Towards Image Semantic Segmentation: From Context to Language

by **Huadong Tang**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Qiang Wu and Min Xu

School of Electrical and Data Engineering

Faculty of Engineering and IT

University of Technology Sydney

December 22, 2025

Certificate of Original Authorship

I, Huadong Tang, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosoph, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220.

Signature:	Production Note: Signature removed prior to publication.
Date:	December 22, 2025

Abstract

Semantic image segmentation is crucial for contemporary computer vision applications. It aims to classify each pixel into a specific category or class. Despite significant advancements in semantic segmentation, current methods still face challenges, including low efficiency and limited capture of contextual dependencies due to structural limitations. This research primarily focuses on improving the semantic segmentation algorithms from three different aspects.

Class-Aware Contextual Information: Leveraging contextual dependencies is a commonly used technique to enhance the performance of image segmentation. However, existing solutions do not effectively catch the class-level association between the pixels along the boundary across the objects of the different classes, but focus more on the local pixel-to-pixel relation. In this thesis, a Class-Aware Affinity module (CAA) is proposed that considers both pixel-to-pixel relation and pixel-to-class association.

Extended Context-Aware Classifier: The vanilla classifier captures global information from the training data, encoded through a fixed set of parameters, including weights and biases. However, each image has a different class distribution, which prevents the classifier from addressing the unique characteristics of individual images. At the dataset level, class imbalance leads to segmentation results being biased towards majority classes, limiting the model's effectiveness in identifying and segmenting minority class regions. In this research, we propose an Extended Context-Aware Classifier (ECAC) that dynamically adjusts the classifier using global (dataset-level) and local (image-level) contextual information.

Open-Vocabulary Semantic Segmentation: Open-vocabulary semantic segmentation relies on precise pixel-level alignment of visual and textual representations, using text as a universal reference to bridge visual disparities across diverse datasets. While prior work has primarily focused on improving visual representations or alignment models, the pivotal role of textual

representations has often been neglected. This research proposes a novel approach that harnesses large language models (LLMs) to produce enriched text prompts, replacing rudimentary templates with semantically detailed descriptions.

Acknowledgements

I express my heartfelt gratitude to my supervisor, Dr. Qiang Wu, and co-supervisor, Dr. Min Xu, for their exceptional guidance, expertise, and unwavering support throughout my doctoral journey. Their mentorship has been instrumental in shaping my research and inspiring me to pursue excellence. I am particularly grateful for their steadfast encouragement and insightful feedback, which helped me navigate the challenges of my PhD, especially during the difficult times of the COVID-19 pandemic. Their dedication not only refined my work but also provided invaluable support, motivating me to overcome obstacles and grow as a researcher.

I also extend my sincere appreciation to Dr. Youpeng Zhao and Dr. Yingying Jiang for their invaluable collaboration and support. Their willingness to engage in discussions, share knowledge, and work together on various aspects of our research fostered a dynamic and inspiring academic environment. Their contributions significantly enriched my doctoral experience and growth.

Finally, I am deeply thankful to my parents for their boundless love, encouragement, and unwavering belief in me throughout this journey. Their constant support and sacrifices provided me with the strength and motivation to persevere through the demands of my doctoral studies. Their faith in my abilities has been a cornerstone of my success, and I am forever grateful for their presence in my life.

Huadong Tang

December 22, 2025

Sydney, Australia

Publications

Publications Related to This Thesis

Journal Papers

1. Hudong Tang, Youpeng Zhao, Chaofan Du, Min Xu, and Qiang Wu, "Caa: Class-aware affinity calculation add-on for semantic segmentation," in Knowledge-Based Systems(KBS),2024.

2. Hudong Tang, Youpeng Zhao, Min Xu, Jun Wang and Qiang Wu, "Classifier Enhancement Using Extended Context and Domain Experts for Semantic Segmentation", IEEE Transactions on Multimedia, 2025.

Conference Papers

1. Huadong Tang, Youpeng Zhao, Yingying Jiang, Zhuoxin Gan, and Qiang Wu, "Class-aware contextual information for semantic segmentation," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023.

Other Publications

1. Hudong Tang, Youpeng Zhao, Yan Huang, Min Xu, Jun Wang and Qiang Wu, "LSMSeg: Unleashing the Power of Large-Scale Models for Open-Vocabulary Semantic Segmentation", Under Review, 2025.

2. Youpeng Zhao, Hudong Tang, Yingying Jiang, A Yong, Qiang Wu, and Jun Wang, "Parameter-efficient vision transformer with linear attention," in IEEE International Conference on Image Processing (ICIP), 2023.

3. Youpeng Zhao, Ming. Lin, Hudong Tang, Qiang Wu, and Jun Wang, "Merino: Entropy-driven design for generative language models on IOT devices," Proceedings of the 39th AAAl Conference on Artificial Intelligence (AAAI-25).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Image semantic segmentation is a fundamental task in computer vision that involves assigning a class label to each pixel in an image (see Figure 1.1), thereby partitioning the image into meaningful regions corresponding to objects or scenes. In contrast to image classification, which assigns a single label to an entire image, or object detection that identifies and localizes objects with bounding boxes, semantic segmentation offers pixel-level understanding, making it critical for applications requiring fine-grained scene interpretation, such as autonomous driving, medical imaging, and augmented reality. The development of semantic segmentation has been closely tied to advancements in machine learning and deep learning. Early approaches relied on hand-crafted features and traditional machine learning algorithms, such as conditional random fields (CRFs) [2] and Markov random fields (MRFs) [3], to model spatial relationships between pixels. However, these methods struggled with complex scenes due to their limited expressive power and reliance on manual feature engineering. The emergence of deep convolutional neural networks (CNNs) marked a significant turning point, with architectures like the Fully Convolutional Network (FCN) [4] introduced by Long et al. (2015) redefining the field. FCNs replaced fully connected layers with convolutional layers, enabling end-to-end training and dense prediction, which greatly improved segmentation accuracy.

Subsequent research has focused on enhancing CNN-based models to address challenges such as capturing multi-scale contextual information, improving boundary delineation, and handling class imbalance. Notable architectures, including U-Net [5] for medical imaging, DeepLab [6]

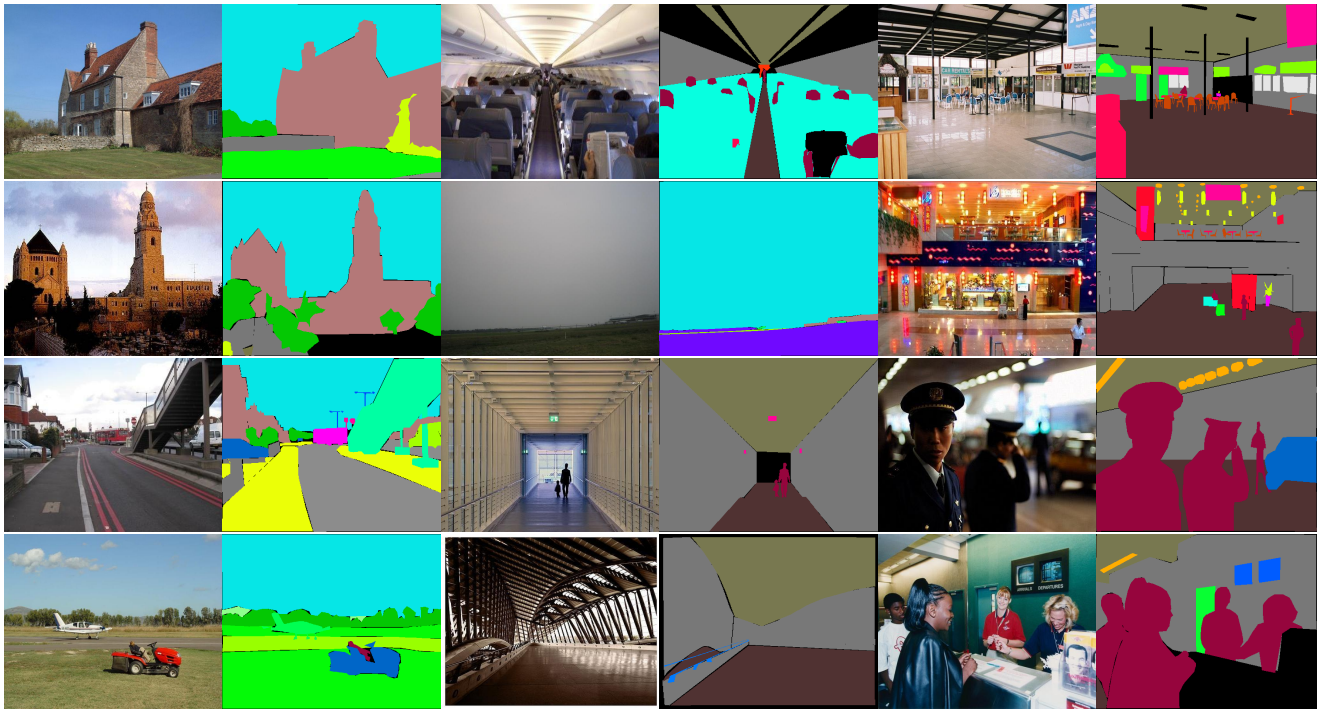


Figure 1.1: **Visualization of semantic segmentation predictions on the ADE20K [1] dataset.**

with atrous convolutions, and SegNet [7] with encoder-decoder structures, have pushed the boundaries of performance. Recent advancements incorporate attention mechanisms (e.g., Vision Transformers [8]) and self-supervised learning to further improve robustness and generalization. Additionally, real-time semantic segmentation has gained traction for resource-constrained environments, with models like EfficientNet [9] and MobileNet [10] optimizing computational efficiency without sacrificing accuracy.

The significance of semantic segmentation extends beyond academic research, as it underpins critical real-world applications. In autonomous driving, pixel-level understanding of road scenes enables precise navigation and obstacle avoidance. In medical imaging, semantic segmentation facilitates automated diagnosis by delineating anatomical structures or pathological regions. Furthermore, its role in robotics, agriculture, and urban planning highlights its versatility and societal impact. Despite these advances, challenges remain, including generalization to unseen domains, handling fine-grained details in high-resolution images, and reducing annotation costs for training data.

Semantic segmentation can be broadly categorized into closed-set and open-set paradigms, which differ in their assumptions about the data encountered during training and inference.

In closed-set semantic segmentation, the model is trained and evaluated on a fixed group of predefined categories, assuming that all test data belongs to these known categories. In contrast, open-set semantic segmentation addresses scenarios where the model must handle previously unseen classes during inference, requiring robust generalization and the ability to distinguish between known and unknown regions. These paradigms present distinct challenges, such as balancing specificity in closed-set settings and adaptability in open-set scenarios, which this thesis aims to explore in depth.

This thesis will investigate image semantic segmentation through the two distinct paradigms: closed-set and open-set segmentation.

1.2 Key Challenges in Image Semantic Segmentation

This section effectively identifies key challenges in image semantic segmentation, focusing on limitations in contextual modeling for closed-set segmentation and inadequate textual representations for open-vocabulary semantic segmentation(OVSS). It highlights the need for improved pixel-to-class associations, robust classifiers for diverse data, and enriched text prompts for OVSS.

1.2.1 Limitations of Closed-Set Image Semantic Segmentation

Closed-set image semantic segmentation, where models are trained to recognize a predefined group of classes, faces significant challenges in capturing and leveraging contextual information effectively. These limitations can be categorized into two primary aspects: image-level contextual information and dataset-level contextual information.

- **Image-level contextual information.** Since the existence of co-occurrent visual patterns [11]–[13], a line of research focuses on modeling context. *So what is the context or reliable context?* The pixel-to-pixel (spatial) relation represents whether two pixels belong to the same class or not, while the pixel-to-class (class-level) association indicates the probability of a pixel belonging to a specific class. However, existing methods only focus on one part. Therefore, we propose that reliable contexts can describe the pixel-to-pixel relation and pixel-to-class association.

- **Dataset-level contextual information.** In the decoding stage, the model restores

the spatial information and applies the class label on each pixel to get the segmentation results. An essential component in the decoder is the classifier that ultimately assigns a label to each pixel. Existing methods generally utilize the classic vanilla classifier to learn a set of fixed parameters from the training data. This results in an inherent challenge when facing highly diverse data (*e.g.* different objects, scenes, or conditions) during training. Additionally, the vanilla classifier is particularly sensitive to class imbalance, where the model tends to prioritize majority classes while neglecting minority classes. To solve the aforementioned problems, we propose an Extended Context-Aware classifier (ECAC) that embeds global (dataset-level) and local (image-level) contextual information and a calibration stage to mitigate the class imbalance issue.

1.2.2 Challenges of Open-Vocabulary Image Semantic Segmentation

Existing strategies to improve alignment fall into three main categories: (1) Refine region-level visual-text alignment; (2) Refine pixel-level visual-text alignment. However, we argue that the quality of textual representations is equally important to achieving precise visual-text alignment in OVSS. Simplistic prompts fall short in three key aspects: First, they lack the detailed semantic information required for fine-grained segmentation tasks, such as differentiating a flower species based on its intricate petal structure and color. Second, the discriminative power of generated text embeddings depends on the CLIP text encoder, which may fail to distinguish between meanings if there are lexical ambiguities. For instance, the word '`bat`' could refer to either '`a flying mammal`' or '`a piece of sports equipment used in baseball`,' and simply encoding the class name will not be enough to differentiate between these two concepts. Third, they fail to leverage multi-modal information, which is crucial for capturing the nuances of complex categories, thereby hindering the model's adaptability to diverse and fine-grained visual contexts.

1.3 Research Problems

Three key challenges stated in Section 1.2 motivate the development of novel methods to address several key research problems as follows.

- **Research Problem 1 - Inadequate Capture of Contextual Dependencies in Closed-Set Semantic Segmentation**: Existing methods prioritize pixel-to-pixel rela-

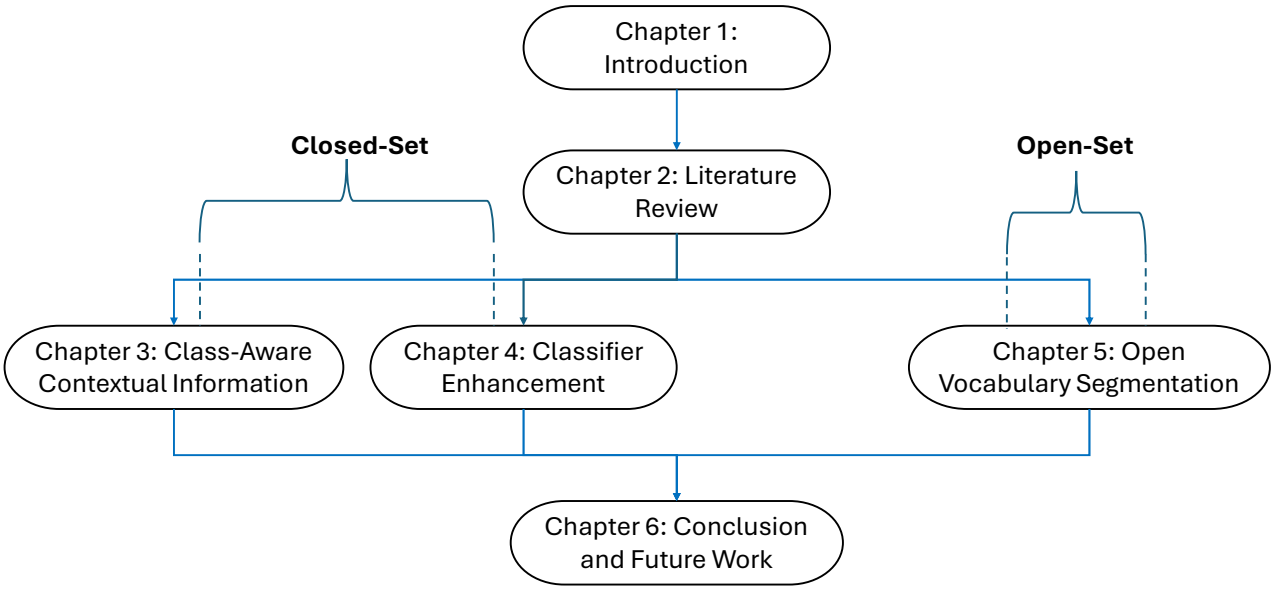


Figure 1.2: **Thesis structure. The diagram outlines how the thesis is organized, showing the closed-set studies in Chapters 3 and 4 and the open-vocabulary study in Chapter 5, following the introduction and literature review and leading to the conclusion in Chapter 6.**

tions but neglect pixel-to-class associations, limiting the ability to model reliable contextual dependencies. This results in poor intra-class compactness and inter-class dispersion, particularly along object boundaries.

