model's output from the vertex coordinates of 16-polygons to the sizes and center coordinates of bounding boxes. Descriptions in the instruction about the output format are also accordingly modified, while the rest of the model's structure remains unchanged. We carry out the experiments using the PASCAL VOC dataset and report the performance based on the AP50 metric. The results shown in Table 4.16 indicate that LLaFS++ achieves the best results compared to other methods

Method	Venue	1-shot			5-shot		
		Set-1	Set-2	Set-3	Set-1	Set-2	Set-3
KFSOD	CVPR'22	44.6	37.8	34.8	60.9	48.1	52.7
Pseudo	CVPR'22	54.5	32.8	48.4	63.2	49.8	59.6
FS-DETR	ICCV'23	45.0	37.3	43.8	52.7	46.6	52.1
σ-Adaptive	ICCV'23	52.3	42.7	47.8	65.9	54.8	60.3
LLaFS++	-	**60.8**	**46.2**	**56.1**	**69.8**	**59.5**	**64.5**

TABLE 4.16: Experimental results on few-shot detection. Following previous methods [176, 158]. the entire PASCAL VOC dataset is divided into three different splits, namely 'Set-1', 'Set-2' and 'Set-3', each consisting of 15 base and 5 novel classes. Results for all 3 splits are reported.

that are specifically designed for few-shot object detection (including KFSOD [176], Pseudo [66], FS-DETR [12], and σ-Adaptive [38]), demonstrating that LLaFS++ is a highly generalizable model with the ability to process different tasks. These experiments provide us with insights that LLMs have great potential to be applied in more few-shot computer vision tasks beyond segmentation, which will be a direction for our future research.

4.3.9 More Discussions About Region-attribute Corresponding Table

As detailed in the method sections, we introduce a region-attribute corresponding table as a fine-grained multi-modal guidance. This table outlines specific attributes of the target class and their aligned regions on the support image, thereby instructing the LLM to carry out segmentation in a more fine-grained and human-like manner. To achieve this alignment, we utilize the enhanced CLIP proposed by [101] to compute the similarity between attribute text features and image features. We find that the similarity map obtained in this manner can effectively reflect the corresponding regions of the attributes. Please refer to the main paper for more details.

The effectiveness of our region-attribute alignment technique is greatly attributed to the advanced pixel-level image-text matching capabilities of the enhanced CLIP. This naturally leads to the question of whether one could directly employ the similarity map S_q, calculated from text features of the class name and the visual features of the query image via the enhanced CLIP, as the final segmentation outcome. Experimental results indicate that although this method yields some positive outcomes, the resultant 68.1% mIoU on PASCAL-5^i fold-0 and 33.5% mIoU on COCO-20^i fold-0 are substantially lower than those attained by our superior method LLaFS++ (77.8% on PASCAL-5^i fold-0 and 50.8% on COCO-20^i fold-0). We further experiment with incorporating S_q as an additional input to the LLM within our LLaFS++ framework. However, this adjustment does not yield significant improvement over the original LLaFS++ (77.9% mIoU for this method and 77.8% for LLaFS++ on PASCAL-5^i fold-0). This lack of enhancement may be attributed to the fact that LLaFS++ already exhibits a powerful ability to perform high-quality segmentation. Thus, further providing a coarse and imprecise preliminary segmentation result of the query image cannot contribute more valuable new information that could further refine the segmentation accuracy.

Another notable point is that our region-attribute corresponding table is generated using the support image (denoted as the support-based table). We have also experimented with creating this table from the query image instead (denoted as the query-based table) and investigated its impact when introduced into the LLaFS++ framework. Our results indicate that replacing the support-based table with the query-based one (73.6% mIoU on PASCAL-5^i), or adding the query-based table as an additional source of information while maintaining the support-based table (75.8% mIoU), cannot improve and may even reduce segmentation accuracy compared to the original LLaFS++ (77.8% mIoU). We attribute this reduced performance to two primary factors: Firstly, the query-based table is inherently noisy due to the unavailability of ground truth for the query image, which hinders the use of masked attention in the Region Encoding Network (REN). Without the ability to isolate foreground regions through masked attention, the REN inadvertently captures a substantial amount of noise from the query image's background, which could potentially mislead the LLM. Secondly, the support-based table, in conjunction with the class name, class attributes, and ground truth of the support image, establishes a comprehensive and logical guidance system that simulates the human cognitive mechanism of 'from general to detailed, from abstract to concrete'. Every component is indispensable for the construction of such a structured guidance. Replacing the support-based table with the query-based one disrupts this guidance due to the absence of a crucial part within this structured information, i.e., query ground truth, thus compromising the LLM's ability to execute segmentation in a human-like and fine-grained manner. Both of the aforementioned two limitations can be attributed to the lack of ground truth data for the query images. If we use the preliminary segmentation result mentioned in the previous paragraph as a substitute for query ground truth to implement masked attention in the REN and to complete the guidance information, the obtained performance of 76.5% mIoU is still lower than the original LLaFS++ (77.8% mIoU). This suboptimal result may be attributed to the coarse and imprecise nature of the preliminary query segmentation, which likely introduces considerable noise and thus impairs the LLaFS++'s ability to segment query images.

In summary, using a preliminary query segmentation result or a query-image-derived region-attribute corresponding table as LLM's additional inputs cannot enhance LLaFS++'s effectiveness. Hence, the framework detailed in thi chapter stands as the optimal design for our proposed method.

4.4 Conclusion

This chapter introduces LLaFS++, a pioneering framework that, for the first time, utilizes large language models (LLMs) for tackling few-shot segmentation and achieves significant success. To adapt LLMs for this visual task, we introduce a segmentation task instruction to provide detailed task definitions, and propose a fine-grained in-context instruction to simulate human cognitive mechanisms and provide fine-grained multimodal reference information. We also propose a pseudo-sample-based curriculum pretraining mechanism to augment the training samples required for instruction tuning, and introduce a novel inference method to mitigate potential oversegmentation hallucinations caused by the regional guidance information. Our thorough experiments validate the effectiveness of LLaFS++, showcasing its capability to consistently surpass existing state-of-the-art methods across various datasets and scenarios. We consider LLaFS++ as a significant step forward in exploring how LLM frameworks can be effectively leveraged for addressing few-shot challenges within the domain of computer vision.

Chapter 5

Continual Semantic Segmentation with Automatic Memory Sample Selection

In this chapter, we tackle the problem of Continual Semantic Segmentation (CSS), which extends static semantic segmentation by incrementally introducing new classes for training. To alleviate the catastrophic forgetting issue in CSS, a memory buffer that stores a small number of samples from the previous classes is constructed for replay. However, existing methods select the memory samples either randomly or based on a single-factor-driven hand-crafted strategy, which has no guarantee to be optimal. In this work, we propose a novel memory sample selection mechanism that selects informative samples for effective replay in a fully automatic way by considering comprehensive factors including sample diversity and class performance. Our mechanism regards the selection operation as a decision-making process and learns an optimal selection policy that directly maximizes the validation performance on a reward set. To facilitate the selection decision, we design a novel state representation and a dual-stage action space. Our extensive experiments on Pascal-VOC 2012 and ADE 20K datasets demonstrate the effectiveness of our approach with state-of-the-art (SOTA) performance achieved, outperforming the second-place one by 12.54% for the 6-stage setting on Pascal-VOC 2012.

5.1 Introduction

Semantic segmentation is an important task with a lot of applications. The rapid development of algorithms [19, 79, 187, 48, 43] and the growing number of publicly available large datasets [35, 180] have led to great success in the field. However, in many scenarios, the static model cannot always meet real-world demands, as the constantly changing environment calls for the model to be constantly updated to deal with new data, sometimes with new classes.

A naive solution is to apply continual learning by incrementally adding new classes to train the model. However, it is not simple as it looks – almost every time, since the previous classes are inaccessible in the new stage, the model forgets the information of them after training for the new classes. This phenomenon, namely catastrophic forgetting, has been a long-standing issue in the field. Furthermore, the issue is especially severe in dense prediction tasks like semantic segmentation.

Facing the issue, existing works [132, 115, 9, 58, 1, 65, 7, 37, 16] propose to perform exemplar replay by introducing a memory buffer to store some samples from previous classes. By doing so, the model can be trained with samples from both current and previous classes, resulting in better generalization. However, since the number of selected samples in the memory is much smaller than those within the new classes, the selected samples are easy to be ignored or cause overfitting when training due to the small number. Careful selection of the samples is required, which naturally brings the question: *How to select the best samples for replay?*

Some attempts have been made to answer the question, aiming to seek the most effective samples for replay. Researchers propose different criteria that are mostly manually designed based on some heuristic factors like diversity [132, 115, 9, 58, 1, 65, 7]. For example, [97] selects the most common samples with the lowest diversity for replay, believing that the most representative samples will elevate the effectiveness of replay. However, the most common samples may not always be the samples being forgotten in later stages. [7] proposes to save both the low-diversity samples near the distribution center and high-diversity samples near the classification boundaries. However, new challenges arise since the memory length is limited, so it is challenging to find the optimal quotas for the two kinds of samples to promote replay effectiveness to the greatest extent. Moreover, most of the existing methods are designed based on a single factor, the selection performance, however, can be influenced by many factors with complicated relationships. For example, besides diversity, memory sample selection should also be *class-dependent* because the hard classes need more samples to replay in order to alleviate the more severe catastrophic forgetting issue. Therefore, we argue that it is necessary to select memory samples in a more intelligent way by considering the more comprehensive factors and their complicated relationships.

Witnessing the challenge, in this chapter, we propose a novel automatic sample selection mechanism for CSS. Our key insight is that selecting memory samples can be regarded as a decision-making task in different training stages, so we formulate the sample selection process as a Markov Decision Process, and we propose to solve it automatically with a reinforcement learning (RL) framework. Specifically, we employ an agent network to make the selection decision, which receives the state representation as the input and selects optimal samples for replay. To help the agent make wiser decisions, we construct a novel and comprehensive state combined with the sample diversity and class performance features. In the process of state computation, the inter-sample similarity needs to be measured. We found the naive similarity measurement by computing the prototype distance is ineffective in segmentation, as the prototype losses the local structure details that are important for making pixel-level predictions. Therefore, we propose a novel similarity measured in a multi-structure graph space to get a more informative state. We further propose a dual-stage action space, in which the agent not only selects the most appropriate samples to update the memory, but also enhances the selected samples to have better replay effectiveness in a gradient manner. All the careful designs allow the RL mechanism to be effective in solving the sample selection problem for CSS.