- **Research Problem 2 - Class Imbalance and Data Diversity in Closed-Set Semantic Segmentation**: Vanilla classifiers with fixed parameters struggle to handle diverse data and are biased toward majority classes due to class imbalance, reducing accuracy for minority classes.

- **Research Problem 3 - Limited Textual Representations in Open-Vocabulary Semantic Segmentation**: Simplistic text prompts lack semantic detail, struggle with lexical ambiguities, and fail to leverage multi-modal information, hindering precise visual-text alignment for fine-grained segmentation.

1.4 Thesis Contribution

The main contributions of this thesis are summarized as follows:

- **Address Research Problem1**: This thesis proposes a Class-Aware Affinity (CAA)

module that alleviates the problem of insufficient intra-class compactness and weak inter-class separation. CAA jointly models pixel–pixel affinities and pixel–class relations, enabling stronger class-discriminative representations. Owing to its plug-and-play design, CAA can be seamlessly integrated into existing segmentation architectures and consistently boosts their performance. Specifically, an affinity map is introduced to model pixel-to-pixel relationships and to construct both intra-class and inter-class representations. A class-center mechanism is further proposed to capture pixel-to-class associations, enabling more effective contextual reasoning. Comprehensive experiments illustrate that CAA can achieve state-of-the-art performance across several datasets, including ADE20K, COCO-Stuff10K, and Pascal-Context.

- **Address Research Problem2**: This thesis proposes an Extended Context-Aware Classifier (ECAC) that addresses this issue by embedding both global (dataset-level) and local (image-level) contextual information, along with a calibration stage, to enhance robustness and fairness in pixel-wise classification. By embedding both global (dataset-level) and local (image-level) contextual information, ECAC enhances pixel-wise classification precision across diverse datasets. We develop a novel framework integrating a dynamically updated memory bank with a teacher-student network paradigm. The memory bank preserves dataset-level class representations, while the teacher-student mechanism, augmented by a calibration stage, refines contextual understanding and mitigates biases, achieving robust performance even for underrepresented classes. Meanwhile, we establish ECAC as a lightweight, plug-and-play module compatible with existing segmentation architectures. Comprehensive experiments on benchmarks such as ADE20K, COCO-Stuff10K, and Pascal-Context demonstrate its superior performance and adaptability, significantly advancing state-of-the-art segmentation capabilities with minimal computational overhead.

- **Address Research Problem3**: This thesis proposes LSMSeg, a pioneering framework that leverages large language models (LLMs) to create detailed, attribute-enriched text prompts, significantly improving text-visual alignment for OVSS. A feature refinement module is proposed by utilizing the precise spatial information of SAM with a category filtering module to reduce computational cost. Extensive experiments across multiple benchmarks demonstrate that LSMSeg achieves state-of-the-art performance in open-vocabulary semantic segmentation.

1.5 Thesis Structure

This thesis is organized into six chapters, including three technical chapters, each of which addresses specific challenges and research problems in image semantic segmentation across closed-set and open-vocabulary paradigms. The structure is illustrated in Figure 1.2, with each chapter contributing to the overarching goal of advancing segmentation performance through novel contextual modeling and textual representation techniques. Below is a detailed overview of each chapter's content, highlighting the key challenges addressed and the research problems solved.

- **Chapter 2**: **Literature Review.** This chapter presents a comprehensive overview of the development of image semantic segmentation, including both conventional fully supervised approaches and recent advances in open-vocabulary segmentation. It summarizes key techniques, challenges, and trends that have shaped the field.

- **Chapter3**: **Class-Aware Contextual Information for Closed-Set Semantic Segmentation.** This chapter tackles the challenge of inadequate contextual dependency modeling in closed-set semantic segmentation, where existing methods focus on pixel-to-pixel relations but neglect pixel-to-class associations. The research problem addressed is the limited intra-class compactness and inter-class dispersion, particularly along object boundaries. The proposed Class-Aware Affinity module (CAA) integrates both pixel-to-pixel and pixel-to-class associations using an affinity map and class centers, enhancing segmentation accuracy. The chapter details the CAA's design, its plug-and-play integration into existing frameworks, and experimental results on datasets like ADE20K, COCO-Stuff10K, and Pascal-Context, demonstrating state-of-the-art performance.

- **Chapter4**: **Classifier Enhancement Using Extended Context for Closed-Set Semantic Segmentation.** This chapter addresses the challenge of class imbalance and data diversity in closed-set segmentation, where vanilla classifiers with fixed parameters bias results toward majority classes and struggle with diverse scenes. The research problem is the inability of traditional classifiers to adapt to varying class distributions and minority classes. The proposed Extended Context-Aware Classifier (ECAC) embeds global (dataset-level) and local (image-level) contextual information through a memory bank and a teacher-student network with a calibration stage. The chapter describes ECAC's lightweight design, its compatibility with existing architectures, and its superior

performance on benchmarks like ADE20K and Pascal-Context, highlighting improved robustness for underrepresented classes.

- **Chapter5**: **Open-Vocabulary Image Semantic Segmentation.** This chapter focuses on the challenge of limited textual representations in open-vocabulary semantic segmentation, where simplistic prompts lack semantic detail, struggle with lexical ambiguities, and fail to leverage multi-modal information. The research problem addressed is the poor visual-text alignment for fine-grained segmentation tasks. The proposed LSMSeg framework leverages large language models (LLMs) to generate enriched text prompts with diverse visual attributes, optimizing text-visual alignment. The chapter outlines LSMSeg's methodology, attribute selection process, and experimental validation across diverse benchmarks, showcasing state-of-the-art performance under open-vocabulary settings.
- **Chapter6**: **Conclusion and Future Work.** The final chapter concisely summarizes the thesis content and highlights its key contributions.

Chapter 2

Literature Review

This chapter provides a comprehensive literature review of recent advances in semantic segmentation, focusing on methods that address these challenges through improved contextual modeling. Specifically, we examine works that explore three key dimensions of contextual information: (1) *image-level context*, which captures spatial and semantic relationships among pixels within an individual image; (2) *dataset-level context*, which leverages global statistical priors and class distribution knowledge across the training corpus; and (3) *open-vocabulary semantic segmentation*, which aims to generalize beyond a fixed set of categories using vision-language models and textual guidance.

The chapter is organized as follows: Section 2.1 reviews techniques that enhance segmentation accuracy by modeling image-level contextual information such as pixel affinities, attention mechanisms, and relational reasoning. Section 2.2 discusses methods that utilize dataset-level knowledge to mitigate issues like class imbalance and label noise, including memory-based approaches and global feature regularization. Section 2.3 surveys the emerging line of research on open-vocabulary semantic segmentation, highlighting the integration of large-scale vision-language models and the generation of descriptive prompts.

By analyzing these three perspectives, this chapter aims to position the contributions of this thesis within the broader research landscape and identify gaps that motivate our proposed solutions.

2.1 Image-Level Contextual Information for Image Semantic Segmentation

Image-level contextual information integrates global and local cues to enhance semantic segmentation. Global context captures overall scene characteristics, like scene categories and object relationships, while local context focuses on specific details, such as textures and edges. By leveraging techniques like image classification, attention mechanisms, and feature aggregation, it provides a comprehensive understanding of the image's semantic and structural content, improving segmentation accuracy and coherence.

This section details the previous methods of image-level contextual information for image semantic segmentation. The first subsection introduces the multi-scale context for image semantic segmentation. However, the only correlation between pixels is the overlap of receptive fields, ignoring the condition that different pixels need different contextual dependencies. The attention-based method tackles this problem by using the attention mechanism. The third subsection introduces the class-aware context for image semantic segmentation, which solves the problem that attention-based methods do not explicitly model the dependencies of pixels among classes. Our thesis is motivated by the class-aware context and proposes a new class-aware affinity module.

2.1.1 Multi-scale context for image semantic segmentation

Multi-scale semantic segmentation methods leverage contextual information from various spatial scales to enhance feature representations for pixel-wise image classification. These approaches aggregate features extracted at different resolutions or receptive fields, capturing both fine-grained local details and broader global contexts to improve segmentation accuracy across diverse object sizes and scene complexities.

The Fully Convolutional Networks (FCNs) [4] is an epoch-making work to promote the advancement of semantic segmentation. Based on this method, aggregating contextual information to enhance feature representation is a common practice. Many recent works [14]–[16] have been proposed to extract discriminative features by combining contextual features. Figure 2.1 provides a typical example of multi-scale context aggregation modules widely used in semantic segmentation. The feature extractor first produces a shared feature map, which is then pro-

cessed by parallel convolutional branches with different kernel sizes or dilation rates. Each branch captures contextual information at a distinct spatial scale, while the global pooling branch provides holistic scene-level cues. By concatenating and fusing these multi-scale representations, the model obtains a more robust and scale-invariant feature embedding, which effectively addresses object-size variation and enhances pixel-level classification. Specifically, Deeplab methods [6], [17], [18] propose atrous spatial pyramid pooling (ASPP) to capture more contexts from multiple scales, while PSPNet [19] employs a pyramid parsing module for global context exploration, achieved through context aggregation across diverse regions. CCL [20] introduces a new approach for scene segmentation, which improves performance by incorporating contextual information and multi-scale features. Some works [20]–[22] aim to capture more comprehensive global context information by extending kernel size or proposing an efficient encoding layer. For instance, ACNet [22] captures pixel-aware context by integrating global and local contexts regarding different pixel requirements. EncNet [21] proposes to obtain a comprehensive global context and selectively emphasize context relevant to specific classes. The advantages of multi-scale methods include improved feature robustness, as they combine

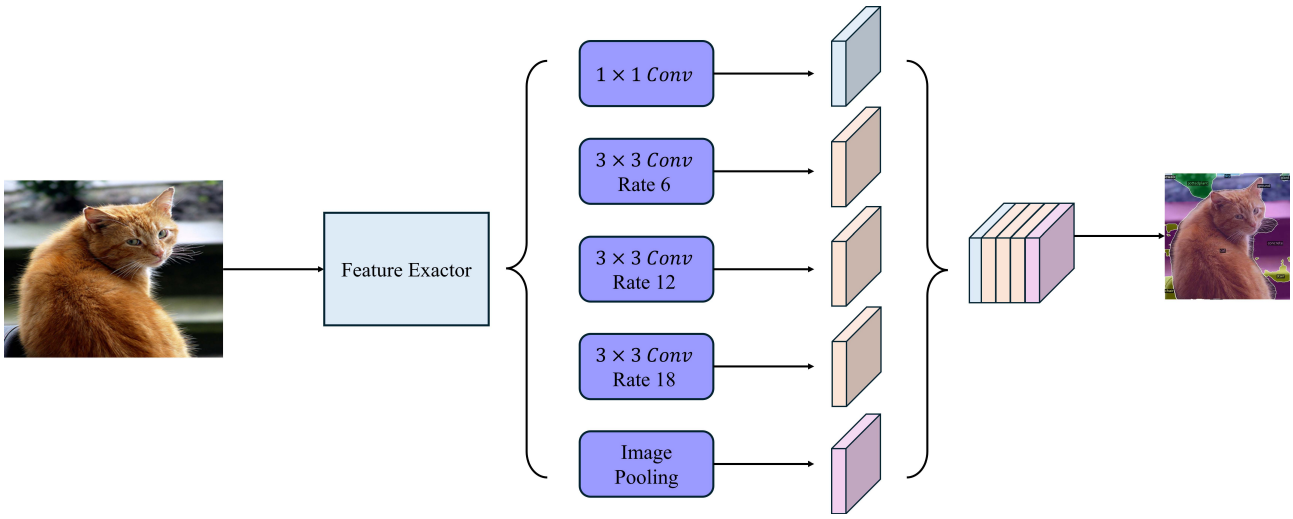


Figure 2.1: **Example of Multi-Scale Context. The feature map is processed by parallel convolution branches with different dilation rates and a global pooling branch, enabling the model to capture local details and large-scale context for handling objects of various sizes.**

local and global cues to better distinguish objects of varying scales, and enhanced contextual understanding, which aids in interpreting complex scenes. However, these methods face challenges such as increased computational complexity due to processing multiple scales, potential

over-smoothing of fine details when prioritizing global context, and difficulty in modeling pixel-specific dependencies with uniform feature aggregation. Additionally, these methods aggregate contextual information with multi-scale local features and cannot capture different dependencies for different pixels. Recently, attention-based models have shown considerable performance and emerged as a widely adopted approach for semantic segmentation.

2.1.2 Attention-based context for image semantic segmentation

The attention mechanism, originally designed for Natural Language Processing (NLP) tasks [23], [24], is a technique that enables models to focus on relevant information while suppressing irrelevant or extraneous data. In semantic segmentation, attention mechanisms are employed to model long-range contextual dependencies, assigning varying importance to different spatial regions or features to enhance pixel-wise classification. Unlike traditional convolutional operations, which stack layers to enlarge receptive fields but are limited to local contextual information and computationally intensive, attention mechanisms efficiently model global context by dynamically weighting relationships between pixels across the entire image. As shown in Figure 2.2, the self-attention module first projects the input features into query, key, and value embeddings. The similarity between the query and key features is used to construct an attention map through softmax normalization, which highlights long-range pixel interactions. This attention map then reweighs the value features to generate context-enhanced representations for semantic segmentation. Building on the attention mechanism, several methods have advanced semantic segmentation. DANet [25] and RCANet [26] propose a two-attention module network to aggregate long-range spatial information. CCNet [13] and SPNet [11] come up with a criss-cross attention module and strip pooling module to capture long-range dependencies while reducing computational complexity. Besides, SANet [27] proposes a new squeeze-and-attention network, which takes into account dense predictions at multiple scales for individual pixels, as well as spatial attention for clusters of pixels. Attention-based methods offer several advantages in semantic segmentation. First, they effectively model long-range dependencies, enabling the model to understand global contextual relationships critical for complex scenes. Second, they improve computational efficiency compared to stacked convolutions by selectively focusing on relevant features. Third, attention mechanisms enhance feature discriminability by emphasizing contextually important regions, leading to more accurate segmentation. However, these methods have limitations. They often require significant memory resources to compute

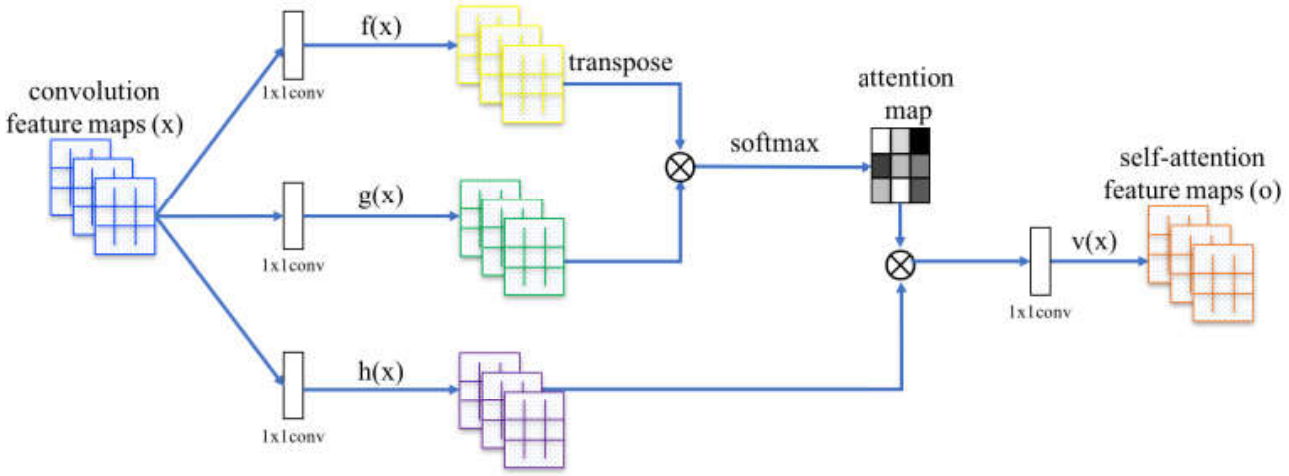


Figure 2.2: **Illustration of Attention mechanism. Adapted from [28]. The input feature map is projected into query $f(x)$, key g(x), and value $h(x)$ branches to compute pixel-wise affinities and generate the self-attention feature map.**

attention maps for high-resolution images, increasing computational overhead. Additionally, attention mechanisms may not explicitly model inter-class pixel dependencies, potentially limiting their ability to differentiate between semantically similar classes. Finally, designing effective attention modules can be complex, requiring careful tuning to balance performance and efficiency.

2.1.3 Class-aware context for image semantic segmentation

Class-aware context methods in semantic segmentation focus on modeling contextual relationships specific to individual semantic classes to enhance pixel-wise classification. These approaches leverage class-specific features, such as intra-class or inter-class relationships, to capture holistic contextual information, enabling pixels to better discern their semantic category within complex scenes. By emphasizing class-level dependencies, these methods enhance the model's ability to differentiate objects belonging to distinct classes.

Several works have advanced class-aware context modeling in semantic segmentation. ACFNet [29] introduces the concept of class centers, which capture holistic context for each class, enabling pixels to distinguish different classes within the entire scene. As shown in Figure 2.3, OCRNet generates soft object regions from pixel features and aggregates them into object-region representations. The pixel–region relations are then computed to measure how strongly each pixel is associated with each object region. These relations are finally used to refine pixel features,

producing class-aware augmented representations that enhance intra-class consistency. However, these methods [29]–[31] primarily focus on intra-class centers, often overlooking inter-class relationships. More recently, CPNet [32] explores affinity-aware context to model pixel-to-pixel relationships, indicating whether two pixels belong to the same class. Despite this advancement, CPNet does not fully differentiate pixels across different classes, limiting its ability to capture comprehensive inter-class dependencies. Class-aware context methods offer several advantages.

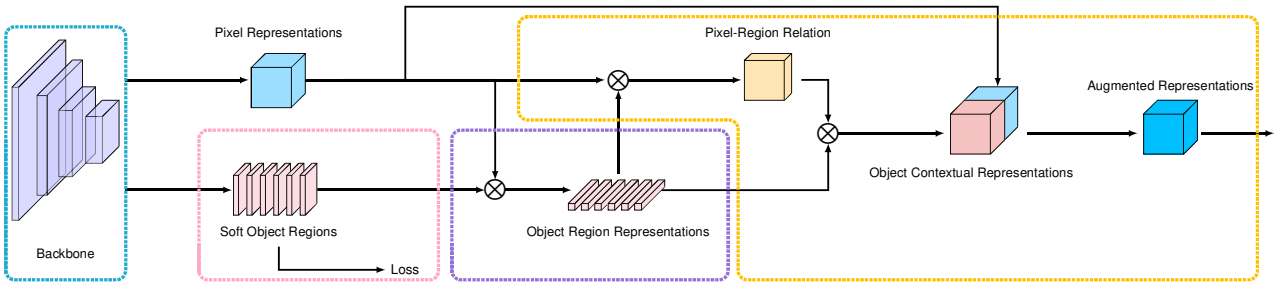


Figure 2.3: **Illustration of OCRNet. Adapted from [30]. The model first generates region representations from pixel features and computes pixel–region relations, which are then used to refine pixel features with class-aware contextual information.**

First, they enhance intra-class consistency by aggregating features specific to each class, improving segmentation accuracy for objects with similar appearances. Second, they provide robust contextual cues, allowing pixels to leverage class-specific global information for better scene understanding. Third, these methods can model complex object relationships, which is particularly useful in scenes with multiple interacting classes. However, class-aware methods have limitations. They often focus on intra-class relationships while neglecting inter-class dependencies, potentially leading to confusion between semantically similar classes. Additionally, extracting class-specific contexts can be computationally expensive, requiring sophisticated modules to model pixel-to-class relationships. In this thesis, a class-aware affinity module is proposed to help existing segmentation frameworks improve performance by integrating pixel-to-pixel relations and pixel-to-class associations.

2.2 Dataset-Level Contextual Information for Image Semantic Segmentation

Dataset-level contextual information enhances image semantic segmentation by leveraging patterns and relationships across an entire dataset, rather than focusing solely on individual images.

It captures dataset-wide characteristics, such as category distributions, co-occurrence patterns, and semantic correlations among classes, to improve segmentation performance. This approach can also mitigate class imbalance issues to some extent by better representing underrepresented categories through dataset-wide patterns.

This section details the previous methods of dataset-level contextual information for image semantic segmentation. The first subsection introduces contrastive learning for image semantic segmentation. However, this approach does not directly integrate dataset-level information. The dataset-based category representations can utilize the dataset to capture rich contextual information, improving the segmentation performance. The third subsection introduces the class imbalance for image semantic segmentation, which solves the problem that previous dataset-level representations do not explicitly mitigate the class imbalance issue. Our thesis is motivated by the class imbalance methods and proposes an extended context-aware classifier.

2.2.1 Contrastive learning for image semantic segmentation

Contrastive learning in semantic segmentation is a technique that enhances feature representations by using a contrastive loss to pull features of similar pixels closer together and push features of dissimilar pixels apart in the feature space. This approach serves as a supervisory signal that encourages the model to learn discriminative features without directly integrating cross-image contextual information during feature decoding.

Recent works have leveraged contrastive learning to incorporate cross-image contextual information for semantic segmentation across various supervision paradigms [33]–[36]. For instance, in weakly supervised learning, methods like [33] use contrastive loss to mine cross-image relationships with limited annotations. In semi-supervised learning, approaches such as [34] exploit unlabeled data to enhance feature representations. For fully supervised learning, methods like [35], [36] utilize memory banks to store and compare features across images, improving contextual understanding. Notably, CIPC [35] proposes a supervised, pixel-wise contrastive learning approach that shifts the traditional image-level training strategy to an inter-image and pixel-to-pixel paradigm. As shown in Figure 2.4, pixel embeddings from multiple images are extracted and stored in a memory bank. For each pixel, positive samples are selected from pixels of the same class across images, while negatives are drawn from different classes. These sampled pairs are then used to optimize a contrastive loss, encouraging intra-class compactness

and inter-class separability. By leveraging pixel-wise or region-wise comparisons across images,

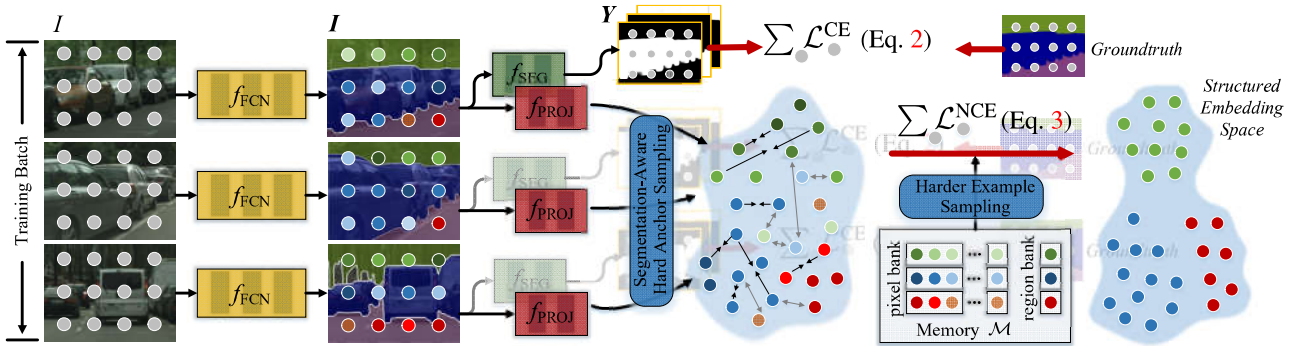


Figure 2.4: **Illustration of Contrastive Learning for Semantic Segmentation. Adapted from [35]. Pixel features are compared across images through memory-bank sampling to form positive and negative pairs, enabling pixel-wise contrastive supervision.**

contrastive learning improves the model's ability to generalize and capture robust semantic patterns for pixel-wise classification. However, this approach implicitly models the cross-image information. And, selecting appropriate positive and negative pairs is challenging, as poor choices can lead to suboptimal feature learning.

2.2.2 Dataset-level category representations for image semantic segmentation

Dataset-level category representations in semantic segmentation involve aggregating contextual information across multiple images to enhance pixel-wise feature representations for specific semantic classes. These methods leverage cross-image relationships to build robust category-specific features, enabling the model to capture consistent patterns for each class across the dataset. By integrating information from diverse images, these approaches aim to improve the generalization and discriminative power of pixel representations for accurate segmentation.

Recent methods, such as MCIBI and MCIBI++ [37], [38], have advanced dataset-level category representations by aggregating cross-image contextual information into original pixel representations to enhance their capability. As illustrated in Figure 2.5, pixel features are first projected and associated with dataset-level memory prototypes, which store aggregated category representations across images. The model computes relations between pixel embeddings and memory entries to generate category-aware representations, which are then fused back into the original pixel features to provide class-consistent contextual enhancement. These approaches utilize

self-attention mechanisms to model relationships across images, improving the robustness of category-specific features.

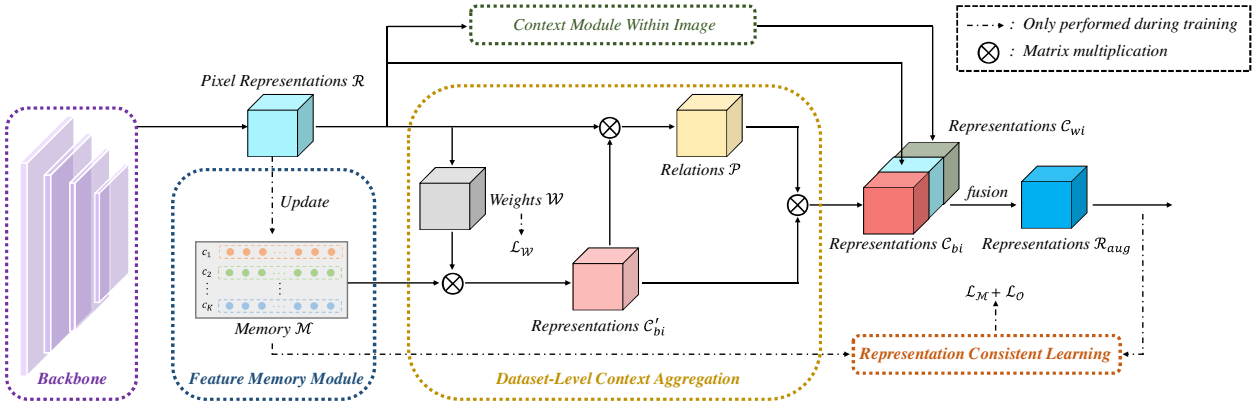


Figure 2.5: **Dataset-level category representations for semantic segmentation. Adapted from [37]. Pixel features interact with memory-stored category representations through cross-image relations, producing enhanced class-specific contextual embeddings.**

Dataset-level category representation methods offer several advantages. First, they enhance feature robustness by incorporating cross-image contextual cues, leading to better generalization across varied scenes and object appearances. Second, they improve class-specific discrimination by modeling consistent category features, which is particularly beneficial for distinguishing semantically similar classes. Third, these methods can leverage large datasets to capture rich contextual information, enhancing performance in complex segmentation tasks. However, these approaches have notable limitations. The computational complexity of aggregating cross-image information, often involving self-attention mechanisms, can be quadratic in memory and computation, making them inefficient for large datasets or high-resolution images. Additionally, despite this design, the memory bank alone cannot fully overcome the biases induced by class imbalance.

2.2.3 Class imbalance for image semantic segmentation

Class imbalance in semantic segmentation occurs when some classes, like backgrounds or common objects, are far more prevalent in a dataset than others, such as rare objects or fine details. This uneven distribution can bias models toward dominant classes, resulting in poor performance on underrepresented ones and lower overall segmentation quality.

Several methods have been proposed to address class imbalance in semantic segmentation.

DAMC [39] tackles class imbalance by introducing a margin calibration strategy that adjusts decision boundaries based on the label distribution. BLV [40] approaches the problem by injecting category-wise variation into logits during training. As shown in Figure 2.6, BLV computes the dataset's category distribution and generates class-wise variation signals, which are then added to pixel-level logits. This encourages balanced logit responses across frequent and rare classes, mitigating bias toward dominant categories. Besides, AUCSeg [41] explores AUC optimization methods in the context of pixel-level long-tail semantic segmentation. However, these methods address class imbalance implicitly by adjusting the loss function or learning dynamics, without explicitly modeling class distributions. In contrast, prototype-based methods [36], [42]–[44] learn representative embeddings for each class, directly enhancing feature discrimination, especially for tail classes. For example, SSA [44] proposes a novel semantic and spatial adaptive classifier to mitigate the class imbalance issue. Nevertheless, these methods still rely on the fixed parameters of the vanilla classifier, which often exacerbate class imbalance and degrade performance on minor categories. Addressing class imbalance offers several advantages. First,

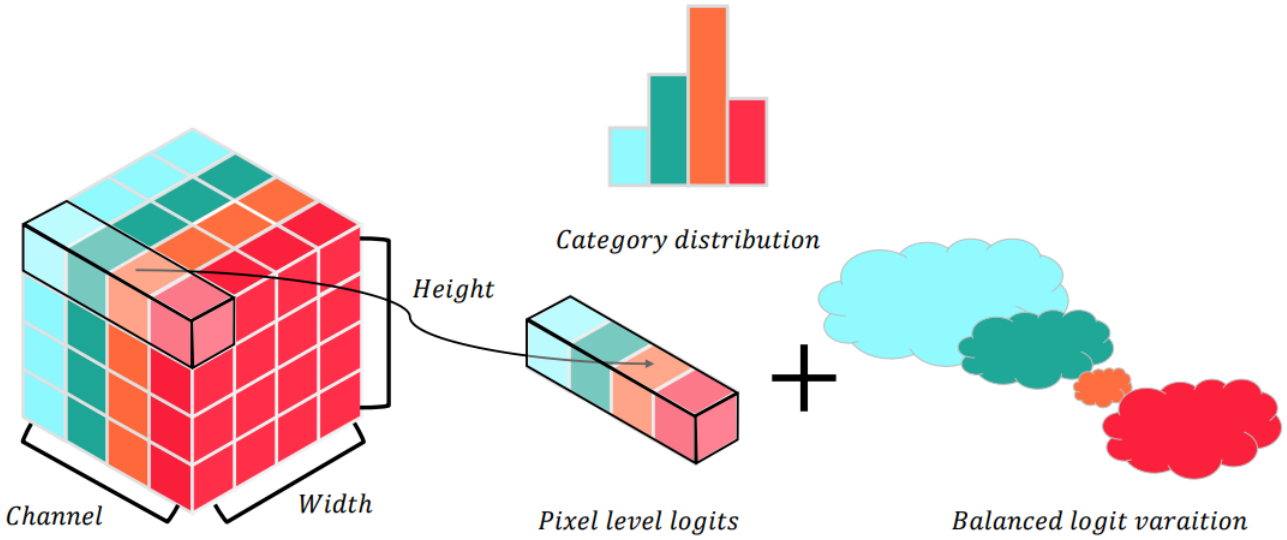


Figure 2.6: **Balanced Logit Variation (BLV). Adapted from [40]. Category distributions are used to inject class-wise logit variation into pixel-level logits during training to alleviate class imbalance.**

it improves segmentation accuracy for underrepresented classes, enhancing the model's ability to handle rare or fine-grained objects. Second, it promotes balanced feature learning, leading to more robust and generalizable models across diverse scenes. Third, mitigating class imbalance enhances the model's applicability in real-world scenarios, where minority classes are often

critical (e.g., detecting small defects or rare objects). However, tackling class imbalance has limitations. Many methods introduce additional computational complexity, such as specialized loss functions or auxiliary modules, which can increase training time and resource demands. Additionally, overemphasizing minority classes may degrade performance on majority classes, requiring careful calibration. Finally, some approaches rely on assumptions about class distributions, which may not generalize well to highly imbalanced or unseen datasets. To address the existing challenge, we propose a simple yet effective approach. It leverages a memory bank to learn dataset-level category features, capturing global distributions to alleviate class imbalance, and uses a teacher-student network to correct biases. A calibration stage further mitigates class imbalance, enhancing segmentation accuracy across all categories.