We perform extensive experiments on Pascal-VOC 2012 and ADE 20K datasets, which demonstrate the effectiveness of our proposed novel paradigm for CSS. Benefiting from the reward-driven optimization, the automatically learned policy can help

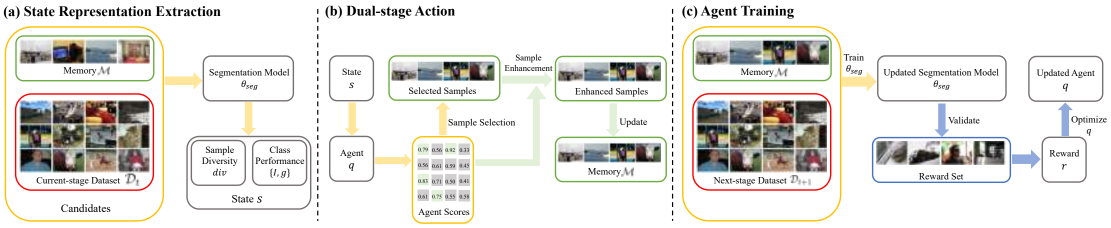


FIGURE 5.1: The overall scheme of our automatic memory sample selection mechanism for CSS. (a) Given the memory $\mathcal{M}$ and current-stage dataset $\mathcal{D}_t$, we first extract the state representation for each sample in $\mathcal{M} \cup \mathcal{D}_t$, which is consisted of the sample diversity and class performance features. (b) Given the state representations, the agent q produces a score for each candidate sample. Based on the scores, we select several samples and enhance them in a gradient-based manner. The memory $\mathcal{M}$ is updated by these samples. (c) The segmentation model θ_{seg} is trained using the updated $\mathcal{M}$ and $\mathcal{D}_{t+1}$. We then validate the updated θ_{seg} on a reward set, resulting in the reward t that is used to optimize agent q.

select the more effective samples, thus resulting in better performance than the previous strategies. On both datasets, our method achieves state-of-the-art (SOTA) performance. To summarize, our contributions are as follows:

- We formulate the sample selection of CSS as a Markov Decision Process, and introduce a novel and effective automatic paradigm for sample replay in CSS enabled by reinforcement learning.
- We design an effective RL paradigm tailored for CSS, with novel state representations containing multiple factors that can guide the selection decision, and a dual-stage action space to select samples and boost their replay effectiveness.
- Extensive experiments demonstrate our automatic paradigm for sample replay can effectively alleviate the catastrophic forgetting issue with state-of-the-art (SOTA) performance achieved.

5.2 Preliminaries

Continual semantic segmentation (CSS) aims to train a segmentation model in T stages continuously without forgetting. In each stage t, a training dataset $\mathcal{D}_t$ can be utilized, where only pixels within the current classes $\mathcal{C}_t$ are labeled, leaving pixels within others classes (including previous classes $\mathcal{C}_{1:t-1}$ and future classes $\mathcal{C}_{t+1:T}$) as the background class. The goal is to allow the model to be able to predict all classes $\mathcal{C}_{1:T}$ after completing all T stages. To alleviate the catastrophic forgetting problem in CSS, an exemplar memory $\mathcal{M}$ that contains a small number of sampled data from the previous classes can be used for replay, so that both $\mathcal{M}$ and $\mathcal{D}_t$ are involved for training.

In the training process, $\mathcal{M}$ is updated once a training stage is completed. This means $\mathcal{M}$ will be refilled by new samples from $\mathcal{M} \cup \mathcal{D}_t$ after the stage t with the learning on $\mathcal{D}_t$ completed. It is obvious that the careful selection of samples for $\mathcal{M}$ could greatly affect the performance, which is also the focus of this chapter.

5.3 Method

5.3.1 Overall

Considering the memory $\mathcal{M}$ with L samples and $\mathcal{D}_t$ with N_t samples, the target of this work is to learn an optimal policy that automatically selects L samples from $\mathcal{M} \cup \mathcal{D}_t$ and put them into $\mathcal{M}$ for the next stage training, driven by maximizing the designed reward reflecting the performance improvement.

The selection decision is made by an agent network that is a three-layered MLP. It converts the CSS to become a decision-making process with the following procedure: 1) Obtaining the state s by assessing the properties of samples that can measure its contribution for replay. 2) Based on s, using the agent q to make an action a that selects L samples to update the memory $\mathcal{M}$. 3) Training the segmentation network with the updated $\mathcal{M}$. 4) Computing the reward r based on the validation performance of the updated segmentation network. 5) Repeating the above steps until completing all T stages. 6) Optimizing agent q based on r from all stages.

As shown in Figure 5.1, in this chapter, we solve the above problem under a reinforcement learning (RL) framework, in which the agent q scores each state s and makes an action a based on the score. Benefiting from the task-specific state representations, a novel selection-enhancement dual-stage action space and the reward-driven optimization, we can optimize the agent to learn an effective selection policy. In the following parts of this section, we illustrate the details of how these components are designed.

5.3.2 State Representation

The state representation s is the key to making the automatic selection decision process possible, as it is the input to and serves as the decision support of the agent network. Designing the state should consider the requirements of the selection policy. Intuitively, an optimal policy should make a selection decision by estimating the potential replay contribution of each sample, and allocate different quotas to different classes as the hard classes suffer from the more severe catastrophic forgetting issue and need more samples to replay. Based on these intuitions, we propose to combine two kinds of cues including ***sample diversity*** and ***class performance*** for constructing state. For an image within class c, sample diversity div measures its novelty, which can reflect the potential replay effectiveness as indicated by previous works [7, 115]. A higher div indicates the sample differs more from other images within the same class c. We calculate it by computing and averaging the inter-sample similarities. The class performance is constructed as the combination of two metrics: 1) accuracy and 2) forgetfulness. We derive accuracy by computing the training IoU I_c for each class c. The hard classes that are trained to the worse performance have the lower IoUs. However, as the IoU measures the current training accuracy, it cannot reflect whether a class is easily forgotten in the future, which is critical for CSS but difficult to measure directly since the future performance is unknown. We thus estimate forgetfulness g_c by measuring the similarities between c with all other classes, motivated by the previous finding that classes that are more similar to other classes are more likely to be forgotten [110]. Eventually, given an image, on all C classes in it, we compute their diversities $\{div_c\}_{c=1}^{C}$, accuracy $\{I_c\}_{c=1}^{C}$ and forgetfulness $\{g_c\}_{c=1}^{C}$, resulting in three groups of features. Then, we calculate the

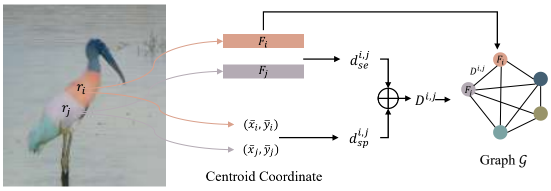


FIGURE 5.2: Illustration of how the graph for computing sample diversity is constructed. In the figure, r_i and r_j denote two superpixels. F_i and F_j refer to the average features for all pixels within them. $(\overline{x}_i, \overline{y}_i)$ and $(\overline{x}_j, \overline{y}_j)$ denote the centroid coordinates of r_i and r_j respectively. $d_{se}^{i,j}$ and $d_{sp}^{i,j}$ refer to the semantic distance and spatial distance. The generated graph $\mathcal{G}$ will be used to compute the sample diversity.

average values of the three groups over different classes, and concatenate them to get the state representation s of the image.

Measuring Similarity in Multi-structure Space

Motivation. Both the sample diversity div and forgetfulness g_c introduced above need to compute the similarity. In previous works, the similarity is mainly measured in the *prototype-level space* [115] or *pixel-level space* [137]. The former condenses the sample into a single prototype feature and then calculates the feature distance. It is computationally efficient, but drops the spatial information and structural details, which leads to errors. For example, two images with completely different local structures or object postures may have similar prototype features, since the prototypes are computed by the average features of all pixels, concealing the differences between local details. Such errors caused by the lack of local details are detrimental to the segmentation task, where local structural information is important for making pixel-level predictions [187]. As a result, the state constructed by the prototype-level similarity leads to poor performance when employed to CSS. The pixel-level one retains the local information, however, it requires an unacceptable computation cost due to the pixel-wise distance calculation and may cause overfitting [74]. Thus, to obtain a more informative similarity, a novel representation space is needed, which should not only retain the spatial and structural information but also be condensed for a reasonable computation cost. Based on the discussion, we propose a novel method that first maps each sample into a *multi-structure graph space* and then measures the inter-sample similarity based on the graph matching. Each vertex of the graph represents a semantic structure, and the edge represents the spatial and semantic correlations, thus a fine-grained similarity can be measured by utilizing the comprehensive information.

Multi-structure Graph. Considering an image with the class c, we represent the region $\mathcal{R}$ within c as a graph $\mathcal{G}$ through the way illustrated by Fig. 5.2. To get the local structural representation, we first use the method as in [74] to generate M superpixels $\{r_m\}_{m=1}^{M}$ $(r_1 \cup r_2 \cup ... \cup r_M = \mathcal{R})$. The motivation for using superpixels is that, according to the construction mechanism of superpixels, each r_m can represent a meaningful semantic structure such as the head of a bird, and condenses the pixel-level representation enabled by clustering pixels with similar features and adjacent positions. Each vertex F_m is then computed as the average feature for all pixels within r_m. We represent the edge of $\mathcal{G}$ as a distance map $D \in \mathbb{R}^{M \times M}$, where the element $D^{i,j}$ denotes the distance between the i-th and j-th vertices. To simultaneously consider the context-aware high-level semantic information and low-level spatial correlation, we combine both the *semantic distance* and *spatial distance* for getting D. Concretely, the semantic distance $d_{se}^{i,j}$ is the L2 distance between F_i and F_j; the spatial distance $d_{sp}^{i,j}$ denotes the Euclidean distance between the two centroid coordinates [1] of the superpixels r_i and r_j, reflecting their relative positions. We normalize $d_{se}^{i,j}$ and $d_{sp}^{i,j}$ to $[0, 1]$ and derive $D^{i,j} = d_{se}^{i,j} + d_{sp}^{i,j}$. Such a graph can capture comprehensive representations such as local structure details and spatial information, which are lost in the prototype space but are crucial for measuring a fine-grained similarity.