2.3 Open Vocabulary Image Semantic Segmentation

Open Vocabulary Image Semantic Segmentation enables models to segment images into semantic regions without being limited to a predefined set of classes, allowing recognition of arbitrary categories described by text or other modalities. The core of this method lies in learning visual-text alignment, which bridges image features with textual descriptions for flexible and generalized segmentation. It leverages language-guided representations to generalize across diverse and unseen categories, making it highly adaptable to real-world scenarios.

This section details the related works of open vocabulary image semantic segmentation. The first subsection introduces region-level alignment for open vocabulary image semantic segmentation. However, this method is a two-stage framework and requires significant computational resources. Pixel-level alignment refines segmentation by aligning individual pixel features with textual or semantic representations, improving segmentation accuracy for complex scenes with fine details. Nevertheless, the pivotal role of the text prompt is ignored. The third subsection introduces the enhanced text prompt methods with large language models, but most of the methods focus on image classification tasks. Motivated by this, our thesis focuses on improving the text prompts for open vocabulary semantic segmentation.

2.3.1 Region-level Alignment for Open Vocabulary Image Semantic Segmentation

Region-level alignment in open vocabulary semantic segmentation refers to techniques that align visual regions or masks with textual descriptions to enable segmentation of arbitrary classes without requiring class-specific training. These methods leverage pre-trained vision-language models, such as CLIP [45], to classify regions generated by a mask proposal network, facilitating segmentation for open vocabulary scenarios where the model must recognize diverse and potentially unseen categories based on textual prompts.

Several works have explored region-level alignment for open vocabulary semantic segmentation. Methods like [46]–[49] utilize a two-stage framework, where a class-agnostic mask generator first extracts region proposals, followed by a pre-trained CLIP model to classify each mask based on textual prompts. Specifically, as illustrated in Figure 2.7, OVSeg [47] proposes fine-tuning the pre-trained CLIP on domain-specific datasets and constructing tailored training data to improve recognition of masked regions. It first uses a class-agnostic segmentation model, such as MaskFormer, to generate region proposals. Each masked region is cropped and paired with a corresponding textual prompt, and both are encoded using CLIP to compute region–text similarity. During training, mask–category pairs are used to adapt CLIP through contrastive losses, enabling the model to classify arbitrary regions based on textual descriptions. However, such two-stage approaches are inefficient, as they rely on separate networks for mask generation and classification, lack integrated contextual information, and incur significant computational overhead due to processing multiple image crops with CLIP. Region-level alignment methods offer several advantages. First, they enable open vocabulary segmentation, allowing the model to generalize to new classes using textual descriptions, which is critical for real-world applications with dynamic or diverse categories. Second, they leverage powerful pre-trained models like CLIP, reducing the need for extensive labeled segmentation data. Third, by focusing on regions, these methods can capture spatially coherent object information, improving segmentation quality for complex scenes. However, these methods have notable restrictions. They often rely on a two-stage framework, which separates mask generation and classification, leading to inefficiencies and suboptimal performance due to the lack of end-to-end optimization. Additionally, processing multiple region proposals incurs high computational costs, especially when using large vision-language models like CLIP. Finally, these methods may struggle with accurately

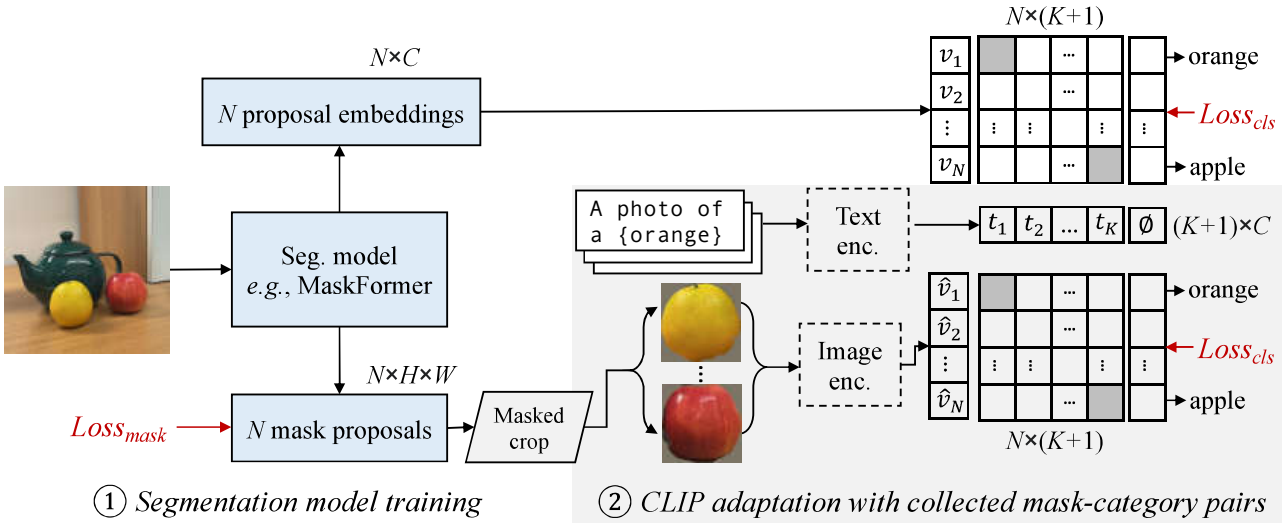


Figure 2.7: **The method of OVSeg. Adapted from [47]. Region proposals are first generated by a class-agnostic segmentation model, and CLIP is then adapted using mask–category pairs to classify each region for open-vocabulary segmentation.**

classifying masked regions or handling contextual information, particularly for background or ambiguous areas, due to domain mismatches in pre-trained models.

2.3.2 Pixel-level Alignment for Open Vocabulary Image Semantic Segmentation

Pixel-level alignment in open vocabulary semantic segmentation refers to techniques that directly align pixel-wise visual features with textual descriptions using a unified vision-language model to enable segmentation of arbitrary classes without class-specific training. Unlike two-stage methods that separate mask generation and classification, pixel-level alignment approaches integrate these processes within a single framework, leveraging pre-trained vision-language models (VLMs) like CLIP to generate segmentation masks based on textual prompts, facilitating open vocabulary scenarios with diverse or unseen categories.

Recent one-stage methods [50]–[52] directly apply a unified vision-language model for open-vocabulary segmentation. SAN [51] attaches a lightweight image encoder to the pre-trained CLIP to generate masks and attention biases. SCAN [52] proposes a semantic integration module to embed the global semantic understanding and a contextual shift strategy to achieve domain-adapted alignment. As shown in Figure 2.8, SED constructs a multimodal cost volume by correlating hierarchical visual features with text embeddings. Spatial and class aggregation

modules are then applied to refine the cost distribution, followed by skip-layer fusion to enhance pixel-wise alignment. This enables the model to produce open-vocabulary predictions in a fully end-to-end manner. Pixel-level alignment methods offer several advantages. First, they pro-

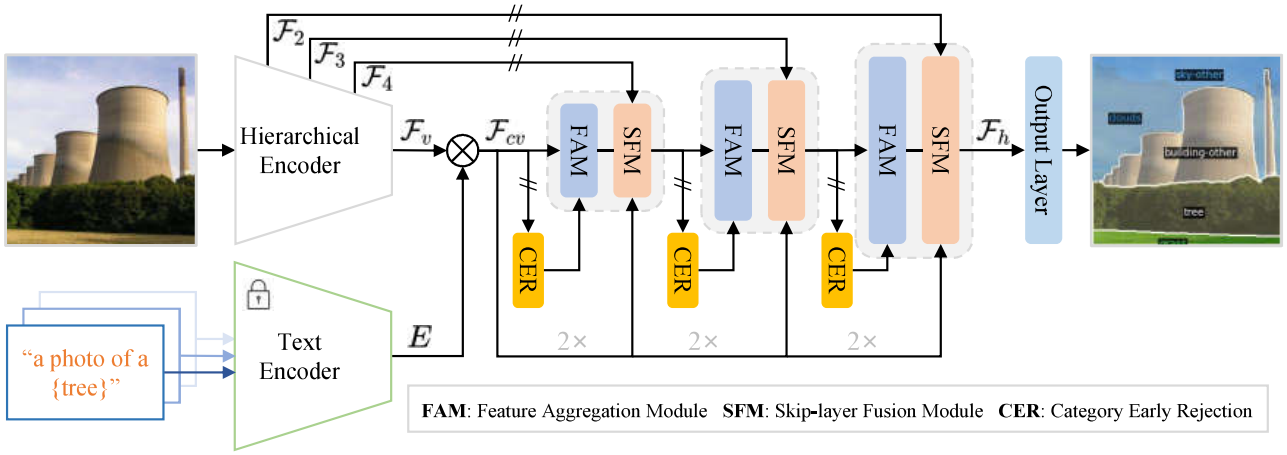


Figure 2.8: **The method of SED. Adapted from [53]. SED aggregates spatial and class information within a multimodal cost volume and fuses hierarchical visual features with text embeddings to generate pixel-level open-vocabulary predictions.**

vide end-to-end optimization, improving efficiency and performance by jointly learning mask generation and classification within a unified model. Second, they leverage the rich semantic understanding of pre-trained VLMs, reducing reliance on extensive labeled segmentation datasets. Third, these methods enable fine-grained segmentation by aligning individual pixels with textual descriptions, enhancing accuracy for complex scenes with diverse objects. However, these approaches have limitations. They often underexploit the role of language, relying heavily on static text embeddings from pre-trained VLMs without refining text attributes for segmentation tasks, which can limit performance on nuanced or domain-specific categories.

2.3.3 Text Prompt Enhancement with Large Language Models

Text prompt enhancement with large language models (LLMs) in semantic segmentation involves leveraging the natural language understanding capabilities of LLMs, such as GPT [55] or LLaMA [56], to generate or refine textual prompts that guide vision-language models (VLMs) in open vocabulary segmentation tasks. These methods enrich text prompts with class-specific or context-aware descriptions, enabling VLMs to better align visual features with semantic categories, particularly for diverse or unseen classes in complex scenes.

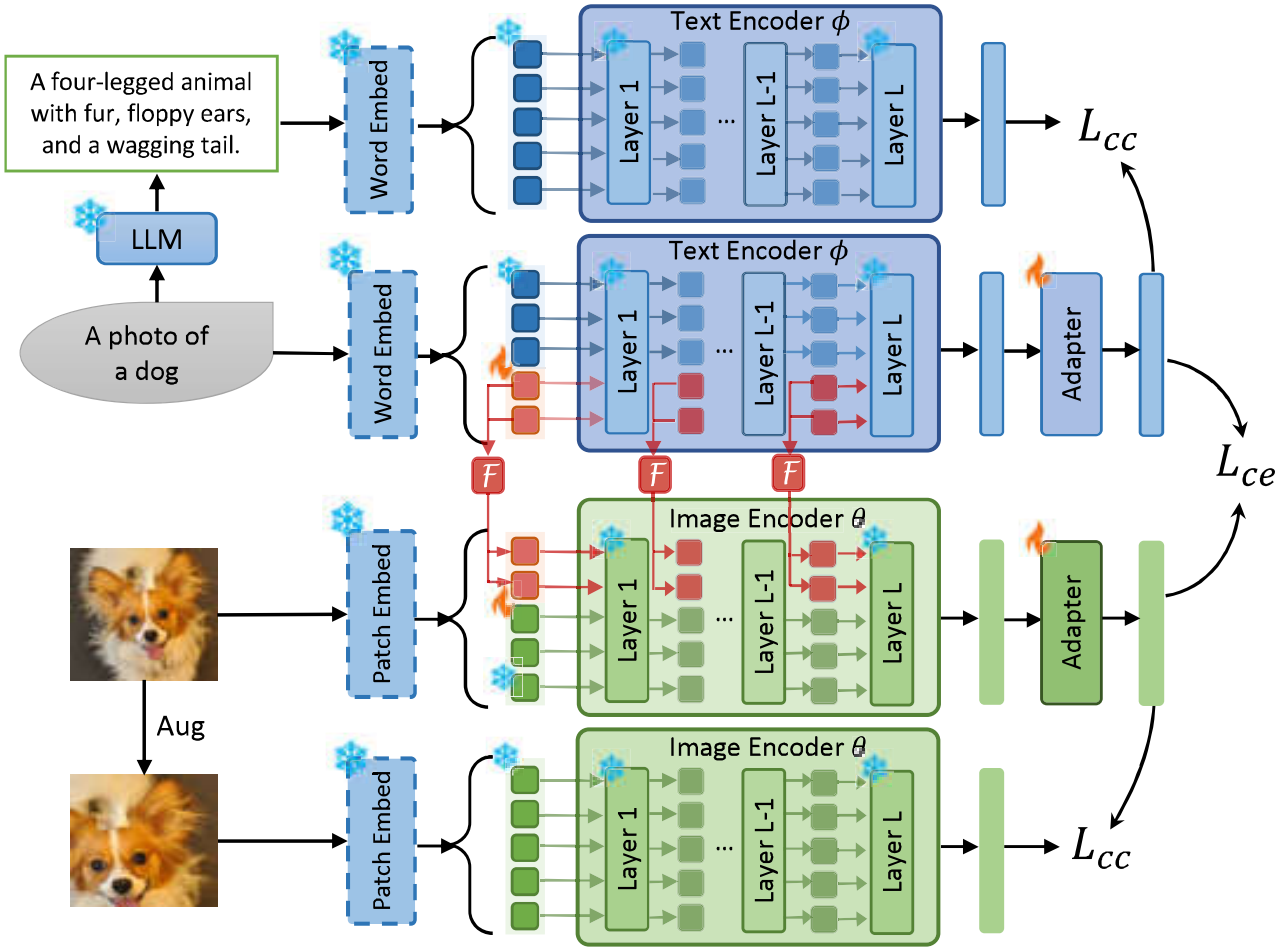


Figure 2.9: **The method of CoPrompt. Adapted from [54]. CoPrompt refines text and image encoders using LLM-generated descriptions and coherence constraints to enhance cross-modal alignment for segmentation.**

Recent works have demonstrated the potential of LLMs to enhance text prompts for semantic segmentation [54], [57]–[59]. CuPL [57] utilizes LLMs to generate class-specific prompt descriptions, which are combined through text prompt ensembling to improve VLM performance. WaffleCLIP [59] employs random descriptors augmented with data-specific concepts generated by LLMs, achieving further improvements in segmentation accuracy. As shown in Figure 2.9, CoPrompt [54] harnesses the expertise of pre-trained LLMs by applying coherence constraints to text prompts and enhancing image data, boosting generalization across diverse datasets. Specifically, LLM-generated descriptions are fed into a word encoder and text encoder to obtain refined text embeddings, while the corresponding images undergo augmentation and are processed by the image encoder. CoPrompt then enforces cross-modal coherence by aligning text and image features through adapter modules and contrastive-style losses, thereby improving performance. Additionally, approaches like LISA [60] explore LLMs for reasoning-based segmentation tasks, highlighting their capability for fine-grained visual understanding.

Text prompt enhancement offers several advantages. First, it improves the performance of VLMs by providing richer, more descriptive prompts, enhancing their ability to recognize nuanced or domain-specific categories. Second, it leverages the generalization capabilities of LLMs, reducing the dependency on manually crafted prompts or extensive labeled data. Third, it enables dynamic adaptation to various tasks, such as open vocabulary segmentation, by generating contextually relevant prompts tailored to specific datasets or classes. In this thesis, we systematically tailor LLM-generated prompts for OVSS by optimizing attribute selection and combination at the pixel level. This task-specific refinement distinguishes our approach from existing efforts, addressing the unique challenge of fine-grained visual-text alignment in segmentation.

Chapter 3

Class-Aware Contextual Information for Semantic Segmentation

This chapter addresses the limitation that existing semantic segmentation approaches often rely heavily on local pixel-to-pixel relations, lacking a broader understanding of class-level associations, particularly along object boundaries where different semantic regions interact. Such methods tend to focus on fine-grained texture or appearance details while overlooking the semantic coherence that should exist among pixels belonging to the same category, especially when they are spatially distant or disrupted by occlusion or clutter.

To overcome this issue, we propose a novel Class-Aware Affinity (CAA) module designed to enhance global contextual modeling by explicitly incorporating class-level semantic cues. The core idea is to capture the affinity between pixels conditioned on their semantic class, thereby encouraging intra-class consistency and enhancing the distinction between different classes—particularly at boundary regions where confusion is most likely to occur.

The CAA module can be seamlessly integrated into existing segmentation frameworks with minimal computational overhead. Through extensive experiments on standard benchmarks, we demonstrate that the proposed approach significantly improves segmentation performance, especially in challenging regions with ambiguous class boundaries, validating the effectiveness and generality of the proposed method.

3.1 Introduction

Building upon the foundation laid by the fully convolutional networks (FCNs) [4], numerous methods have achieved remarkable advancements. However, the FCN has a problem of low efficiency and only provides insufficient contextual dependencies for the reason of structural weaknesses. The limitation of insufficient background information greatly affects its segmentation accuracy. Given this, researchers primarily focus on two aspects to enhance segmentation performance: (i) design a better encoder structure [61]–[63]; and (ii) model reliable contextual information [19], [21], [25].

Since the existence of co-occurrent visual patterns [11]–[13], a series of works focus on modeling context. Early study is mainly about multi-scale context for semantic segmentation, which exploits dilated convolutions or pyramid pooling to obtain feature maps by aggregating multi-scale contexts. Specifically, PSPNet [19] employs pyramid spatial pooling to aggregate context. However, they only can capture local features and bring limited contextual information. Deeplab [6], [17], [18] family introduces the atrous spatial pyramid pooling (ASPP) to capture local context from different scales of the image. Some other methods utilize dot self-attention to extract long-range dependencies. Non-local network [64] first proposes to utilize a self-attention module to learn global contextual dependencies. Inspired by this, some works [11], [25], [65], [66] further utilize the attention mechanism for semantic segmentation. PSANet [67] introduces a pyramid spatial attention mechanism for capturing features at multiple scales. Meanwhile, DANet [25] introduces channel attention to cooperate with spatial attention to capture contextual information globally. [68] designs to enhance feature extraction and prediction stages by considering channel perspectives. However, these methods only focus on pixel-to-pixel dependencies. Multi-scale contexts are established within predefined regions, with the only pixel correlation being the overlap of receptive fields. They only focus on local pixel relationships, leading to the category confusion problem on a semantic level. Besides, attention-based contexts only catch the relation on the local texture level. We argue that they should consider the association between the pixel and the class context produced by the given image.

Apart from this, ACFNet [29] and OCRNet [30] introduced pixel-to-class relations by leveraging class-center representations, while ISNet [12] and CPNet [32] further explored class-region contexts. However, these methods typically focus on either pixel-to-pixel or pixel-to-class relations,

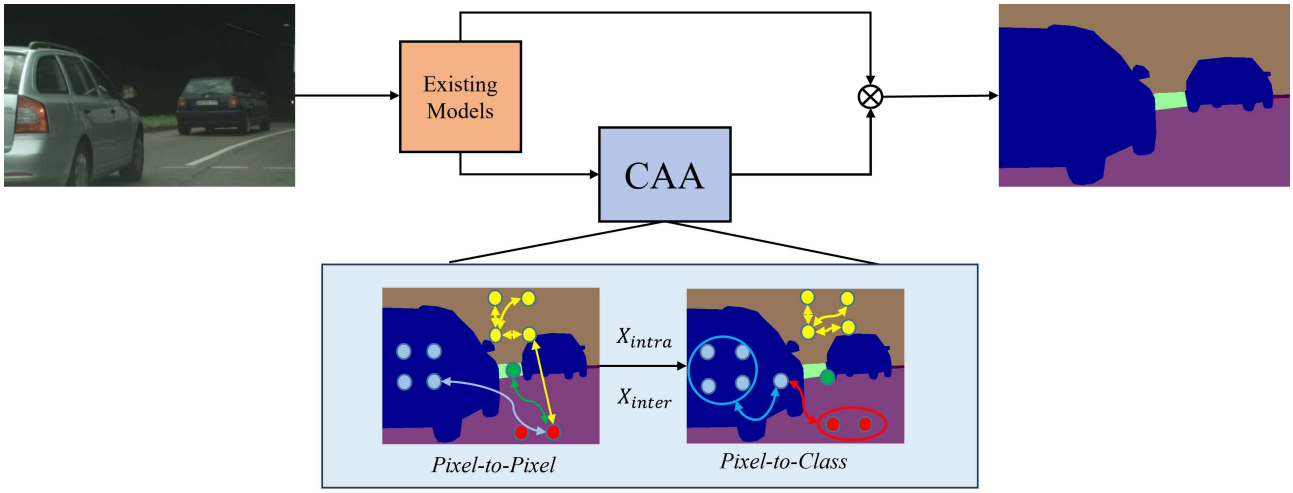


Figure 3.1: **The concept of our CAA. We first explore pixel-to-pixel relation: whether two pixels belong to the same class or not. Then, we calculate the pixel-to-class relation: the relation between the i^{th} pixel and other pixels in the j^{th} class region, *i.e.*, the blue point and the blue car region.**

not both. We argue that reliable context should integrate both *pixel-to-pixel* relations (indicating whether two pixels belong to the same class) and *pixel-to-class* associations (indicating the probability of a pixel belonging to a specific class).

To this end, we introduce a Class-Aware Affinity Module (CAA), which is regarded as an additional supplementary module to the existing mainstream segmentation framework [4], [17], [19] (see Figure 3.1). The existing mainstream architectures provide feature extraction to aggregate the spatial information. Therefore, we further utilize those features to capture pixel-to-pixel relation to enhance pixel representations and get intra- and inter-class representations X_{intra} and X_{inter}. Intra-class representations refer to the augmentation of contextual dependencies among pixels belonging to the same class, whereas inter-class representations belong to the heightened relationships among pixels of distinct classes. Every colored dot represents a pixel along with its corresponding class. We just know if two pixels belong to the same class, however, we still do not explicitly build the relationship between pixel and class region *i.e.*, the relation between blue dots ('*car*'), yellow dots ('*tunnel*'), green dots ('*terrain*') and the red dots ('*road*'). We further model the pixel-to-class associations *e.g.*, the probability of blue dots belonging to the 'car' region.

Specifically, we design an affinity map to generate two types of representations for each pixel. One only considers the dependency between pixels of the same class, which is intra-class repre-

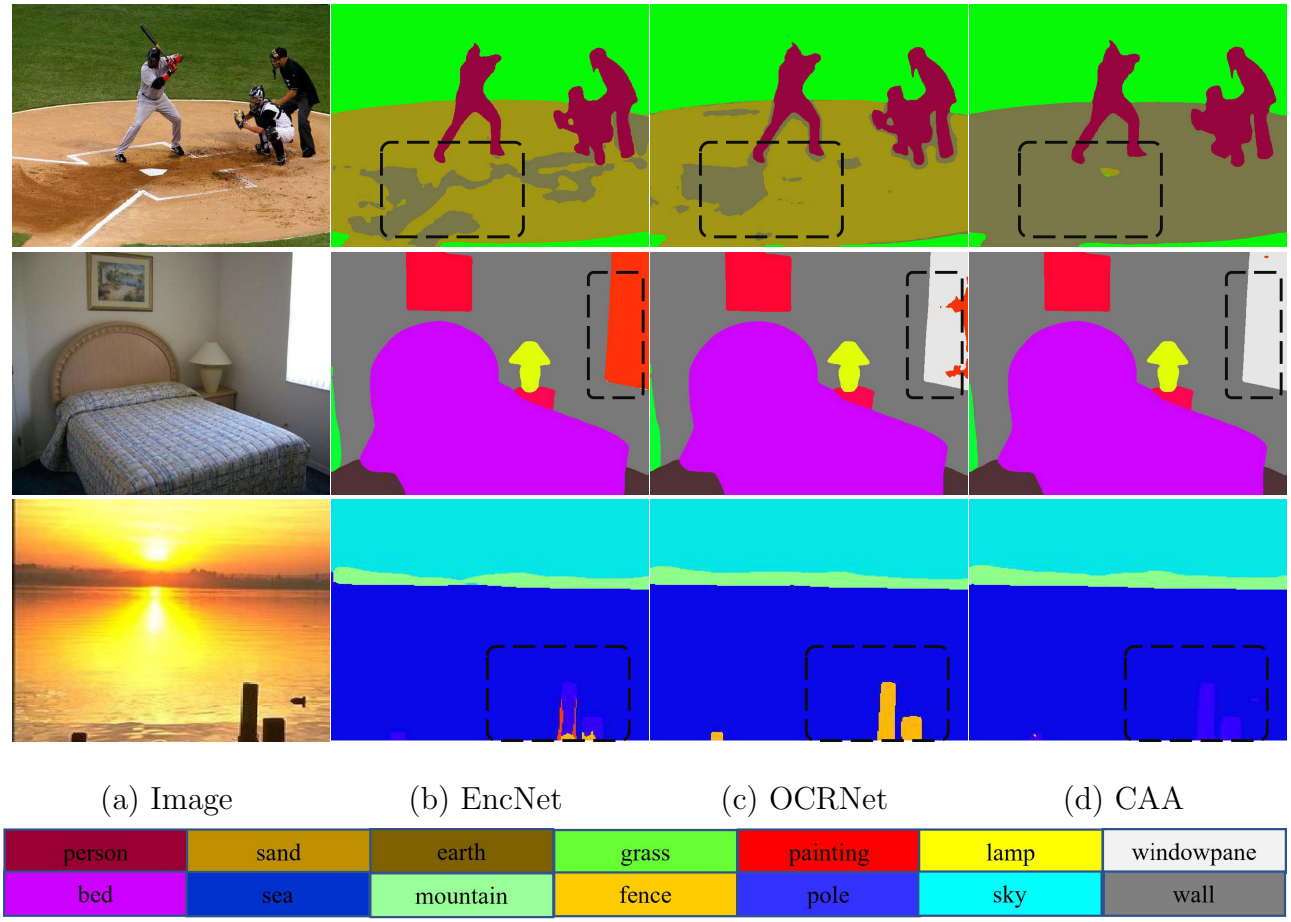


Figure 3.2: **Examples of segmentation results for ADE20K. The results of EncNet and OCRNet are shown in (b) and (c), which explore the pixel-to-pixel relations and pixel-to-class relations, respectively. Our proposed CAA module combines the relation of pixels-pixels and pixels-class association for the final prediction. Obviously, our method achieves better prediction than the methods mentioned above, as shown in (d).**

sentation. Another considers the relation of pixels across different classes, which is inter-class representations. Besides, an affinity loss is developed to enforce the network to better generate pixel-to-pixel relations. With the affinity map, we can better learn whether two pixels belong to the same class or not. Due to the fact that the label assigned to a pixel signifies the category of the object it is associated with, we further calculate the relation between pixels and the class center. The class center is determined through the aggregation of feature vectors from all pixels within the same class label.

We show examples of segmentation results produced by the attention-based (pixel-to-pixel) method (EncNet [21]) and class-level (pixel-to-class) method (OCRNet [30]) in Figure 3.2. In

the first row and the last row of Figure 3.2, both EncNet [21] and OCRNet [30] recognize 'earth' as 'sand' and 'pole' as 'fence', while our CAA (joining pixel-to-pixel relation and pixel-to-class association together in a unified framework) successfully recognizes the class 'earth' and 'pole'. In the second row, OCRNet [30] performs better than EncNet [21] but still makes some mistakes (recognizes 'windowpane' as 'painting'), while our CAA recognizes the right class.

In a nutshell, this thesis presents the following key contributions:

- We introduce a class-aware affinity module (CAA) to mitigate the issue of intra-class compactness and inter-class dispersion. Our CAA explores both pixel-to-pixel relations and pixel-to-class associations. This module can be effortlessly incorporated into existing segmentation frameworks and improve the performance of the corresponding model in which CAA is added.

- An affinity map is proposed to learn the pixel-to-pixel relation and generate intra- and inter-class representations.

- Class center is proposed to explore the pixel-to-class associations for further corresponding context calculation.

3.2 Proposed Method

3.2.1 Motivation

Analyzing the pixel-to-class association provides the most lucid depiction of the semantic interdependence between the specified pixels and their adjacent context. For instance, it is uncommon to find a bike in water. Therefore, classifying the presence of water can help minimize the likelihood of erroneously identifying an object in the water as a bike. Hence, it is necessary to enhance the pixel-to-class association.

Meanwhile, pixels near object boundaries are prone to misclassification in semantic segmentation. To address this, we assess whether two adjacent pixels belong to the same class, leveraging pixel-to-pixel relations together with pixel-to-class association to reduce boundary ambiguity. To this end, we design a Class-Aware Affinity Module (Figure 3.3) that enhances intra-class compactness while suppressing inter-class dependency.

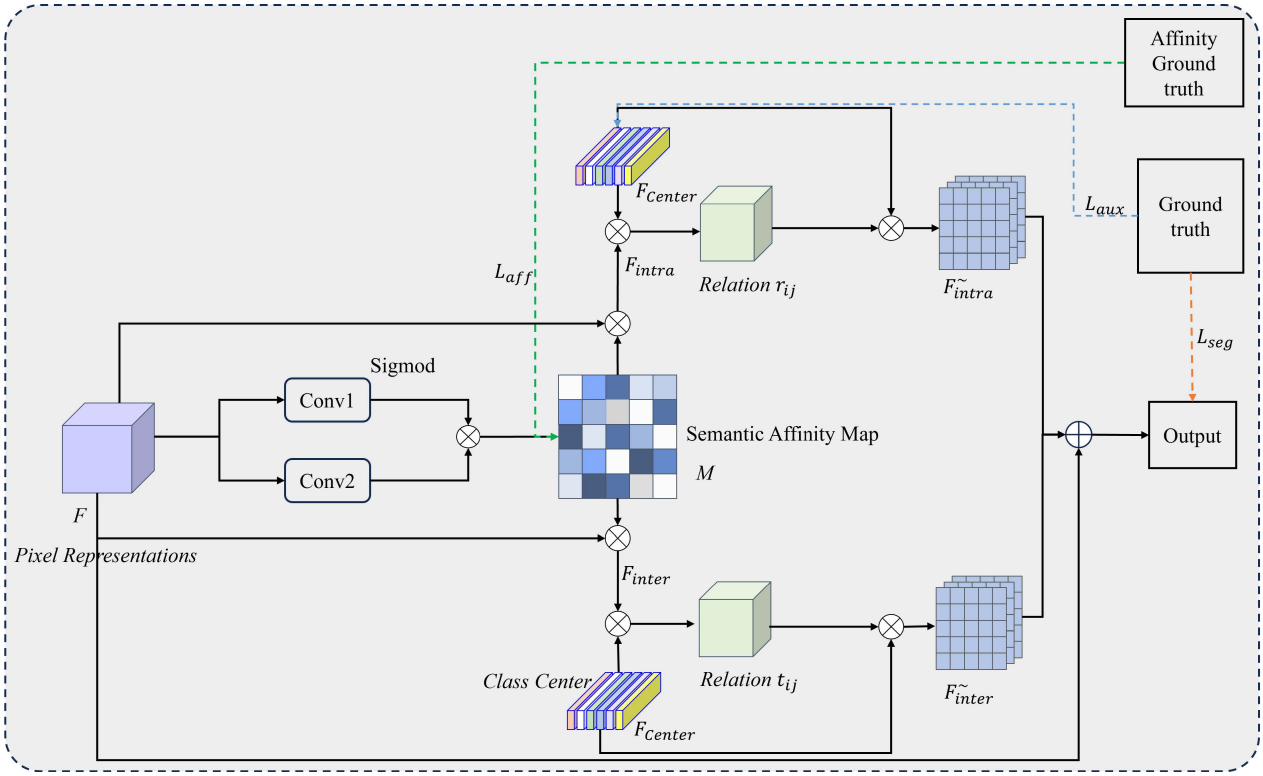


Figure 3.3: **Illustrating the pipeline of CAA. CAA explores the pixel-to-pixel relation and pixel-to-class association by leveraging the semantic affinity and class center. Semantic affinity explores the pixel dependencies to learn pixel-to-pixel relations. We utilize the class center to further calculate the pixel-class dependencies by considering category representations.**

3.2.2 Overview

We propose a lightweight Class-Aware Affinity (CAA) module that enhances semantic segmentation by jointly modeling pixel-to-pixel relations and pixel-to-class associations. As shown in Figure 3.3, the backbone first extracts pixel features and produces a coarse segmentation map. These are used to construct (1) a semantic affinity map that captures intra- and inter-class pixel relations, and (2) class centers that summarize class-specific feature representations.