Inter-graph Similarity. After mapping images into the graph space, we use the matching algorithm to measure the similarities. For two graphs $\mathcal{G}_i$ and $\mathcal{G}_j$, the Sinkhorn algorithm [36] is applied for aligning them, in which the transport cost tc is obtained by solving the optimal transport problem. Specifically, considering a graph $\mathcal{G} = \{F_m, \{D^{m,n}\}_{n=1}^{M}\}_{m=1}^{M}$, we first generate a single-vector representation for each vertex by aggregating other vertices. The aggregation is achieved through weighted sum written as:

$$\hat{F}_m = \frac{1}{\sum_{n=1}^{M} W_{m,n}} \sum_{n=1}^{M} W_{m,n} F_n, \tag{5.1}$$

Where the weight $W_{m,n}$ is formulated by:

$$W_{m,n} = \exp\left(-D^{m,n}\right). \tag{5.2}$$

In this way, $\mathcal{G}$ is represented as $\{\hat{F}_m\}_{m=1}^{M}$. For simplify, we denote $\mathcal{G}_i$ for the i-th image as $\{\hat{F}_m^i\}_{m=1}^{M}$. Next, we match $\mathcal{G}_i$ and $\mathcal{G}_j$ by solving an optimal transport (OT) task:

$$\underset{A}{\text{Min}} \sum_{a,b} A_{a,b} M_{a,b}, \tag{5.3}$$

where A is the transportation plan that implies the alignment information and M is the cost matrix. $M_{a,b}$ measures the transport cost from the a-th vertex $\hat{F}_a^i$ in $\mathcal{G}_i$ to the b-th vertex $\hat{F}_b^j$ in $\mathcal{G}_i$, which is written as:

$$M_{a,b} = 1 - \text{Cos}\left(\hat{F}_a^i, \hat{F}_b^j\right), \tag{5.4}$$

where Cos denotes the cosine similarity. The unique solution A^* can be calculated through Sinkhorn's algorithm:

$$A^* = \text{diag}\left(\mathbf{u}\right) K \text{diag}\left(\mathbf{v}\right), \tag{5.5}$$

[1] Considering a superpixel $r = \{(x_i, y_i)\}_{i=1}^{N}$, the centroid coordinate $(\overline{x}, \overline{y})$ is computed as: $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, $\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$.

where the vectors $\mathbf{u}$ and $\mathbf{v}$ are obtained through the above iterations:

$$\begin{aligned}\mathbf{v}^{t=0} &= \frac{\mathbf{1}_m}{\mathbf{v}^{t+1}}, \\ \mathbf{u}^{t+1},\ \mathbf{v}^{t+1} &= \frac{\mathbf{1}_n}{K\mathbf{u}^{t+1}}\end{aligned} \tag{5.6}$$

we set the iteration number to be 5. Finally, the transport cost tc is computed as:

$$tc = \sum_{a,b} A^*_{a,b} M_{a,b}, \tag{5.7}$$

which measures the similarity between $\mathcal{G}_i$ and $\mathcal{G}_j$. A higher tc represents the lower similarity of the two graphs. The details for this step are presented in supplementary materials. As the edge distance $D^{i,j}$ is computed with both the semantic and spatial distance, the computed tc after matching can reflect both the semantic and spatial similarity. For example, considering two regions for the 'person' class, we can measure both whether they wear similar clothes (semantic similarity) and whether they are with the same body posture (spatial similarity), capturing the comprehensive fine-grained representations.

Representation Computation. We use the above-mentioned similarity measurement to compute the sample diversity div and forgetfulness g in state representations. For an image with the c-th class, let $\mathcal{G}$ be its graph. We introduce a support set $\mathcal{S}_c = \{\mathcal{G}^i_c\}_{i=1}^{N_c}$ to contain several graphs for other images within the same class c. For each previous class in $\mathcal{C}_{1:t-1}$, we construct $\mathcal{S}_c$ as the set of all images saved in the memory. For each current class in $\mathcal{C}_t$ that has a larger number of samples, to relieve the computation burden, we randomly sample 10% from all images to form $\mathcal{S}_c$. We will show in supplementary material that div computed from a sampled set can be effective enough. A diverse and novel sample is likely to have low similarities compared to other samples within the same class. We thus get div by computing the average similarities by:

$$div = \frac{1}{|\mathcal{S}_c|} \sum_{\mathcal{G}^i_c \in \mathcal{S}_c} \mathrm{Sim}\left(\mathcal{G}, \mathcal{G}^i_c\right), \tag{5.8}$$

where Sim refers to the inter-graph similarity measurement introduced above. To get the forgetfulness g_c for each class c, we first construct a representative set $\hat{\mathcal{S}}_c = \{\mathcal{G}^i_c\}_{i=1}^{\hat{N}_c}$ containing the top 10% samples in $\mathcal{S}_c$ with the lowest diversity scores. These samples are most similar to other samples in c so they can represent the class-level properties. Then forgetfulness g_c is gotten as the class-wise similarity computed by:

$$g_c = \frac{1}{|\hat{\mathcal{S}}_c|} \sum_{\mathcal{G}^i_c \in \hat{\mathcal{S}}_c} \frac{1}{|\mathcal{C}_{1:t}| - 1} \sum_{j \in \mathcal{C}_{1:t} \backslash c} \frac{1}{|\hat{\mathcal{S}}_j|} \sum_{\mathcal{G}^k_j \in \hat{\mathcal{S}}_j} \mathrm{Sim}\left(\mathcal{G}^i_c, \mathcal{G}^k_j\right). \tag{5.9}$$

Eventually, the obtained div and g are combined with the accuracy I, generating the state representations that can help make a wiser selection decision.

5.3.3 Dual-stage Action with Sample Selection and Enhancement

After getting the state information s^i for each sample, we use an agent network q to produce a score $q(s^i)$ by taking s^i as the input. A higher score indicates the sample is more suitable for replay. Thus, we regard agent score as the replay effectiveness indicator, and utilize it to drive a novel action space for the RL mechanism that has two stages: *sample selection* and *sample enhancement*.

Concretely, we first select memory samples by L ones with the highest agent scores, which is written as:

$$a = \underset{i \in [1, L+N_t]}{\mathrm{TopL}} \; q\left(s^i\right). \tag{5.10}$$

After that, instead of directly using the static selected samples for training in the next stage, we further propose an enhancement operation that edits each sample to be more effective for replay. This is motivated by our observation of the agent scores for the selected samples. We notice that, only 10% of the selected samples have agent scores exceeding 0.8 (the theoretical maximum score is 1). The phenomenon shows that such samples are the best possible choice from the imperfect candidates, but not the ideally perfect samples for replay. Thus, despite achieving better performance by selecting the most adequate samples, there is still room to further improve the replay effectiveness if we can enhance the samples to reach higher scores. We thus implement enhancement through a gradient-based manner by maximizing the agent score. Concretely, we regard the state s^x as a feature computed from input image x along with $\mathcal{M}$ and $\mathcal{D}_t$ under the segmentation network parameters θ_{seg} with the state computing function f_s, which is formulated as:

$$s^x = f_s\left(x; \mathcal{M}, \mathcal{D}_t, \theta_{seg}\right). \tag{5.11}$$

Then the agent score is generated by $q(s^x)$. We perform a gradient update on x so that the agent score $q(s^x)$ moves towards the larger direction reflecting the better replay effectiveness, which is written as:

$$\begin{aligned} x' &= x + \epsilon \nabla_x q\left(s^x\right) \\ &= x + \epsilon \nabla_x q\left(f_s\left(x; \mathcal{M}, \mathcal{D}_t, \theta_{seg}\right)\right), \end{aligned} \tag{5.12}$$

where ϵ is a hyper-parameter to control an adequate updating rate so that the image label remains unchanged. With the higher agent score, the resulted x' can be more effective and is stored into $\mathcal{M}$ for replay.

5.3.4 Reward and Optimization

Our selection policy aims to allow the segmentation model trained with the memory $\mathcal{M}$ to achieve better performance. Therefore, the reward for optimizing agent should reflect how much the memory samples derived by the agent policy can benefit the CSS training. To implement the goal, we divide a subset from the training set to get a reward set $\mathcal{D}^{reward}$, and define reward r_t at the t-th stage as the validation accuracy on $\mathcal{D}^{reward}$ evaluated on the segmentation model that has completed the t-th stage. With reward derived, following DQN algorithm [136], the agent is optimized by the

Algorithm 1 Agent Training Algorithm.

```
**Input:** agent network $q$, segmentation network parameters $\theta_{seg}$, dataset $\mathcal{D}_1$.
**for** $y$ **in** $1, ..., Y$ **do**
    Create a new task having $T_y$ continual stages with class partitions $\{\mathcal{C}_{t_y}\}_{t_y=1}^{T_y}$.
    Partition $\mathcal{D}_1$ to $\mathcal{D}_1^{train}$ and $\mathcal{D}_1^{reward}$
    Initialize $\theta_{seg}$, initialize $\mathcal{M}$ as an empty set
    **for** $t_y$ **in** $1, ..., T_y$ **do**
        Train $\theta_{seg}$ on $\mathcal{M} \cup \mathcal{D}_1^{train,t_y}$
        Compute state $s_t$ (Sec.5.3.2) and agent scores $q(s_t)$
        Select and enhance samples (Sec.5.3.3), update $\mathcal{M}$
        **if** $t_y > 1$ **then**
            Compute reward $r_{t_y}$ (Sec.5.3.4)
        **end if**
    **end for**
    Update $q$ by Eq. 5.13
**end for**
**Return:** $q$
```

temporal difference (TD) error formulated as:

$$
\begin{aligned}
TD\left(\theta, \hat{\theta}\right) = \frac{1}{T-1} \sum_{t=1}^{T-1} \Bigg(& r_{t+1} + \frac{\gamma}{L} \sum_{i=1}^{L} q\left(s_{t+1}^{a_{t+1}^{i}}; \hat{\theta}\right) \\
& - \frac{1}{L} \sum_{i=1}^{L} q\left(s_t^{a_t^i}; \theta\right) \Bigg)^2 ,
\end{aligned}
\tag{5.13}
$$

where $s_t^{a_t^i}$ refers to the state representation of the i-th selected sample in the t-th stage, θ and $\hat{\theta}$ refer to the agent's policy and off-policy parameters respectively. Following [136], $\hat{\theta}$ is periodically updated based on θ, aiming to save the learned Q-value.

5.3.5 Agent Training and Deployment

With the above-introduced RL mechanism for CSS, we then present the agent training and deployment method in this section. We denote $\mathcal{D}_1$ as the dataset for first-stage training. According to CSS protocol [37], $\mathcal{D}_1$ contains multiple classes (usually more than half of the total). Thus, it can provide sufficient information for training an effective agent. The detailed training process is shown in Alg.1. We train the agent for Y iterations. In each iteration, we randomly divide $\mathcal{D}_1$ into the training set $\mathcal{D}_1^{train}$ and the reward set $\mathcal{D}_1^{reward}$, and set a new CSS task by reallocating the classes observed in each stage. This helps the agent to learn a more general policy with training from diverse settings.

Once the agent training is completed, we can deploy it on the whole set $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{T}$, selecting and enhancing memory samples at the end of each stage and using them for replay in the next stage.

	19-1(2 stages)			**15-5(2 stages)**			**15-1(6 stages)**		
Method	0-19	20	all	0-15	16-20	all	0-15	16-20	all
Joint	79.45	72.94	79.14	79.77	72.35	77.43	78.88	72.63	77.39
EWC	26.90	14.00	26.30	24.30	35.50	27.10	0.30	4.30	1.30
LwF-MC	64.40	13.30	61.90	58.10	35.00	52.30	6.40	8.40	6.90
ILT	67.75	10.88	65.05	67.08	39.23	60.45	8.75	7.99	8.56
MiB	70.57	22.82	68.30	75.30	48.68	68.96	39.47	14.50	33.53
RCN	-	-	-	78.80	52.00	72.40	70.60	23.70	59.40
REMINDER	76.48	32.34	74.38	76.11	50.74	70.07	68.30	27.23	58.52
SDR	68.52	23.29	66.37	75.21	46.72	68.64	43.08	19.31	37.42
PLOP	75.35	37.35	73.54	75.73	51.71	70.09	65.12	21.11	54.64
Ours	79.40	42.80	77.66	79.31	55.88	73.73	78.54	50.82	71.94

TABLE 5.1: Comparison results on Pascal-VOC 2012.