The two components provide complementary contextual cues: affinity maps strengthen spatial consistency, while class centers enable pixels to reason over class-level semantics. By integrating these cues, CAA produces refined feature representations that reduce intra-class variation and improve inter-class distinction. Due to its modular design, CAA can be seamlessly plugged into existing segmentation frameworks to achieve consistent performance gains.

3.2.3 Class Center

The feature generator receives the input image I and projects it into a high-dimensional feature $F \in \mathbb{R}^{D\times H\times W}$ and coarse segmentation result $F_{coarse} \in \mathbb{R}^{N_{class}\times H\times W}$. To minimize the computational expenses, we apply 1×1 $conv \rightarrow BN \rightarrow ReLU$ operations to decrease the channel dimension to $D^{'}$, where BN stands for batch normalization and $ReLU$ is the Rectified Linear Unit function. Then, we reshape F_{coarse} to $\mathbb{R}^{N_{class}\times HW}$ and F to $F' \in \mathbb{R}^{D^{'}\times HW}$, as illustrated in Figure 3.4. After that, the class center, denoting a pixel-level probability output for each class, is computed by:

$$F_{center} = Softmax(F_{coarse}) \otimes F'^{T} \tag{3.1}$$

where $F_{center} \in \mathbb{R}^{N_{class}\times D^{'}}$. We learn the class center with the supervision of ground truth using cross-entropy loss during the training phase. The class center aids the model in comprehensively learning representations for all classes from a global perspective. Moreover, we can calculate the consistency between a pixel and each class center to improve the segmentation performance.

It is worth noting that unlike approaches employing multiple static prototypes to handle intra-class variance across a dataset, our class center is *dynamically* computed from the specific input image. Since F_{center} aggregates feature representations directly from the visible regions of the current scene, a single dynamic center is sufficient to capture the instance-specific global context without the computational redundancy of multiple prototypes.

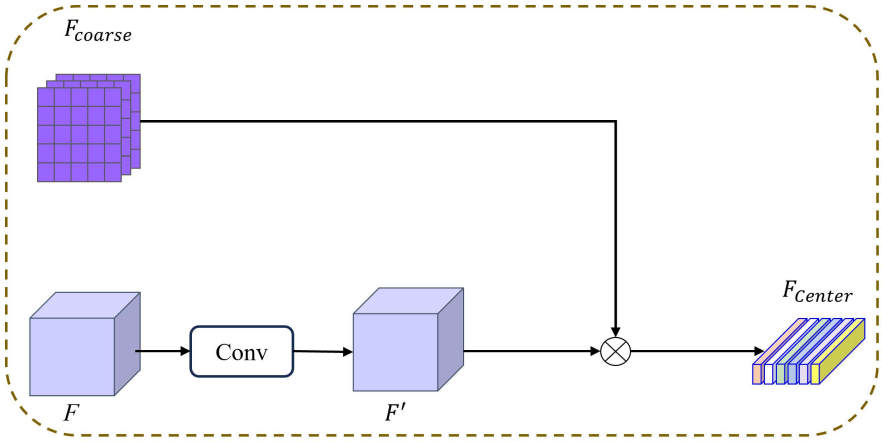


Figure 3.4: **Illustration of Class Center. Class centers are computed by aggregating pixel features F' within coarse segmentation regions F_{coarse}, enabling subsequent pixel-to-class association.**

3.2.4 Semantic Affinity

Affinity Map

Intra-class compactness and inter-class dispersion will determine segmentation performance to some extent. However, some works [18], [19] propose to aggregate local features as a mixture, which can result in the misclassification of distinct categories. To simulate extensive contextual information, we incorporate affinity maps to represent refined local features by distinguishing pixels of the same category from those of distinct categories. As depicted in Figure 3.3, $F \in \mathbb{R}^{D\times H\times W}$ represents the local feature, where $H \times W$ represents the resolution and D represents the dimension of channels.

Then, to reduce the channel dimension, we apply two 1×1 convolutional layers on feature F to produce Affinity Map, denoted as M, with dimensions $M \in \mathbb{R}^{(H\times W)\times(H\times W)}$. We directly learn pixel-wise correlations across intra- and inter-class regions using the affinity map, capturing whether the i^{th} pixel and j^{th} pixel belong to the same category. We further extract pixel representations of intra- and inter-class in the following manner:

$$F_{intra} = M \otimes F \tag{3.2}$$

$$F_{inter} = (1 - M) \otimes F \tag{3.3}$$

the feature size of F_{intra} and F_{inter} are $H \times W$, and $\otimes$ represents matrix multiplication.

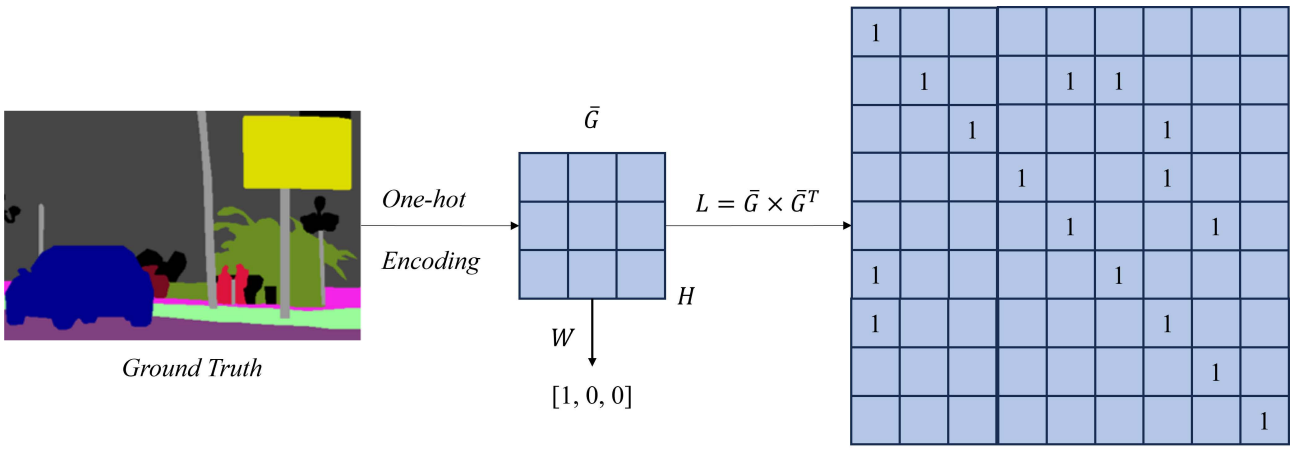


Figure 3.5: **The process of generating the Affinity Ground Truth. We down-sample the ground truth and apply one-hot encoding to obtain $\bar{G}$. Then, matrix multiplication is conducted to generate the affinity ground truth.**

We employ binary cross-entropy loss to supervise the affinity map, treating it as a binary classification task for each pixel, distinguishing between the same class or different classes. In particular, we employ a down-sample operation to reshape ground truth G and match the size of feature F. Next, we perform one-hot encoding to obtain the transformed ground truth denoted as $\bar{G}$. Ultimately, a multiplication operation is conducted between $\bar{G}$ and its transpose to derive the affinity ground truth: $L = \bar{G} \times \bar{G}^T$. The process of constructing L is illustrated in Figure 3.5. From the affinity ground truth, we can learn that $L_{ij} = 1$ means the i^{th} pixel and the j^{th} from the original image are in the same class, while $L_{ij} = 0$ means the i^{th} pixel and the j^{th} from the original image are in different classes.

Intra-class Association

These pixel representations only contain the spatial relation between any two pixels. That is, we can only know whether two pixels belong to the same category or not, but the category information has not been extracted. Thus, we compute the association between intra-class pixel representations F_{intra} and class center representation F_{center} in the following manner:

$$r_{ij} = \frac{exp(F_{intra_i} \cdot F_{center}j)}{\sum_{j=1}^{N} exp(F_{intra_i} \cdot F_{center}j)} \tag{3.4}$$

where r_{ij} stands for the correlation between the i^{th} pixel and other pixels in the j^{th} class center. Furthermore, it consolidates the correlation between the i^{th} pixel and other pixels within the same category.

We enhance the feature representations by conducting a matrix multiplication between the relation s_{ij} and the class context representation F_{center} to yield the output $\tilde{F_{intra}}$ as outlined below:

$$\tilde{F_{intra}} = \sum_{j=1}^{N} (r_{ij} F_{center}j) \tag{3.5}$$

$\tilde{F_{intra}}$ refers to the augmented intra-class representations.

Inter-class Association

To mitigate the issue of inter-class dispersion, we calculate the correlation between the inter-class pixel representations F_{inter} and the class context representation $F_{center}j$. Much like the intra-class association, we also have:

$$t_{ij} = \frac{exp(F_{inter_i} \cdot F_{center}j)}{\sum_{j=1}^{N} exp(F_{inter_i} \cdot F_{center}j)} \tag{3.6}$$

t_{ij} denotes the relation between the i^{th} pixel and other pixels in the j^{th} class center. We enhance the feature representations by conducting a matrix multiplication between the relation t_{ij} and the class center representation $F_{center}j$ to obtain the output $\tilde{F_{inter}}$ as outlined below:

$$\tilde{F_{inter}} = \sum_{j=1}^{N}(t_{ij}F_{center}j) \tag{3.7}$$

$\tilde{F_{inter}}$ represents the inter-class representations. We further combine F, $\tilde{F_{intra}}$ and $\tilde{F_{inter}}$ for final prediction:

$$F_{final} = \Delta(Concat(F, \tilde{F_{intra}}, \tilde{F_{inter}})) \tag{3.8}$$

where Δ denotes a convolutional layer utilizing a 1×1 filter to decrease the dimension of output channels.

3.2.5 Loss Function

To supervise affinity learning and class-center estimation, we employ three loss terms: the affinity loss, the auxiliary loss for class-center learning, and the final segmentation loss.

Affinity Loss. Let $G = \{g_n\}_{n=1}^{N^2}$ denote the ground-truth affinity map and $A = \{a_n\}_{n=1}^{N^2}$ the predicted affinity map. Each element $g_n, a_n \in [0, 1]$ indicates whether the n-th pixel pair belongs to the same class. The affinity loss is computed using binary cross-entropy:

$$\mathcal{L}_{aff} = -\frac{1}{N^2}\sum_{n=1}^{N^2}\left[g_n \log(a_n) + (1 - g_n)\log(1 - a_n)\right]. \tag{3.9}$$

Auxiliary Loss. To guide the training of the class-center estimation, we introduce an auxiliary pixel-wise cross-entropy loss. Given predicted probabilities $P = \{p_n\}_{n=1}^{N^2}$ and ground-truth labels $Y = \{y_n\}_{n=1}^{N^2}$:

$$\mathcal{L}_{aux} = -\frac{1}{N^2}\sum_{n=1}^{N^2}\left[y_n \log(p_n) + (1 - y_n)\log(1 - p_n)\right]. \tag{3.10}$$

Final Segmentation Loss. The final segmentation output $P' = \{p'_n\}_{n=1}^{N^2}$ is supervised with the same pixel-wise cross-entropy:

$$\mathcal{L}_{seg} = -\frac{1}{N^2}\sum_{n=1}^{N^2}\left[y_n \log(p'_n) + (1 - y_n)\log(1 - p'_n)\right]. \tag{3.11}$$

Overall Objective. The final loss combines the three components:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{seg} + \lambda_2 \mathcal{L}_{aff} + \lambda_3 \mathcal{L}_{aux}, \tag{3.12}$$

where $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 0.4$ balance the contributions of each term.

3.2.6 Integration with other methods

Our CAA module can be effortlessly integrated into existing segmentation frameworks, such as FCN [4], PSPNet [19], DeepLabV3 [17], UperNet [69], and ISNet [12], serving as a plug-and-play component to enhance their capabilities. These frameworks primarily act as feature extractors, providing spatial and semantic representations of the input image. As illustrated in Figure 3.1, we utilize their extracted features and further enrich them with our CAA module, which simultaneously strengthens both pixel-to-pixel (spatial) and pixel-to-class (semantic) correlations, leading to consistent performance improvements.

3.3 Experiments

We evaluate the effectiveness of our approach on three distinct semantic segmentation benchmarks: ADE20K [1], COCO-Stuff dataset [70], and Pascal-Context dataset [71].

3.3.1 Datasets

- ADE20K. ADE20K is a challenging semantic segmentation dataset, composed of more than 20,000 images. The dataset contains 150 semantic categories, *i.e.*, sky, road, grass as well as discrete objects such as people, cars, and beds. It is partitioned into 20k/2k/3k images for training, validation, and testing respectively.

- COCO-Stuff. The COCO-Stuff10k dataset consists of 171 semantic classes, comprising 81 thing classes and 91 stuff classes. The training set comprises 9,000 images, while the testing set comprises 1,000 images.

- PASCAL-Context. The PASCAL-Context dataset encompasses 59 semantic pixel-level categories in all its training images. This includes 4,998 images designated for training and 5,105 images allocated for testing.

3.3.2 Implementation Details

We utilize Pytorch and MMSegmentation [72] toolbox to conduct the experiment. Then, we utilize the ImageNet [73] pre-trained ResNet [74] and transformer *e.g.*, ViT [8] and Swin-Transformer [75] as the backbone. We trained our model on two NVIDIA A40 GPUs, each with 48GB of memory.

For network optimization, we employ the stochastic gradient descent (SGD) algorithm with a momentum value of 0.9 and the "poly" learning rate policy with a factor of $(1 - \frac{iter}{iter_{max}})^{0.9}$. Additionally, we incorporate Synchronized batch normalization (SyncBN) during the model training process. Following the approach of prior studies [25], [30], [76], we utilize a multi-scale ratio ranging from 0.5 to 1.75 and apply data augmentation techniques like flipping and random cropping. In addition, we adopt the mean intersection of union (mIoU) as the evaluation metric. More specific settings are introduced for different benchmarks.

- ADE20K: For ADE20K, the initial learning rate is 0.02, crop size is 512×512, and weight decay is 0.0005. If not specified, we set 160k training iterations with batch size 16.
- COCO-Stuff: For COCO-Stuff, the initial learning rate is 0.001, crop size is 512×512, and weight decay is 0.0001. If not specified, we set 60k training iterations with batch size 16.
- PASCAL-Context: For PASCAL-Context, the initial learning rate is 0.001, crop size is 512×512, and weight decay is 0.0001. If not specified, we set 60k training iterations with a batch size of 16.

3.3.3 Comparisons with State-of-the-art Methods

In this section, we perform experiments on diverse benchmarks and compare the performance with other methods.

Results on ADE20K

In Table 3.1, we present a comparison of our results with those of other competitive methods on the ADE20K dataset. ADE20K is a comprehensive scene-parsing dataset, comprising a total of 150 semantic classes. We maintain consistent training and testing configurations for this dataset. To validate the efficacy of CAA, we conduct experiments and compare the results

Method	Backbone	ADE20K	COCO-Stuff	Pascal-Context
CCL [20]	ResNet101	-	35.70	51.60
PSPNet [19]	ResNet101	43.29	38.86	53.50
PSANet [67]	ResNet101	43.77	-	-
EncNet [21]	ResNet101	44.65	-	51.70
CFNet [77]	ResNet101	44.89	-	54.00
DANet [25]	ResNet101	45.22	39.70	52.60
OCRNet [30]	ResNet101	45.28	39.50	54.37
OCNet [76]	ResNet101	45.40	39.10	54.00
SpyGR [78]	ResNet101	-	39.90	52.80
CTNet [31]	ResNet101	45.94	-	55.50
RANet [79]	ResNet101	-	40.70	54.90
SPNet [11]	ResNet101	45.60	-	54.50
ACNet [22]	ResNet101	45.90	40.10	54.10
CPNet [32]	ResNet101	46.27	-	53.90
FLANet [80]	ResNet101	46.68	-	-
PCAA [81]	ResNet101	46.74	-	55.60
ISNet [12]	ResNet101	47.31	41.60	-
UperNet [69]	Swin-Large	52.10	48.60	60.30
UperNet+CAR [82]	Swin-Large	-	44.88	58.97
GSS-FT-W [83]	Swin-Large	48.54	-	-
SegDeformer [84]	Swin-Large	53.90	49.51	-
TSG [85]	Swin-Large	54.2	-	63.3
UperNet+CAC [43]	Swin-Large	54.43	-	-
UperNet [86]	InternImage-B	51.30	45.30	61.34
UperNet+CAA (*ours*)	ResNet101	47.45	41.00	55.57
UperNet+CAA (*ours*)	ViT-Large	50.35	45.64	61.11
UperNet+CAA (ours)	InternImage-B	51.86	46.64	62.08
UperNet+CAA (*ours*)	Swin-Large	**54.59**	**49.96**	**64.38**

Table 3.1: **Comparisons with the state-of-the-art methods. We employ a multi-scale and flipped testing approach to compare segmentation performance on the validation set of ADE20K, the testing set of COCO-Stuff, and the testing set of Pascal-Context. We utilize mIoU as the evaluation metric, and the best performance is in bold.**

(a) Image (b) UperSwin (c) UperSwin+CAA (d) Ground Truth

Figure 3.6: **Visualizations on the ADE20K validation set. We compare the qualitative results of UperNet+Swin and our UperNet+Swin+CAA.**

with various methods. The previous best method ISNet [12] utilizes semantic-level and image-level context for semantic segmentation, achieving 47.31% mIoU. Our UperNet+CAA achieves 47.45% mIoU with ResNet-101 that is 2.17%, 1.18% and 0.14% higher than OCRNet [30], CPNet [32], and ISNet [12] respectively. Owing to the effectiveness of the proposed CAA, our UperNet+CAA achieves 51.86% mIoU by adopting InternImage-B. Furthermore, employing the Swin-Large backbone, our UperNet+CAA achieves a new state-of-the-art performance with a mIoU of 54.59%.

Moreover, to validate the efficacy of CAA, we provide visualizations of qualitative results obtained from the val set of ADE20K (refer to Figure 3.6). The figure illustrates that our UperNet+Swin+CAA effectively handles issues related to intra-class compactness and inter-class dis-

persion, resulting in superior segmentation outcomes compared to the baseline UperNet+Swin.

Results on COCO-Stuff

COCO-Stuff is a challenging benchmark, including 81 thing classes (objects with distinct shapes, *e.g.*, car and person) and 91 stuff classes (amorphous background regions, *e.g.*, grass and sky). Table 3.1 compares the results with CCL [20], DANet [25], OCRNet [30], ISNet [12], etc. When utilizing ResNet-101 as a pre-trained network, our UperNet+CAA achieves 41.00% mIoU, exceeding OCRNet [76] and DANet [25] with a mIoU of 1.5% and 1.3%, respectively. When leveraging a more robust backbone InternImage-B, our method achieves mIoU of 46.64% mIoU. We have also integrated CAA into the widely used Swin-Large backbone and UperNet+CAA obtains 49.96% mIoU. These results prove the effectiveness of our CAA.

Results on PASCAL-Context

We also perform experiments on the PASCAL-Context to conduct a comparative analysis with existing methods. Table 3.1 showcases the segmentation outcomes. With ResNet-101 as the pre-trained network, our UperNet+CAA attains a mIoU of 55.57%. We exceed PSPNet [19], DANet [25] and CPNet [32] with a mIoU of 2.07%, 2.97%, and 1.67%. By employing InternImage-B, UperNet+CAA achieves an impressive mIoU of 62.08%. Furthermore, our approach, UperNet+CAA with Swin-Large, establishes a mIoU of 64.38%.

In general, our UperNet+CAA attains the best result on ADE20K and Pascal-Context. Although we do not reach the best result with ResNet101 on the COCO-Stuff dataset compared with ISNet [12] and PCAA [81], we still rank within the top-3. This achievement showcases the competitiveness and effectiveness of our approach across multiple benchmarks. By achieving top rankings across various datasets, our model demonstrates its robustness and adaptability in addressing diverse visual understanding tasks.

3.3.4 Ablation Study

For Class Center and Semantic Affinity

We conduct ablation experiments on the ADE20K val set to verify the efficacy of our CAA. For the ablation studies, however, we use FCN [4] with a ResNet-50 backbone. This choice is intentional: FCN is computationally lightweight and conceptually simple, making it more

Baseline	CC	SA	Backbone	mIoU(%)
✓			ResNet50	36.10
✓	✓		ResNet50	40.39
✓		✓	ResNet50	42.43
✓	✓	✓	ResNet50	43.12

Table 3.2: **Ablation study conducted on the ADE20K validation set. "CC" denotes the use of only Class Center, while "SA" denotes the use of only Semantic Affinity. All methods employ a single scale for testing.**

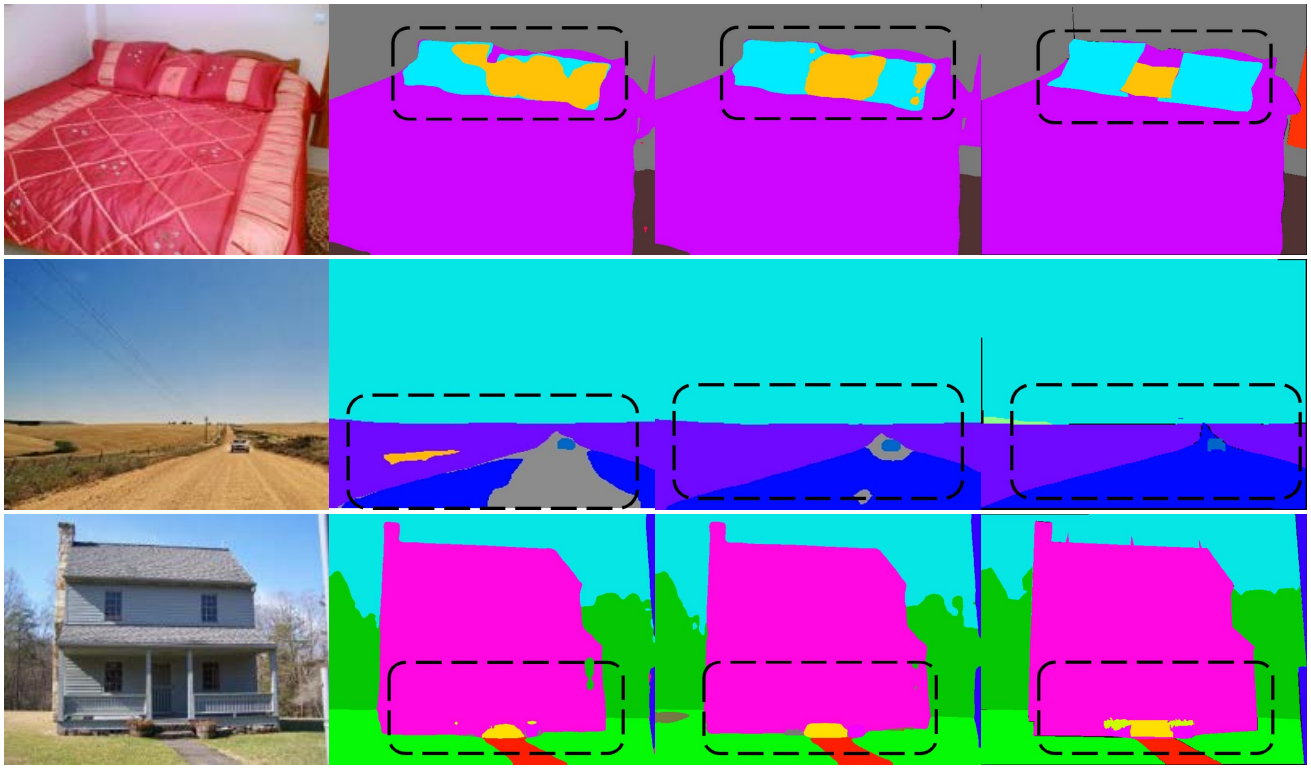
suitable for isolating and analyzing the effect of our proposed CAA without interference from complex decoder designs or heavy backbones such as ResNet-101, ViT, or Swin Transformer. To better show the importance of the class center and semantic affinity, we separate our CAA into Class Center (CC) and Semantic Affinity (SA), respectively.

CAA without Semantic Affinity (CAA w/o SA)

To mitigate the problem of category confusion, class-level context is very important. A car barely appears in the sky, so we need pixel-to-class association to predict a clear class for an object. As indicated in Table 3.2, the class center improves the performance significantly. In comparison to the baseline FCN (ResNet-50), it achieves a 40.39% mIoU on the ADE20K validation set, marking a notable improvement of 4.29%. Besides, we show the visualization segmentation results of intra-class representations in Figure 3.7. Likewise, it is justifiable to investigate the impact of pixel-to-pixel relationships.

CAA without Class Center (CAA w/o CC)

The objective of semantic segmentation is to allocate a semantic label to each individual pixel. Hence, pixel-level context plays a critical role in achieving superior performance. As indicated in Table 3.2, it is evident that the pixel-to-pixel relation yields a noteworthy enhancement of 6.33% mIoU on the ADE20K dataset in contrast to the baseline method, demonstrating that different pixels need different contextual dependencies. We also show the visualization segmentation results of inter-class representations in Figure 3.8.



(a) Image (b) CAA w/o SA (c) CAA (d) Ground Truth

Figure 3.7: **Illustration of segmented examples of CAA and CAA w/o SA.**

Class-Aware Affinity (CAA)

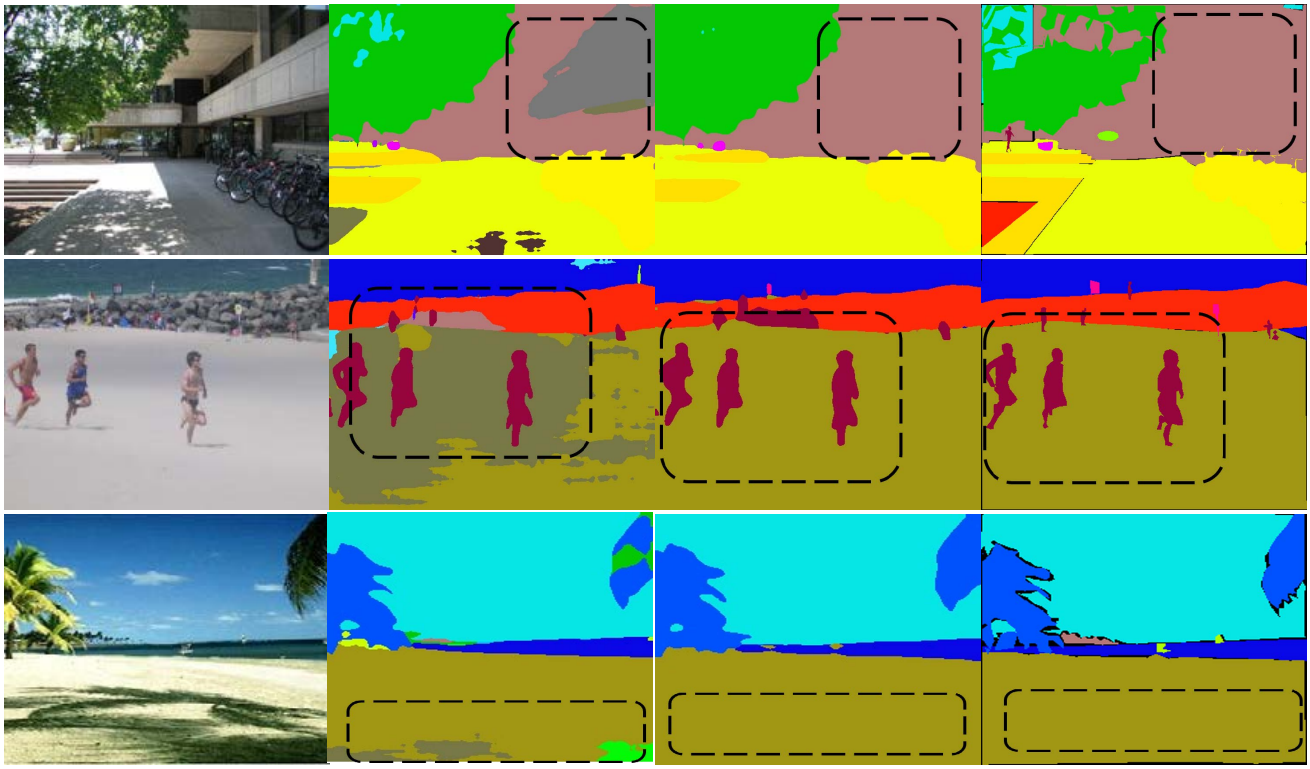
Existing methods do not explicitly model the pixel-to-pixel relation and pixel-to-class association. We propose to combine these two dependencies to aggregate the contextual information. As illustrated in Table 3.2, the CAA outperforms the baseline method by 7.02% mIoU on the ADE20K dataset. To further substantiate the effectiveness, we show the visualization of qualitative results in Figure 3.7 and Figure 3.8. From the results, we can indicate that pixel-to-pixel and pixel-to-class relations are complementary.

For single class association

We can extract intra- and inter-class pixel representations from our semantic affinity. Experiments show the effectiveness of the two-pixel representations.

Intra-class contextual information

Since each pixel has a semantic label, the intra-class contextual information is vital for semantic image segmentation. As shown in Table 3.3, the intra-class representations lead to a substantial



(a) Image (b) CAA w/o CC (c) CAA (d) Ground Truth

Figure 3.8: **Illustration of segmented examples of CAA and CAA w/o CC.**

performance boost. In comparison to the baseline FCN (ResNet-50), the intra-class representations achieve a mIoU of 42.76% on the ADE20K validation set, marking a notable improvement of 6.66%. Similarly, it is reasonable to explore the effect of inter-class representations.

Inter-class contextual information

The inter-class contextual information could differentiate the pixels in different classes. As shown in Table 3.3, we can see that inter-class representations bring an improvement of 6.29% mIoU on ADE20K, demonstrating that different pixels need different contextual dependencies. The performance of the inter-class is inferior to the intra-class so we can conclude that intra-class representations are more important.