5.4 Experiments

5.4.1 Implementation Details

Our method contains two phases: agent training and policy deployment. The first phase trains the agent network to get the selection policy, while the latter phase employs the trained policy for the CSS training.

For the deployment phase, the hyper-parameters settings follow the previous work [37]. Concretely, we adopt SGD as the optimizer, where the momentum value is 0.9 and the initial learning rate is 1e-2 with the 'poly' learning rate decay schedule. For each continual stage, the network is trained for 30 epochs on Pascal VOC and 60 epochs for ADE20K. The batch size is 24 for both datasets. Following [16], the memory length $|\mathcal{M}|$ is 100 and 300 for Pascal-VOC 2012 and ADE20K, respectively. Following [74], the superpixel number M for computing sample diversity is 5, ϵ in Eq.5.12 is 0.1.

For the agent training phase, we use the different hyper-parameters settings to speed up training. Concretely, in this phase, we use Deeplabv3 with ResNet18 backbone as the segmentation model. The training epochs Y in Alg.1 is 1000. We randomly partition 10% of whole data into the training set and leave others as the reward set. For each continual stage, the network is trained for 5 epochs on Pascal VOC and 8 epochs for ADE20K. The segmentation network is optimized by SGD with the initial rate being 0.01, and the agent network is optimized by Monmentumn with the learning rate being 0.1.

5.4.2 Comparisons with the State-of-the-arts

We compare the segmentation performance of our method with other state-of-the-art CSS methods on two datasets, including Pascal-VOC 2012 and ADE 20K. The performance is evaluated with three metrics. The first one is the mIoU over the initial classes $\mathcal{C}_1$, and the second one measures the mIoU for all incremental classes $\mathcal{C}_{2:T}$. The third metric (all) denotes the mIoU for all observed classes $\mathcal{C}_{1:T}$. In experiments, We follow previous works [37, 110] by using Deeplab-v3 with the ResNet-101 backbone as the segmentation model. Following [16], the memory length $|\mathcal{M}|$ is 100 and 300 for

	100-50(2 stages)			**100-10(6 stages)**			**100-5(11 stages)**		
Method	0-100	101-150	all	0-100	101-150	all	0-100	101-150	all
Joint	44.34	28.21	39.00	44.34	28.21	39.00	44.34	28.21	39.00
ILT	18.29	14.40	17.00	0.11	3.06	1.09	0.08	1.31	0.49
MiB	40.52	17.17	32.79	38.21	11.12	29.24	36.01	5.66	25.96
SDR	37.40	24.80	33.20	12.13	28.94	34.48	33.02	10.63	25.61
PLOP	41.87	14.89	32.94	40.48	13.61	31.59	35.72	12.18	27.93
REMINDER	41.55	19.16	34.14	38.96	21.28	33.11	36.06	16.38	29.54
Ours	44.06	24.96	37.74	43.88	25.14	37.67	43.35	18.53	35.13

TABLE 5.2: Comparison results on ADE 20K.

Pascal-VOC 2012 and ADE20K, respectively. We adopt the widely-used pseudo label mechanism for training the segmentation network.

Table 5.1 presents the performance on Pascal-VOC 2012 for three different settings including 19-1 (2 stages), 15-5 (2 stages) and 15-1 (6 stages). Our method achieves state-of-the-art performance. On the three settings, our method achieves 77.66%, 73.73%, and 71.94% mIoUs on the 'all' metric, outperforming the second-place method by 3.29%, 1.33%, and 12.54%, respectively. The improvement is especially significant for the 15-1 (6 stages) setting, which is quite challenging due to the more severe catastrophic forgetting issue caused by a larger number of continuous stages. Our method, with carefully selecting and enhancing the replay samples, shows elevated effectiveness under such a challenging scenario.

The comparison results with the ADE 20K are shown in Table 5.2. For 3 different settings including 100-50 (2 stages), 100-10 (6 stages) and 100-5 (11 stages), our method achieves 37.74%, 37.67% and 35.13% mIoUs on the 'all' metric, improving the second-place one by 2.60%, 4.56% and 5.59% respectively, showing its effectiveness and advantage.

5.4.3 Comparison with Other Sample Selection Strategies

To verify the effectiveness of our RL-driven automatic replay mechanism, we validate and compare it with other sample selection methods in the CSS task. The experiments are conducted on Pascal-VOC 2012 under the 15-1 (6 stages) setting. The results are shown in Table.5.3. The compared methods include three types: 1) the random selection strategy; 2) the previously-proposed hand-crafted strategies including iCaRL [115], Rainbow [7], CBES [162] and SSUL [16]. Both iCaRL and Rainbow are diversity-based selection criteria. CBES and SSUL are two class-balanced sample selection strategies that are specially designed for CSS. Besides, to validate the effectiveness of the automatic learning mechanism, we also design a new hand-crafted strategy using the same factors as our method (sample diversity and class performance). The newly-designed one is based on our visualization of the learned policy introduced in Sec. 5.4.5. It shows selecting the common samples is effective for the hard classes with bad performance, while selecting the diverse samples is better for the simple classes with good performance. Thus, we design a strategy where the most common samples with the lowest diversity scores are selected for the top 50% low-performance classes, while the most diverse samples with the highest diversity scores are selected for other high-performance classes. We denote the new-designed (N) hand-crafted (H) strategy (S) as NHS. On the 'all' metric, random selection achieves 63.15% mIoU. By smartly selecting the appropriate samples based on heuristic rules, iCaRL, Rainbow CBES and SSUL achieve 65.62%, 66.09% and 66.39% and 66.37% mIoUs, respectively, and NHS further improves it to 66.82% by considering more factors with the complicated relationship. Considering these methods only select samples, for a fair comparison, we report the result of our method w/o the enhancement operation. It achieves 70.02% mIoU, not only outperforms the previously-proposed iCaRL, Rainbow, CBES and SSUL, showing the elevated effectiveness of the novel selection approach; but also outperforms NHS using the same set of factors, demonstrating the significant advantages of the reward-driven automatic policy learning mechanism over the hand-crafted strategies.

Selection Strategy	0-15	16-20	all
Random Selection	72.82	32.21	63.15
iCaRL [115]	73.91	39.11	65.62
Rainbow [7]	74.03	40.70	66.09
CBES [162]	74.15	41.57	66.39
SSUL [16]	74.20	41.33	66.37
NHS	74.50	42.25	66.82
Ours (w/o Enhancement)	77.54	45.98	70.02

TABLE 5.3: Comparison with other sample selection strategies. NHS denotes a new-designed hand-crafted strategy using the same factors as our method (sample diversity and class performance). For a fair comparison, we report the result of our method w/o the enhancement operation.

5.4.4 Ablation Study

In this section, we perform ablation study to verify the effectiveness of different components in our method. All experiments are conducted on Pascal-VOC 2012 under the 15-1 (6 stages) setting.

Ablation of Selection-enhancement Dual Stage Action

We conduct experiments to verify the effectiveness of the proposed selection-enhancement dual-stage action paradigm, with results shown in Table 5.4. Our method with both the sample selection and enhancement actions achieves 71.94% mIoU on the 'all' metric. By removing the enhancement operation, the performance decreases to 70.02%. By further removing both enhancement and selection procedures so that the memory is randomly filled, the performance is only 63.15%, 8.79% lower than our method. The results indicate that both the selection and enhancement operations can effectively boost CSS performance.

Method	0-15	16-20	all
Ours	78.54	50.82	71.94
Ours w/o Enhancement	77.54	45.98	70.02
Ours w/o Enhancement & Selection	72.82	32.21	63.15

TABLE 5.4: Ablation results of the selection-enhancement dual-stage action.

Ablation of State Representation Design

We then validate different components of the designed state representations and the results are presented in Table 5.5. The state representation contains three parts: 1) sample diversity div; 2) accuracy I and 3) forgetfulness g. The latter two constitute the class performance feature. In addition to validating the three parts, we also test using a common diversity metric instead of our novel one. Such a metric measures the inter-sample similarity by directly computing the distance between their prototype features. We name it as $div\_prototype$. Using div shows significant performance improvement (69.83% → 71.94%) to div_common, demonstrating the effectiveness of our novel graph-based similarity.

Method	0-15	16-20	all
Ours	78.54	50.82	71.94
Ours w/o div	74.09	33.33	64.39
Ours w/o I	76.50	42.08	68.30
Ours w/o g	77.79	45.32	70.06
Ours w/o $\{I, g\}$	76.18	36.03	66.68
Ours w/o div w/ $div\_prototype$	76.93	47.16	69.83

TABLE 5.5: Ablation results of the state representations.

Ablation of Memory Length.

In the results comparison section, for the fair comparison, we follow [16] by setting the memory length $\mathcal{M}$ to 100 and 300 for Pascal-VOC 2012 and ADE20K, respectively. We further validate the performance for the 15-1(6 stages) setting on Pascal-VOC 2012 by using memories with different lengths ranging from 50 to 300. The results are shown in Figure 5.3. We can observe that a larger memory brings better performance. As the memory length increases from 50 to 300, the mIoU on the 'all' metric increases from 63.56 to 75.33.

Ablation of Superpixel Number.

In order to compute the sample diversity, each region is divided into M superpixels for constructing the graph. Here we perform experimenters to validate how M affects the performance and present the results in Figure 5.4. As can be observed, the performance keeps stable when M is larger than 3 and smaller than 9, while a too large M leads to

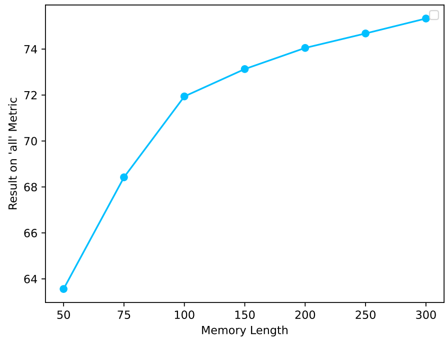


FIGURE 5.3: Ablation results of memory length. As the memory length increases from 50 to 300, the mIoU on the 'all' metric increases from 63.56 to 75.33.

the over segmenting that negatively affects the performance to some extent. Generally speaking, our method is non-sensitive to the hyper-parameter M, demonstrating its high robustness.

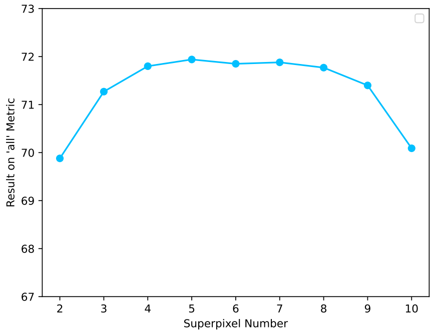


FIGURE 5.4: Ablation results of superpixel number.

5.4.5 Analysis of the Learned Policy

We further analyze the learned sample selection policy both qualitatively and quantitatively to offer more insights into how our method works. After analyzing the learned policy, we can observe the following rules:

(1) Low-performance classes require more replay samples. As shown in Figure 5.5, on Pascal-VOC 2012 dataset, we count the number of selected samples for different classes with different performances. From left to right, the horizontal axis represents the classes from low to high performance. We can find the negative correlation between class performance and the selected sample number. The low-performance classes are less accurate or more easily to be forgotten, so more samples are required for replay to alleviate the more severe catastrophic forgetting issue.

(2) Classes with different performances require different kinds of samples. We further investigate the learned strategy for classes with different performances. We visualize the diversity of the selected samples for three representative classes: 'chair',

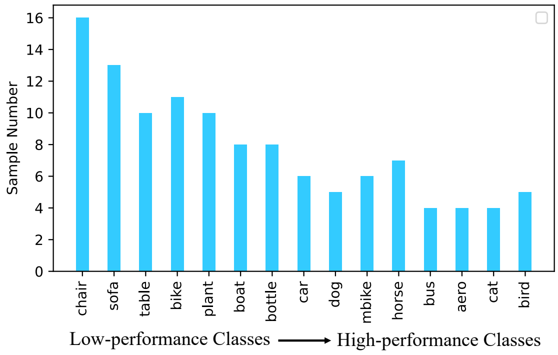
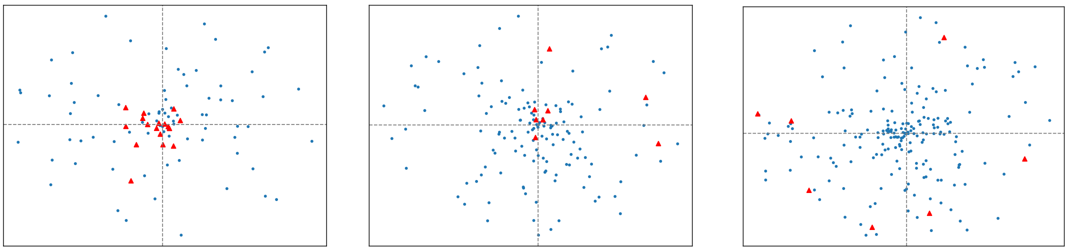


FIGURE 5.5: The numbers of selected samples for different classes. The horizontal axis from left to right represents classes from poor to good performance.

'boat', and 'bird.' 'chair' is a hard class with a low class performance, 'bird' is an easy class with a high class performance, and 'boat' has a medium class performance. The results are shown in Figure 5.6, where the red triangles represent the selected samples, and the blue dots denote other samples that are not selected. Triangles or dots closer to the center represent samples with lower diversity. As can be observed, for the low-performance class 'chair', most red triangles are distributed in the center, indicating the agent selects common samples with low diversity. On the contrary, for the high-performance class 'bird', the high-diversity samples are selected. For the middle-performance class 'boat', both the common and diverse samples are selected. We believe the different degrees of forgetfulness for different classes can explain the learned policy. For hard classes where the catastrophic forgetting is more severe, most samples including both the high-diversity novel ones and low-diversity common ones are forgotten after the model trains on new classes, so using the more common and representative samples can learn a classification space covering most samples. On the contrary, for easy classes with relatively minor catastrophic forgetting issues, the common samples can still be remembered in the next stage while the high-diversity samples are easier to be forgotten. Thus, replay with high-diversity samples can be more effective.



(A) Chair

(B) Boat

(C) Bird

FIGURE 5.6: (Best viewed in color). Visualization of the diversity for the selected samples of three classes including 'chair', 'boat' and 'bird'. The red triangles represent the selected samples and the blue dots denote other samples that are not selected. Triangles or dots closer to the center represent samples with the lower diversity.

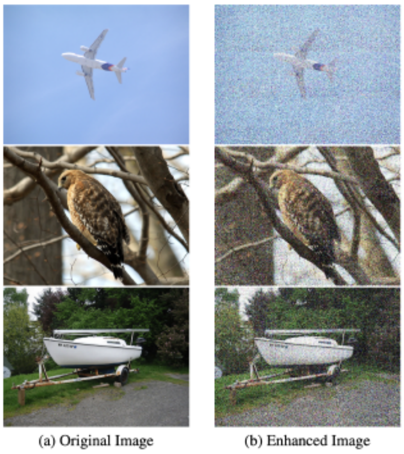


FIGURE 5.7: Comparison between the original images and images after enhancement.

5.4.6 Complexity Discussion

Training the agent network requires additional time. According to Alg. 1, the theoretically additional cost is $O(Y)$ higher than the time for deployment. However, we argue that agent training is an offline process and we can use a shallower segmentation network and a smaller dataset for training. With these simplifications, we get a computation-efficient agent training process where the training time is 1.16 times that of the deployment phase for the 15-5 (2 stages) setting on Pascal-VOC 2012. Also, the agent trained on one dataset can be deployed on other datasets. Thus the agent only needs to be trained once and can be deployed to different CSS tasks. Experimental results demonstrate this capability. Specifically, for the 'all' metric, using the agent trained on Pascal-VOC 2012 to deploy on the 100-50(2 stages) setting of ADE 20K achieves 34.87% mIoU, and using the agent trained on ADE 20K to deploy on 19-1(2 stages) setting of Pascal-VOC 2012 achieves 74.96% mIoU, with both cases showing good performance. The results demonstrate the high generalization of our method. In realistic applications, the agent only needs to be trained once and then can be used on several different CSS tasks without the extra computation cost for agent retraining.

5.4.7 Visualizations

Visualization of Sample Enhancement

Our method includes a novel enhancement action. It enables the selected samples to have the better replay effectiveness by maximizing their agent scores through gradient-based editing. We presents some comparison results between original images and the enhanced images in Figure 5.7. We also provide a quantitative comparison in Figure 5.8 to show the agent score distributions for all selected samples before and after enhancement, where the horizontal axis represents different score intervals, and the vertical axis indicates the proportion of samples falling into each interval. We can observe that after enhancement, there are more samples with high agent scores. This demonstrates

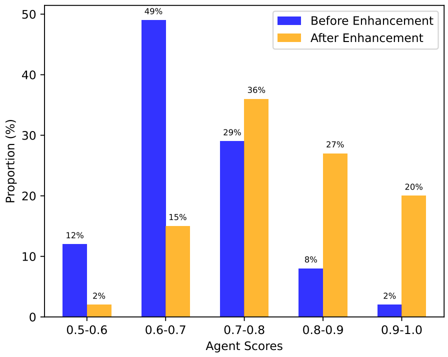


FIGURE 5.8: Comparison of agent score distributions for all selected samples before and after enhancement. The horizontal axis represents different score intervals. The vertical axis indicates the proportion of samples falling into each interval.

that the gradient-based enhancement effectively increases agent scores, thus promoting the replay performance.

Visualization of Segmentation Results

In Figure 5.9, we present the segmentation results on the Pascal-VOC 2012 validation set using the model trained in the CSS task. We compare our method with the replay approach using the randomly selected samples. Thanks to the proposed mechanism that automatically learns an optimal policy and uses it to select and enhance the most adequate samples, our method can be more effective to alleviate the catastrophic forgetting problem in CSS, thus achieving the better results.

5.5 Conclusion

In this chapter, we propose a novel and automatic memory selection paradigm. It significantly facilitates alleviating the severe catastrophic forgetting issue through more effective memory management in the Continual Semantic Segmentation (CSS) task. We propose a novel learning-based approach with an agent network to automatically learn the policy. The input representation to the agent network is tailored for the CSS task. We also use the agent network to further perform a novel sample enhancement operation through a gradient-based approach to boost the effectiveness of selected samples. This research work provides valuable insights into the memory selection of continual semantic segmentation and practical tools that is readily applicable. Our method is effective and general, as shown by our extensive experiments with state-of-the-art (SOTA) performance.

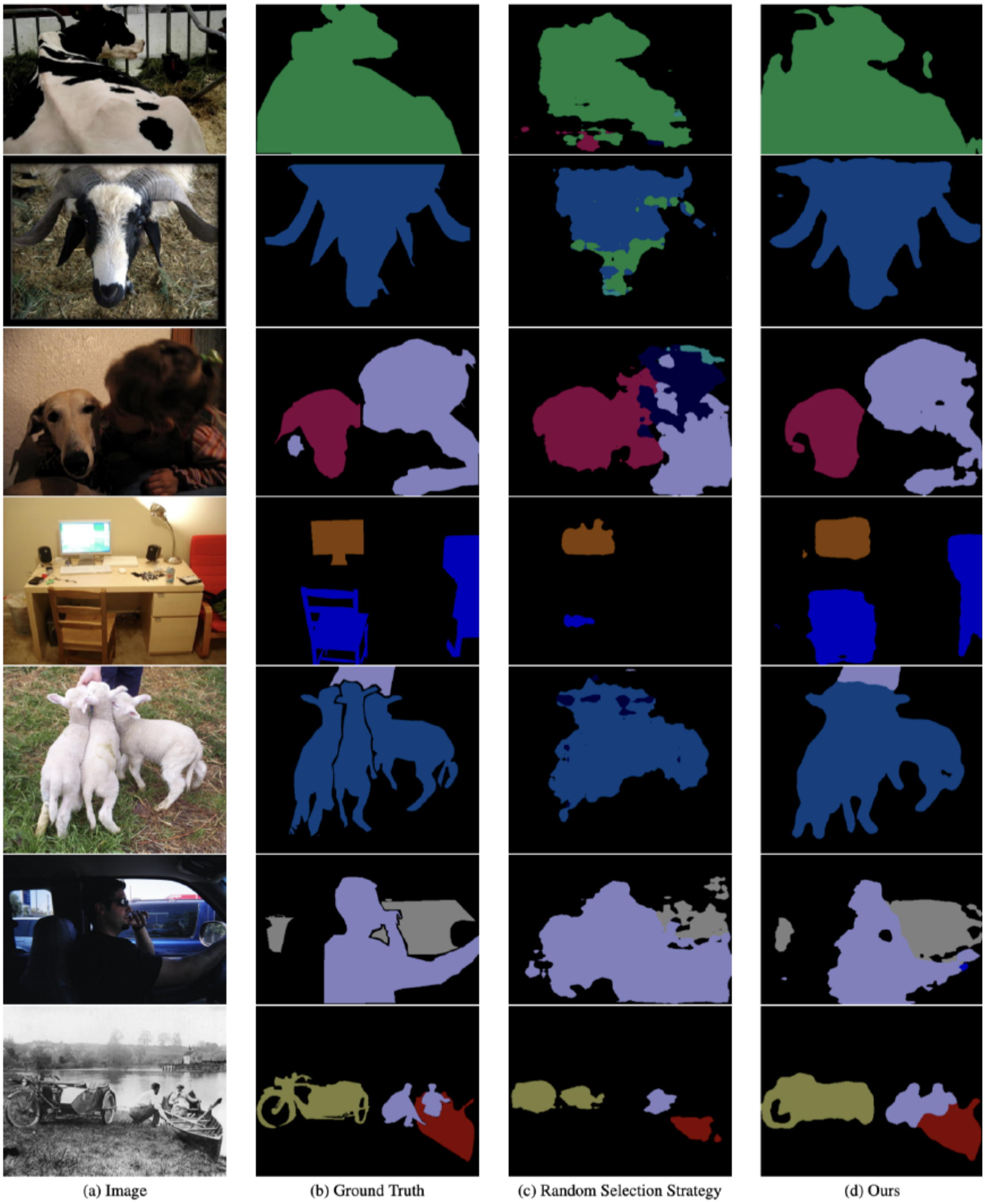


FIGURE 5.9: The segmentation visualization comparison results between our method and random selection strategy.

Chapter 6

Conclusions and Future Work

In this thesis, we mainly focus on addressing two issues encountered in the multi-stage segmentation training paradigms where data is provided incrementally rather than all at once: (1) low performance due to insufficient training data in the incomplete dataset, and (2) the problem of catastrophic forgetting where previously learned knowledge is often lost after training on new data.

We first propose a novel method for few-shot segmentation, aimed at training new classes with minimal samples. This research identifies the issue of background context bias in existing few-shot segmentation approaches and introduces an innovative solution to mitigate this problem, leading to improved performance. The proposed method employs an iterative framework consisting of three successive steps: Query Prediction, Support Modulation, and Information Cleansing. This structure effectively addresses foreground feature misalignment and reduces noise accumulation, establishing a recurrent optimization process that continuously refines segmentation results. Experiments conducted on multiple datasets demonstrate the efficacy of our approach, which achieves state-of-the-art (SOTA) segmentation performance. This work contributes valuable insights and advancements to the field of few-shot segmentation, enhancing the efficiency and effectiveness of segmentation model training.

While the first work presented in this thesis demonstrates notable improvements and high effectiveness, it remains constrained by its reliance on the limited and potentially biased information derived from a small number of labeled samples. This limitation inherently restricts the segmentation model's ability to achieve higher performance. To address this challenge, we leverage the rich knowledge embedded in large language models (LLMs) for few-shot segmentation, proposing a novel framework named LLaFS++, which achieves significant advancements. To adapt LLMs for this visual task, we introduce a segmentation task instruction to provide detailed task definitions, and propose a fine-grained in-context instruction to simulate human cognitive mechanisms and provide fine-grained multimodal reference information. We also propose a pseudo-sample-based curriculum pretraining mechanism to augment the training samples required for instruction tuning, and introduce a novel inference method to mitigate potential oversegmentation hallucinations caused by the regional guidance information. Our thorough experiments validate the effectiveness of LLaFS++, showcasing its capability to consistently surpass existing state-of-the-art methods across various datasets and scenarios. We consider LLaFS++ as a significant step forward in exploring how LLM frameworks can be effectively leveraged for addressing few-shot challenges within the domain of computer vision.

We also propose a new method for the task of continual semantic segmentation, focused on addressing the issue of catastrophic forgetting, enabling the model to retain its ability to handle previous classes after training on new class data. In this method, we propose an automatic selection mechanism through reinforcement learning, choosing suitable samples from previous stages for replay in future stages and thus mitigating catastrophic forgetting. Extensive experiments across multiple datasets and protocols demonstrate the high effectiveness and generalization of this method, which achieves state-of-the-art performance while utilizing only 1% of the total training data for replay. We consider this work as an important method that can provide valuable insights into memory selection in continual semantic segmentation, making the replay methods in this research domain more effective and and the catastrophic forgetting problem to be mitigated better.

In recent years, the rapid advancement of large vision-language models (LVLMs) has opened up new possibilities for image segmentation. My future work will focus on developing the more generalizable, reliable, and efficient LVLM-based segmentation approaches with the following characteristics:

- **(a) Higher Generalizability**. To enhance the versatility and practical applicability, the LVLM-based segmentation models are expected to exhibit robust generalization capabilities across various dimensions, including: *(1) Domain generalizability*: The ability to handle inputs from diverse domains, such as medical images, remote sensing images, and more. *(2) Modality generalizability*: The capacity to process segmentation tasks across different modalities, such as images, videos, point clouds, and 3D scenes. *(3) Task generalizability*: The capability to tackle a wide range of segmentation tasks, including semantic segmentation, instance segmentation, panoptic segmentation, in-context segmentation, referring segmentation, and reasoning-based segmentation. A highly generalizable model offers greater utility in real-world applications, making it more valuable for practical deployment.

- **(b) Enhanced Reliability**. LVLMs are prone to hallucination, often generating outputs entirely unrelated to the input. This issue also manifests in LVLM-based segmentation methods, which frequently produce segmentation masks that are not relevant to the input. To mitigate hallucination and improve the reliability of LVLM-based segmentation methods, research efforts should be focused on the following aspects: *(1) Higher-quality datasets*: Construct strictly curated, high-quality segmentation datasets with less noise to minimize erroneous information. *(2) Improved training methods*: Employ advanced techniques such as knowledge augmentation and human preference learning to enhance training effectiveness and improve model performance. *(3) Better inference mechanisms:* Incorporate methods like verification mechanisms, self-reflection, and chain-of-thought reasoning during inference to enhance reliability.

- **(c) Improved efficiency**. Large models typically require substantial computational and data resources during both training and inference. Developing more efficient LVLM-based segmentation methods is essential for facilitating their practical deployment. Specifically, attentions should be focused on the following aspects: *(1) Training efficiency*: Reduce the training time, the number of GPUs required, and the amount of training data needed for LVLM-based segmentation models. *(2) Inference efficiency*: Minimize the model's parameter size to enable more efficient deployment and faster inference speeds. Enhancing efficiency in both training and inference stages will make LVLM-based segmentation methods more accessible and feasible for real-world applications.

Developing more **generalizable, reliable, and efficient** LVLM-based segmentation methods can enhance model performance, improve safety, and reduce usage costs. These advancements are crucial for enabling the practical application of such methods in domains that demand stability, generalization, and computational efficiency, such as autonomous driving and robotic control. Therefore, this will be a key focus of my future research directions.

Bibliography

[1] Rahaf Aljundi et al. “Gradient based sample selection for online continual learning”. In: *Proceedings of the Advances in Neural Information Processing Systems* 32 (2019).

[2] Manoj Alwani, Yang Wang, and Vashisht Madhavan. “DECORE: Deep Compression with Reinforcement Learning”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12349–12359.

[3] Salem Saleh Al-Amri, Namdeo V Kalyankar, et al. “Image segmentation by using threshold techniques”. In: *arXiv preprint arXiv:1005.4020* (2010).

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495.

[5] Jinze Bai et al. “Qwen-vl: A frontier large vision-language model with versatile abilities”. In: *arXiv preprint arXiv:2308.12966* (2023).

[6] Yuntao Bai et al. “Training a helpful and harmless assistant with reinforcement learning from human feedback”. In: *arXiv preprint arXiv:2204.05862* (2022).

[7] Jihwan Bang et al. “Rainbow memory: Continual learning with a memory of diverse samples”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8218–8227.

[8] Andreas Bieniek and Alina Moga. “An efficient watershed algorithm based on connected components”. In: *Pattern Recognition* 33.6 (2000), pp. 907–916.

[9] Zalán Borsos, Mojmir Mutny, and Andreas Krause. “Coresets via bilevel optimization for continual learning and streaming”. In: *Proceedings of the Advances in Neural Information Processing Systems* 33 (2020), pp. 14879–14890.

[10] Malik Boudiaf et al. “Few-shot segmentation without meta-learning: A good transductive inference is all you need?” In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13979–13988.

[11] Tom Brown et al. “Language models are few-shot learners”. In: *Proceedings of the Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901.

[12] Adrian Bulat et al. “FS-DETR: Few-Shot DEtection TRansformer with prompting and without re-training”. In: *International Conference on Computer Vision*. 2023, pp. 11793–11802.

[13] Nicolas Carion et al. “End-to-end object detection with transformers”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 213–229.

[14] Fabio Cermelli, Matthieu Cord, and Arthur Douillard. “Comformer: Continual learning in semantic and panoptic segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3010–3020.

[15] Fabio Cermelli et al. "Modeling the background for incremental learning in semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9233–9242.

[16] Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. "SSUL: Semantic Segmentation with Unknown Label for Exemplar-based Class-Incremental Learning". In: *Proceedings of the Advances in Neural Information Processing Systems* 34 (2021), pp. 10919–10930.

[17] Kai Chen et al. "Hybrid task cascade for instance segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4974–4983.

[18] Liang-Chieh Chen et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[19] Liang-Chieh Chen et al. "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[20] Liang-Chieh Chen et al. "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062* (2014).

[21] Tianrun Chen et al. "Deep3dsketch: 3d modeling from free-hand sketches with view-and structural-aware adversarial training". In: *arXiv preprint arXiv:2312.04435* (2023).

[22] Tianrun Chen et al. "Deep3dsketch+: Rapid 3d modeling from single free-hand sketches". In: *International Conference on Multimedia Modeling*. Springer. 2023, pp. 16–28.

[23] Tianrun Chen et al. "Img2CAD: Conditioned 3-D CAD Model Generation From Single Image With Structured Visual Geometry". In: *IEEE Transactions on Industrial Informatics* (2025).

[24] Tianrun Chen et al. "Reality3dsketch: Rapid 3d modeling of objects from single freehand sketches". In: *IEEE Transactions on Multimedia* 26 (2023), pp. 4859–4870.

[25] Tianrun Chen et al. "Reasoning3D–Grounding and Reasoning in 3D: Fine-Grained Zero-Shot Open-Vocabulary 3D Reasoning Part Segmentation via Large Vision-Language Models". In: *arXiv preprint arXiv:2405.19326* (2024).

[26] Tianrun Chen et al. "Sam-adapter: Adapting segment anything in underperformed scenes". In: *International Conference on Computer Vision*. 2023, pp. 3367–3375.

[27] Tianrun Chen et al. "SAM2-Adapter: Evaluating & Adapting Segment Anything 2 in Downstream Tasks: Camouflage, Shadow, Medical Image Segmentation, and More". In: *arXiv preprint arXiv:2408.04579* (2024).

[28] Tianrun Chen et al. "xLSTM-UNet can be an Effective Backbone for 2D & 3D Biomedical Image Segmentation Better than its Mamba Counterparts". In: *2024 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE. 2024, pp. 1–8.

[29] Ting Chen et al. "A generalist framework for panoptic segmentation of images and videos". In: *International Conference on Computer Vision*. 2023, pp. 909–919.

[30] Bowen Cheng, Alex Schwing, and Alexander Kirillov. “Per-pixel classification is not all you need for semantic segmentation”. In: *Proceedings of the Advances in Neural Information Processing Systems* 34 (2021), pp. 17864–17875.

[31] Bowen Cheng et al. “Masked-attention mask transformer for universal image segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1290–1299.

[32] Bowen Cheng et al. “Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12475–12485.

[33] Gong Cheng, Chunbo Lang, and Junwei Han. “Holistic prototype activation for few-shot segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.4 (2022), pp. 4650–4666.

[34] Wei Cong et al. “Cs2K: Class-specific and Class-shared Knowledge Guidance for Incremental Semantic Segmentation”. In: *European Conference on Computer Vision*. 2024.

[35] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[36] Marco Cuturi. “Sinkhorn distances: Lightspeed computation of optimal transport”. In: *Proceedings of the Advances in Neural Information Processing Systems* 26 (2013).

[37] Arthur Douillard et al. “Plop: Learning without forgetting for continual semantic segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4040–4050.

[38] Jinhao Du et al. “σ-Adaptive Decoupled Prototype for Few-Shot Object Detection”. In: *International Conference on Computer Vision*. IEEE. 2023, pp. 18904–18914.

[39] Mark Everingham et al. “The pascal visual object classes challenge: A retrospective”. In: *International Journal of Computer Vision* 111 (2015), pp. 98–136.

[40] Mingyuan Fan et al. “Rethinking BiSeNet For Real-time Semantic Segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9716–9725.

[41] Qi Fan et al. “Self-support few-shot semantic segmentation”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 701–719.

[42] Jun Fu et al. “Dual attention network for scene segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3146–3154.

[43] Xiao Fu et al. “Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation”. In: *International Conference on 3D Vision*. IEEE. 2022, pp. 1–11.

[44] Shivam Garg et al. “What can transformers learn in-context? a case study of simple function classes”. In: *Proceedings of the Advances in Neural Information Processing Systems* 35 (2022), pp. 30583–30598.

[45] Jia Gong et al. “Meta agent teaming active learning for pose estimation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11079–11089.

[46] Yizheng Gong et al. "Continual Segmentation with Disentangled Objectness Learning and Class Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024, pp. 3848–3857.

[47] Rafael C Gonzales and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[48] Jiaqi Gu et al. "Multi-Scale High-Resolution Vision Transformer for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2022, pp. 12094–12103.

[49] Jiaxin Guo et al. "A Visual Navigation Perspective for Category-Level Object Pose Estimation". In: *arXiv preprint arXiv:2203.13572* (2022).

[50] Sina Hajimiri et al. "A strong baseline for generalized few-shot semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 11269–11278.

[51] Kaiming He et al. "Mask r-cnn". In: *International Conference on Computer Vision*. 2017, pp. 2961–2969.

[52] Sunghwan Hong et al. "Cost aggregation with 4d convolutional swin transformer for few-shot segmentation". In: *European Conference on Computer Vision*. Springer. 2022, pp. 108–126.

[53] Mir Rayat Imtiaz Hossain et al. "Visual Prompting for Generalized Few-shot Segmentation: A Multi-scale Approach". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024.

[54] Neil Houlsby et al. "Parameter-efficient transfer learning for NLP". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2790–2799.

[55] Edward J Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. 2021.

[56] Tao Hu et al. "Attention-based multi-context guiding for few-shot semantic segmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8441–8448.

[57] Kai Huang et al. "Prototypical Kernel Learning and Open-set Foreground Perception for Generalized Few-shot Semantic Segmentation". In: *International Conference on Computer Vision*. 2023, pp. 19256–19265.

[58] David Isele and Akansel Cosgun. "Selective experience replay for lifelong learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[59] Jitesh Jain et al. "Oneformer: One transformer to rule universal image segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2989–2998.

[60] Deyi Ji et al. "Discrete Latent Perspective Learning for Segmentation and Detection". In: *International Conference on Machine Learning*. 2024.

[61] Deyi Ji et al. "Structural and Statistical Texture Knowledge Distillation for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2022, pp. 16876–16885.

[62] Deyi Ji et al. "Tree-of-table: Unleashing the power of llms for enhanced large-scale table understanding". In: *arXiv preprint arXiv:2411.08516* (2024).

[63] Deyi Ji et al. "View-centric multi-object tracking with homographic matching in moving uav". In: *arXiv preprint arXiv:2403.10830* (2024).

[64] Siyu Jiao et al. "Mask Matching Transformer for Few-Shot Segmentation". In: *Proceedings of the Advances in Neural Information Processing Systems*.

[65] Xisen Jin et al. "Gradient-based Editing of Memory Examples for Online Task-free Continual Learning". In: *arXiv preprint arXiv:2006.15294* (2020).

[66] Prannay Kaul, Weidi Xie, and Andrew Zisserman. "Label, verify, correct: A simple few shot object detection method". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14237–14247.

[67] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. "Occlusion-aware instance segmentation via bilayer network architectures". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[68] Alexander Kirillov et al. "Panoptic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9404–9413.

[69] Alexander Kirillov et al. "Segment anything". In: *International Conference on Computer Vision*. 2023, pp. 4015–4026.

[70] Xin Lai et al. "Lisa: Reasoning segmentation via large language model". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024.

[71] Chunbo Lang et al. "Base and meta: A new perspective on few-shot segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[72] Chunbo Lang et al. "Learning what not to segment: A new perspective on few-shot segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8057–8067.

[73] Youngwan Lee and Jongyoul Park. "Centermask: Real-time anchor-free instance segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13906–13915.

[74] Gen Li et al. "Adaptive prototype learning and allocation for few-shot segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8334–8343.

[75] Junnan Li et al. "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". In: *International Conference on Machine Learning* (2023).

[76] Liulei Li et al. "Semantic hierarchy-aware segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[77] Xiang Li et al. "Fss-1000: A 1000-class dataset for few-shot segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2869–2878.

[78] Xiang Lisa Li et al. "Contrastive decoding: Open-ended text generation as optimization". In: *arXiv preprint arXiv:2210.15097* (2022).

[79] Xiangtai Li et al. "Semantic Flow for Fast and Accurate Scene Parsing". In: *European Conference on Computer Vision*. Springer. 2020, pp. 775–793.

[80] Zhiqi Li et al. “Panoptic segformer: Delving deeper into panoptic segmentation with transformers”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1280–1289.

[81] Zihan Lin, Zilei Wang, and Yixin Zhang. “Continual Semantic Segmentation via Structure Preserving and Projected Feature Alignment”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 345–361.

[82] Zihan Lin, Zilei Wang, and Yixin Zhang. “Preparing the Future for Continual Semantic Segmentation”. In: *International Conference on Computer Vision*. 2023, pp. 11910–11920.

[83] Haotian Liu et al. “Visual Instruction Tuning”. In: *Proceedings of the Advances in Neural Information Processing Systems*. 2023.

[84] Jiang Liu et al. “PolyFormer: Referring image segmentation as sequential polygon generation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18653–18663.

[85] Weide Liu et al. “Crnet: Cross-reference networks for few-shot segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4165–4173.

[86] Xiao Liu et al. “P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks”. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2022, pp. 61–68.

[87] Xinyu Liu et al. “Delving into Shape-aware Zero-shot Semantic Segmentation”. In: *arXiv preprint arXiv:2304.08491* (2023).

[88] Yaoyao Liu, Bernt Schiele, and Qianru Sun. “RMM: Reinforced memory management for class-incremental learning”. In: *Proceedings of the Advances in Neural Information Processing Systems* 34 (2021), pp. 3478–3490.

[89] Yongfei Liu et al. “Part-aware prototype network for few-shot semantic segmentation”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 142–158.

[90] Yuanwei Liu et al. “Intermediate prototype mining transformer for few-shot semantic segmentation”. In: *Proceedings of the Advances in Neural Information Processing Systems* 35 (2022), pp. 38020–38031.

[91] Yuanwei Liu et al. “Learning non-target knowledge for few-shot semantic segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11573–11582.

[92] Zhikang Liu and Lanyun Zhu. “Label-guided attention distillation for lane segmentation”. In: *Neurocomputing* 438 (2021), pp. 312–322.

[93] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[94] Xiaoliu Luo et al. “Pfenet++: Boosting few-shot semantic segmentation with the noise-filtered context-aware prior mask”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[95] Andrea Maracani et al. "Recall: Replay-based continual learning in semantic segmentation". In: *International Conference on Computer Vision*. 2021, pp. 7026–7035.

[96] Umberto Michieli and Pietro Zanuttigh. "Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1114–1124.

[97] Umberto Michieli and Pietro Zanuttigh. "Incremental learning techniques for semantic segmentation". In: *International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[98] Juhong Min, Dahyun Kang, and Minsu Cho. "Hypercorrelation squeeze for few-shot segmentation". In: *International Conference on Computer Vision*. 2021, pp. 6941–6952.

[99] Sewon Min et al. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 11048–11064.

[100] Rohit Mohan and Abhinav Valada. "Efficientps: Efficient panoptic segmentation". In: *International Journal of Computer Vision* 129.5 (2021), pp. 1551–1579.

[101] Jishnu Mukhoti et al. "Open vocabulary semantic segmentation with patch aligned contrastive learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19413–19423.

[102] Gregory L Murphy and Douglas L Medin. "The role of theories in conceptual coherence." In: *Psychological review* 92.3 (1985), p. 289.

[103] Khoi Nguyen and Sinisa Todorovic. "Feature weighting and boosting for few-shot segmentation". In: *International Conference on Computer Vision*. 2019, pp. 622–631.

[104] Jiahao Nie et al. *Cross-Domain Few-Shot Segmentation via Iterative Support-Query Correspondence Mining*. 2024.

[105] Youngmin Oh, Donghyeon Baek, and Bumsub Ham. "Alife: Adaptive logit regularizer and feature replay for incremental semantic segmentation". In: *Proceedings of the Advances in Neural Information Processing Systems* 35 (2022), pp. 14516–14528.

[106] Atsuro Okazawa. "Interclass Prototype Relation for Few-Shot Segmentation". In: *European Conference on Computer Vision*. Springer. 2022, pp. 362–378.

[107] Gilhan Park et al. "Mitigating Background Shift in Class-Incremental Semantic Segmentation". In: *European Conference on Computer Vision*. 2024.

[108] Baolin Peng et al. "Instruction tuning with gpt-4". In: *arXiv preprint arXiv:2304.03277* (2023).

[109] Bohao Peng et al. "Hierarchical Dense Correlation Distillation for Few-Shot Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 23641–23651.

[110] Minh Hieu Phan et al. "Class Similarity Weighted Knowledge Distillation for Continual Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16866–16875.

[111] Jie Qin et al. “Freeseg: Unified, universal and open-vocabulary image segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19446–19455.

[112] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[113] Hanoona Rasheed et al. “Glamm: Pixel grounding large multimodal model”. In: *arXiv preprint arXiv:2311.03356* (2023).

[114] Nikhila Ravi et al. “Sam 2: Segment anything in images and videos”. In: *arXiv preprint arXiv:2408.00714* (2024).

[115] Sylvestre-Alvise Rebuffi et al. “icarl: Incremental classifier and representation learning”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2001–2010.

[116] Zhongwei Ren et al. “PixelLM: Pixel Reasoning with Large Multimodal Model”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024.

[117] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[118] Baptiste Roziere et al. “Code llama: Open foundation models for code”. In: *arXiv preprint arXiv:2308.12950* (2023).

[119] Amirreza Shaban et al. “One-shot learning for semantic segmentation”. In: *arXiv preprint arXiv:1709.03410* (2017).

[120] Chao Shang et al. “Incrementer: Transformer for class-incremental semantic segmentation with knowledge distillation focusing on old class”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7214–7224.

[121] Xinyu Shi et al. “Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 151–168.

[122] Dongsub Shim et al. “Online class-incremental continual learning with adversarial shapley value”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 11. 2021, pp. 9630–9638.

[123] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. “Amp: Adaptive masked proxies for few-shot segmentation”. In: *International Conference on Computer Vision*. 2019, pp. 5249–5258.

[124] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), pp. 484–489.

[125] Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. “Reinforcement learning in robotic applications: a comprehensive survey”. In: *Artificial Intelligence Review* 55.2 (2022), pp. 945–990.

[126] Yixuan Su et al. “Pandagpt: One model to instruction-follow them all”. In: *arXiv preprint arXiv:2305.16355* (2023).

[127] Yanpeng Sun et al. "VRP-SAM: SAM with visual reference prompt". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024, pp. 23565–23574.

[128] Maoqing Tian et al. "Eliminating background-bias for robust person re-identification". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5794–5803.

[129] Zhi Tian et al. "Instance and panoptic segmentation using conditional convolutions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2022), pp. 669–680.

[130] Zhuotao Tian et al. "Generalized few-shot semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11563–11572.

[131] Zhuotao Tian et al. "Prior guided feature enrichment network for few-shot segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.2 (2020), pp. 1050–1065.

[132] Rishabh Tiwari et al. "GCR: Gradient Coreset Based Replay Buffer Selection For Continual Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 99–108.

[133] Marco Toldo, Umberto Michieli, and Pietro Zanuttigh. "Learning with style: Continual semantic segmentation across tasks and domains". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[134] Hugo Touvron et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[135] Jasper RR Uijlings et al. "Selective search for object recognition". In: *International Journal of Computer Vision* 104 (2013), pp. 154–171.

[136] Hado Van Hasselt, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1. 2016.

[137] Haochen Wang et al. "Few-shot semantic segmentation with democratic attention networks". In: *European Conference on Computer Vision*. Springer. 2020, pp. 730–746.

[138] Huyong Wang, Huisi Wu, and Jing Qin. "Incremental Nuclei Segmentation from Histopathological Images via Future-class Awareness and Compatibility-inspired Distillation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024, pp. 11408–11417.

[139] Jin Wang et al. "Rethinking Prior Information Generation with CLIP for Few-Shot Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024.

[140] Kaixin Wang et al. "Panet: Few-shot image semantic segmentation with prototype alignment". In: *International Conference on Computer Vision*. 2019, pp. 9197–9206.

[141] Wenhai Wang et al. "Visionllm: Large language model is also an open-ended decoder for vision-centric tasks". In: *arXiv preprint arXiv:2305.11175* (2023).

[142] Wenjian Wang et al. "Remember the difference: Cross-domain few-shot semantic segmentation via meta-memory transfer". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7065–7074.

[143] Xin Wang, Yudong Chen, and Wenwu Zhu. "A survey on curriculum learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), pp. 4555–4576.

[144] Xinlong Wang et al. "Solo: A simple framework for instance segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.11 (2021), pp. 8587–8601.

[145] Yan Wang et al. "FVP: Fourier Visual Prompting for Source-Free Unsupervised Domain Adaptation of Medical Image Segmentation". In: *arXiv preprint arXiv:2304.13672* (2023).

[146] Yuan Wang, Naisong Luo, and Tianzhu Zhang. "Focus on query: Adversarial mining transformer for few-shot segmentation". In: *Proceedings of the Advances in Neural Information Processing Systems* 36 (2023), pp. 31524–31542.

[147] Yuan Wang, Rui Sun, and Tianzhu Zhang. "Rethinking the Correlation in Few-Shot Segmentation: A Buoys View". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7183–7192.

[148] Yuan Wang et al. "Adaptive Agent Transformer for Few-Shot Segmentation". In: *European Conference on Computer Vision*. Springer. 2022, pp. 36–52.

[149] Zhenyu Wang et al. "Combating noise: semi-supervised learning by region uncertainty quantification". In: *Proceedings of the Advances in Neural Information Processing Systems* 34 (2021), pp. 9534–9545.

[150] Edward J Wisniewski and Bradley C Love. "Relations versus properties in conceptual combination". In: *Journal of memory and language* 38.2 (1998), pp. 177–202.

[151] Wei Wu et al. "Temporal Complementarity-Guided Reinforcement Learning for Image-to-Video Person Re-Identification". In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2022, pp. 7319–7328.

[152] Zhuofan Xia et al. "Gsva: Generalized segmentation via multimodal large language models". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024, pp. 3858–3869.

[153] Jia-Wen Xiao et al. "Endpoints weight fusion for class incremental semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7204–7213.

[154] Enze Xie et al. "Polarmask: Single shot instance segmentation with polar representation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12193–12202.

[155] Enze Xie et al. "Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), pp. 5385–5400.

[156] Enze Xie et al. “SegFormer: Simple and efficient design for semantic segmentation with transformers”. In: *Proceedings of the Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.

[157] Guo-Sen Xie et al. “Scale-aware graph neural network for few-shot semantic segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5475–5484.

[158] Jingyi Xu, Hieu Le, and Dimitris Samaras. “Generating features with increased crop-related diversity for few-shot object detection”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19713–19722.

[159] Qianxiong Xu et al. “Hybrid Mamba for Few-Shot Segmentation”. In: *Proceedings of the Advances in Neural Information Processing Systems*. 2024.

[160] Qianxiong Xu et al. “Self-Calibrated Cross Attention Network for Few-Shot Segmentation”. In: *International Conference on Computer Vision*. 2023.

[161] Qianxiong Xu et al. “Unlocking the Power of SAM 2 for Few-Shot Segmentation”. In: *arXiv preprint arXiv:2505.14100* (2025).

[162] Shipeng Yan et al. “An em framework for online incremental learning of semantic segmentation”. In: *ACM International Conference on Multimedia*. 2021, pp. 3052–3060.

[163] Lihe Yang et al. “Mining latent classes for few-shot segmentation”. In: *International Conference on Computer Vision*. 2021, pp. 8721–8730.

[164] Yong Yang et al. “MIANet: Aggregating Unbiased Instance and General Information for Few-Shot Semantic Segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7131–7140.

[165] Jaehong Yoon et al. “Online coreset selection for rehearsal-based continual learning”. In: *arXiv preprint arXiv:2106.01085* (2021).

[166] Changqian Yu et al. “Bisenet: Bilateral segmentation network for real-time semantic segmentation”. In: *European Conference on Computer Vision*. 2018, pp. 325–341.

[167] Bo Yuan, Danpei Zhao, and Zhenwei Shi. “Learning At a Glance: Towards Interpretable Data-Limited Continual Semantic Segmentation Via Semantic-Invariance Modelling”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[168] Yuqian Yuan et al. “Osprey: Pixel Understanding with Visual Instruction Tuning”. In: *arXiv preprint arXiv:2312.10032* (2023).

[169] Ying Zang et al. “From Air to Wear: Personalized 3D Digital Fashion with AR/VR Immersive 3D Sketching”. In: *IEEE Transactions on Visualization and Computer Graphics* (2025).

[170] Ying Zang et al. “Resmatch: Referring expression segmentation in a semi-supervised manner”. In: *Information Sciences* 694 (2025), p. 121709.

[171] Anqi Zhang et al. “Bridge the Points: Graph-based Few-shot Segment Anything Semantically”. In: *arXiv preprint arXiv:2410.06964* (2024).

[172] Chang-Bin Zhang et al. "Representation Compensation Networks for Continual Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7053–7064.

[173] Chi Zhang et al. "Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5217–5226.

[174] Chi Zhang et al. "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation". In: *International Conference on Computer Vision*. 2019, pp. 9587–9595.

[175] Gengwei Zhang et al. "Few-shot segmentation via cycle-consistent transformer". In: *Proceedings of the Advances in Neural Information Processing Systems* 34 (2021), pp. 21984–21996.

[176] Shan Zhang et al. "Kernelized few-shot object detection with efficient integral aggregation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19207–19216.

[177] Zekang Zhang et al. "Coinseg: Contrast inter-and intra-class representations for incremental segmentation". In: *International Conference on Computer Vision*. 2023, pp. 843–853.

[178] Zheng Zhang et al. "PSALM: Pixelwise SegmentAtion with Large Multi-Modal Model". In: *arXiv preprint arXiv:2403.14598* (2024).

[179] Hengshuang Zhao et al. "Pyramid scene parsing network". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2881–2890.

[180] Bolei Zhou et al. "Scene parsing through ade20k dataset". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 633–641.

[181] Deyao Zhu et al. "Minigpt-4: Enhancing vision-language understanding with advanced large language models". In: *arXiv preprint arXiv:2304.10592* (2023).

[182] Lanyun Zhu et al. "Addressing Background Context Bias in Few-Shot Segmentation through Iterative Modulation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 3370–3379.

[183] Lanyun Zhu et al. "Continual semantic segmentation with automatic memory sample selection". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3082–3092.

[184] Lanyun Zhu et al. "CPCF: A Cross-Prompt Contrastive Framework for Referring Multimodal Large Language Models". In: *Forty-second International Conference on Machine Learning*. 2025.

[185] Lanyun Zhu et al. "IBD: Alleviating Hallucinations in Large Vision-Language Models via Image-Biased Decoding". In: *arXiv preprint arXiv:2402.18476* (2024).

[186] Lanyun Zhu et al. "Learning Gabor Texture Features for Fine-Grained Recognition". In: *International Conference on Computer Vision*. 2023, pp. 1621–1631.

[187] Lanyun Zhu et al. "Learning Statistical Texture for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12537–12546.

[188] Lanyun Zhu et al. "Llafs: When large language models meet few-shot segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2024.

[189] Lanyun Zhu et al. "LLaFS++: Few-Shot Image Segmentation With Large Language Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[190] Lanyun Zhu et al. "Not every patch is needed: Towards a more efficient and effective backbone for video-based person re-identification". In: *IEEE Transactions on Image Processing* (2025).

[191] Lanyun Zhu et al. "Popen: Preference-based optimization and ensemble for lvlm-based reasoning segmentation". In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 30231–30240.

[192] Lanyun Zhu et al. "Replay Master: Automatic Sample Selection and Effective Memory Utilization for Continual Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[193] Zhen Zhu et al. "Asymmetric non-local neural networks for semantic segmentation". In: *International Conference on Computer Vision*. 2019, pp. 593–602.