Intra+Inter-class contextual information

We further combine the intra- and inter-class representations. As depicted in Table3.3, CAA outperforms the baseline method by 7.02% mIoU on the ADE20K dataset. From the results, we can indicate that the intra- and inter-class representations are complementary and promoted by

Baseline	Intra-class	Inter-class	Backbone	mIoU(%)
✓			ResNet50	36.10
✓	✓		ResNet50	42.76
✓		✓	ResNet50	42.39
✓	✓	✓	ResNet50	43.12

Table 3.3: **Ablation study conducted on the ADE20K validation set. All methods employ a single scale for testing.**

CAA	Aux	Aff	Backbone	mIoU(%)
✓			ResNet50	40.45
✓	✓		ResNet50	41.44
✓		✓	ResNet50	42.36
✓	✓	✓	ResNet50	43.12

Table 3.4: **Ablation study of loss function on the ADE20K validation set. Aux represents only using auxiliary loss and the final cross-entropy loss. Aff represents only using affinity loss and the final cross-entropy loss.**

each other. In a word, it is necessary to synchronously capture intra- and inter-class contexts to make them promote each other.

For loss function

The CAA module leverages both an affinity loss and an auxiliary loss to boost segmentation performance. We explore the significance of these two losses in this section. The results are presented in Table 3.4. Only using auxiliary loss and the final cross-entropy loss, our CAA module attains a mIoU of 41.44%. On the contrary, only utilizing the affinity loss and the ultimate cross-entropy loss leads to a mIoU of 42.36%. When leveraging both affinity loss and auxiliary loss, we achieve a mIoU of 43.12%, indicating that both the affinity map and class context need to be supervised.

Computational Complexity and Training time

We report the computational complexity of our CAA integrated with existing segmentation frameworks in Table 3.5, including the increased parameters, computation complexity, and accuracy. The results are calculated based on input size $512 \times 64 \times 64$ (8 times down-sampled from 512×512). Since CAA aims to aggregate the contextual information by modeling pixel-to-

Method	Params(M)	FLOPs(G)	mIoU (%)
PSPNet [19]	23.14	77.71	42.48
CCNet [13]	24.00	99.65	42.08
Deeplabv3 [17]	42.28	168.91	42.66
UperNet [69]	40.58	179.75	42.05
PSPNet [19] + CAA	26.75	90.76	44.12
CCNet [13] + CAA	27.60	112.69	43.98
Deeplabv3 [17] + CAA	45.88	181.96	44.23
UperNet [69] + CAA	44.18	192.48	43.81

Table 3.5: **Comparison of computational complexity and accuracy on ADE20K.**

pixel relation and pixel-to-class association, CAA is complementary to the existing segmentation frameworks. After integrating CAA with existing segmentation schemes, the whole model complexity is still acceptable. As an illustration, when incorporating CAA with PSPNet, the parameters and FLOPs experience a mere increase of 3.61M and 23.05G, respectively. However, the segmentation performance improves from 42.48% to 44.12% on ADE20k.

Besides, we also report the time of training 6k iterations and inference time per image on Pascal-Context testing set in Table 3.6. There is no doubt that the training and inference time is increased because of the additional processing carried out by the proposed CAA. However, the clear value is that additional CAA provides clear performance boosting in terms of mIoU.

Intergrated with Various Semantic Segmentation Frameworks

Our CAA module seamlessly integrates with pre-existing segmentation frameworks. To affirm the effectiveness and robustness of our approach, we incorporate the proposed CAA into five existing segmentation frameworks, namely FCN, PSPNet, UperNet, DeepLabV3, and ISNet. The performance comparison across three benchmark datasets: ADE20k, COCO-Stuff, and Pascal-Context, is presented in Table 3.7. We conduct single-scale testing to ensure a fair comparison. In the case of the vanilla segmentation framework (i.e., FCN), integrating CAA leads to a remarkable improvement of 7.02% in mIoU on ADE20K, 4.37% on COCO-Stuff, and 5.06% on Pascal-Context. With a stronger baseline like DeeplabV3, CAA elevates the mIoU

Method	Training(min)	Inference(ms/p)	mIoU
FCN [4]	61	59.6	45.76
PSPNet [19]	69	57.6	50.21
Deeplabv3 [17]	91	67.0	50.73
UperNet [69]	44	66.3	48.90
FCN [4] +CAA	75(↑23%)	68.6(↑15%)	50.82
PSPNet [19] + CAA	91(↑32%)	72.6(↑26%)	52.05
Deeplabv3 [17] + CAA	114(↑25%)	79.3(↑18%)	52.37
UperNet [69] + CAA	71(↑61%)	78.2(↑18%)	51.57

Table 3.6: **Comparison of training time and inference time on Pascal-Context. ms/p represents the time of inferencing a picture.**

performance from 42.66% to 44.23% on ADE20K, from 35.73% to 37.39% on COCO-Stuff, and from 50.73% to 52.37% on Pascal-Context. The smaller gain on newer models (*e.g.*, ISNet) is expected since these architectures already provide strong context modeling, leaving limited space for further improvements. This performance comparison highlights the flexibility and applicability of the CAA module.

3.4 Conclusion

This chapter addresses the challenge of improving pixel-wise segmentation accuracy by capturing fine-grained contextual relationships, particularly under class imbalance. The motivation is to enhance segmentation frameworks by modeling pixel-to-pixel and pixel-to-class relationships, which are often overlooked, leading to suboptimal class differentiation.

To this end, we propose a novel Class-Aware Affinity (CAA) module that effectively models semantic affinity by learning class centers and their associations with pixel-level representations. Specifically, the CAA module extracts both pixel-to-pixel relations and pixel-to-class associations to construct a semantic affinity map, which determines whether a pair of pixels belongs to the same semantic category. Based on this map, we derive intra-class and inter-class representations that capture detailed contextual cues within and across classes. These are further aggregated to update class center representations dynamically, thereby improving class discrimination.

The principle behind CAA lies in its ability to jointly model semantic affinity and class-specific

Method	Backbone	ADE20K	COCO-Stuff	Pascal-Context
FCN	ResNet50	36.10	30.86	45.76
FCN+CAA	ResNet50	43.12 (+7.02)	35.23 (+4.37)	50.82 (+5.06)
CCNet	ResNet50	42.08	35.43	47.93
CCNet+CAA	ResNet50	43.98 (+1.9)	36.27(+0.84)	49.79 (+1.86)
UperNet	ResNet50	42.05	35.80	48.90
UperNet+CAA	ResNet50	43.81 (+1.76)	37.08 (+1.28)	51.57 (+2.67)
PSPNet	ResNet50	42.48	36.33	50.21
PSPNet+CAA	ResNet50	44.12 (+1.64)	37.59 (+1.26)	52.05 (+1.84)
Deeplabv3	ResNet50	42.66	35.73	50.73
Deeplabv3+CAA	ResNet50	44.23 (+1.57)	37.39 (+1.66)	52.37 (+1.64)
ISNet	ResNet50	43.77	38.06	51.74
ISNet+CAA	ResNet50	43.98 (+0.21)	38.35 (+0.29)	52.08 (+0.34)

Table 3.7: **We evaluate the performance of integrating CAA into various mainstream frameworks across different benchmarks.**

contextual relationships, which enables more precise class center estimation. By leveraging these centers in the learning process, the model becomes more sensitive to subtle semantic differences and better equipped to handle class imbalance. Furthermore, the CAA module is lightweight and modular, making it easy to integrate into a wide range of existing segmentation frameworks without significant computational overhead.

Comprehensive experiments conducted on three challenging semantic segmentation benchmarks—ADE20K, COCO-Stuff, and Pascal-Context—demonstrate the effectiveness of our method. The CAA-enhanced framework achieves mIoU scores of 54.59% on ADE20K, 49.96% on COCO-Stuff, and 64.38% on Pascal-Context, showing consistent improvements over strong baselines. These results confirm that our method enhances both global context understanding and local semantic alignment, leading to more accurate segmentation maps.

In summary, our work contributes a simple yet powerful module for modeling semantic affinity in a class-aware manner, offering a practical solution to long-standing challenges in semantic segmentation.

Chapter 4

Classifier Enhancement Using Extended Context and Domain Experts for Semantic Segmentation

In the previous chapter, we proposed the Class-Aware Affinity (CAA) module, which aims to capture class-level associations across object boundaries—an aspect often overlooked in conventional segmentation frameworks that mainly focus on localized pixel-to-pixel relationships. While the CAA module improves the understanding of semantic boundaries and enhances intra-class consistency, it remains inherently limited by the information present within a single image.

However, semantic segmentation tasks in real-world scenarios frequently involve highly diverse class distributions across different images. A fixed classifier trained on a global label distribution may struggle to adapt to the varying semantic contexts found in individual images, leading to suboptimal performance—especially when rare classes or domain-specific concepts are involved. This is further exacerbated by class imbalance issues that result in biased predictions toward dominant categories.

To tackle these challenges, this chapter introduces a method for enhancing the classifier by leveraging cross-image contextual information and domain-specific knowledge embedded within the training dataset. Specifically, we propose an Extended Context-Aware Classifier (ECAC) that dynamically integrates both global (dataset-level) and local (image-level) contextual cues to

guide pixel-wise classification. The ECAC module adjusts its prediction behavior by considering not only the visual appearance of local regions but also the high-level semantic context drawn from similar images and the overall dataset structure.

By embedding this extended context and domain-informed strategy into the classifier design, ECAC significantly improves segmentation performance across diverse datasets and contributes to a more robust and generalizable semantic understanding.

4.1 Introduction

The existing encoder-decoder structure extracts features and then uses up-sampling to restore the spatial details, resulting in significant enhancements in semantic segmentation. In the decoding stage, the model restores the spatial information and applies the class label on each pixel to get the segmentation results. An essential component in the decoder is the classifier that ultimately assigns a label to each pixel. Existing methods generally utilize the classic vanilla classifier to learn a set of fixed parameters from the training data. This results in an inherent challenge when facing highly diverse data (*e.g.* different objects, scenes, or conditions) during training. Additionally, the vanilla classifier is particularly sensitive to class imbalance, where the model tends to prioritize majority classes while neglecting minority classes. To solve this problem, several methods [39]–[41] propose to address class imbalance by modifying the loss function, such as using weighted cross-entropy to increase the importance of minority classes or exploring the potential of AUC optimization in pixel-level long-tail problems. However, these approaches often rely on indirect adjustments, requiring careful hyperparameter tuning and potentially failing to address underlying feature representation or data distribution issues. Other methods [36], [37], [42] leverage prototypes or memory banks to mitigate the imbalance issue by maintaining robust feature representations for all classes.

Though effective, these methods still suffer from feature inconsistency when test images exhibit varying data distributions, a challenge that can be exacerbated by class imbalance during training, where minority class features may be underrepresented. To this end, the open question is: *Can we leverage information from the memory bank to enhance the classifier, thereby addressing class imbalance issues and improving prediction accuracy?* To tackle the aforementioned problems, this thesis proposes an Extended Context-Aware classifier (ECAC) that embeds global (dataset-level) and local (image-level) contextual information and a calibration

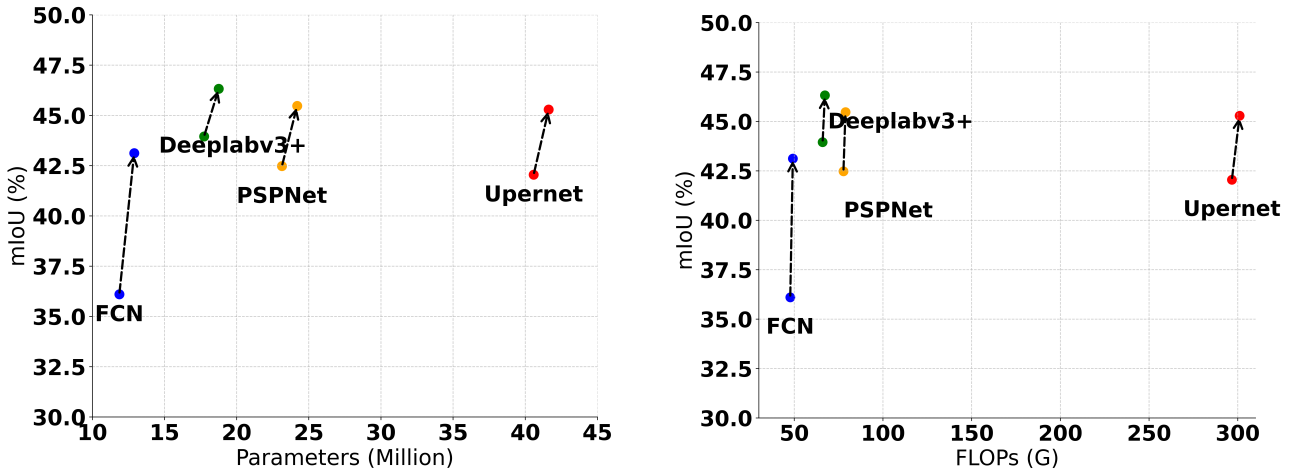
stage to mitigate the class imbalance issue.

First, we establish a dynamically updated memory bank to explicitly store dataset-level category information, serving as a dataset-level class center for each category. The memory bank independently updates each class, ensuring that even minority classes with limited samples retain sufficient category information. However, despite this design, the memory bank alone cannot fully overcome the biases induced by class imbalance. To address this, we adopt a teacher-student network framework inspired by [43], implementing knowledge distillation where the teacher network, informed by GT labels, generates a GT-based class center, and the student network leverages the image-level class center derived from individual images to mimic it. These representations are then combined with the memory bank via a concatenation operation to form an enhanced classifier. Since GT labels are unavailable during inference, this approach enables the classifier to better approximate GT-level performance, improving classification accuracy under imbalanced conditions. The learning process is supervised by the Kullback-Leibler (KL) divergence, ensuring effective knowledge transfer from the teacher to the student. Additionally, we propose an intra-variance loss to transfer intra-class feature variance from the teacher output to the student output, further enhancing the student model's segmentation performance. Second, to further improve robustness, we additionally introduce a calibration stage to adjust classification scores by applying a learnable linear transformation, aligning the classifier output with the true class distribution. This ensures a more balanced and robust prediction by dynamically fine-tuning the logits through learnable parameters, ultimately improving the fairness and accuracy of the model. Moreover, our approach improves the performance while only slightly increasing the model complexity, as demonstrated in Figure 4.1.

In a nutshell, our main contributions to this work are:

- We propose an Extended Context-Aware Classifier (ECAC) that effectively tackles the issues of class imbalance and varying class distributions in semantic segmentation. By embedding both global (dataset-level) and local (image-level) contextual information, ECAC enhances pixel-wise classification precision across diverse datasets.

- We develop a novel framework integrating a dynamically updated memory bank with a teacher-student network paradigm. The memory bank preserves dataset-level class representations, while the teacher-student mechanism, augmented by a calibration stage, refines contextual understanding and mitigates biases, achieving robust performance even



(a) Parameters (b) FLOPs

Figure 4.1: **Analysis of inference complexity and accuracy for ADE20K. The arrows in the figure represent the improvement achieved by our ECAC method. The lower ends of the arrows represent the performance of the original methods, while the upper ends show the improved performance after incorporating ECAC. Our ECAC significantly improves the segmentation performance while bringing a little computational complexity.**

for underrepresented classes.

- We establish ECAC as a lightweight, plug-and-play module compatible with existing segmentation architectures. Comprehensive experiments on benchmarks such as ADE20K, COCO-Stuff10K, and Pascal-Context demonstrate its superior performance and adaptability, significantly advancing state-of-the-art segmentation capabilities with minimal computational overhead.

4.2 Proposed Method

4.2.1 Overview

Our proposed ECAC framework is illustrated in Figure 4.2. We first set up a dynamically updated memory bank to store global (dataset-level) category information. This ensures that even minority classes with limited samples maintain sufficient class-specific representations, unlike a vanilla classifier that might struggle to adequately represent minority classes due to its dependence on balanced training across the dataset. However, it alone cannot fully address biases from class imbalance, and image-level features lack the rich context of Ground-

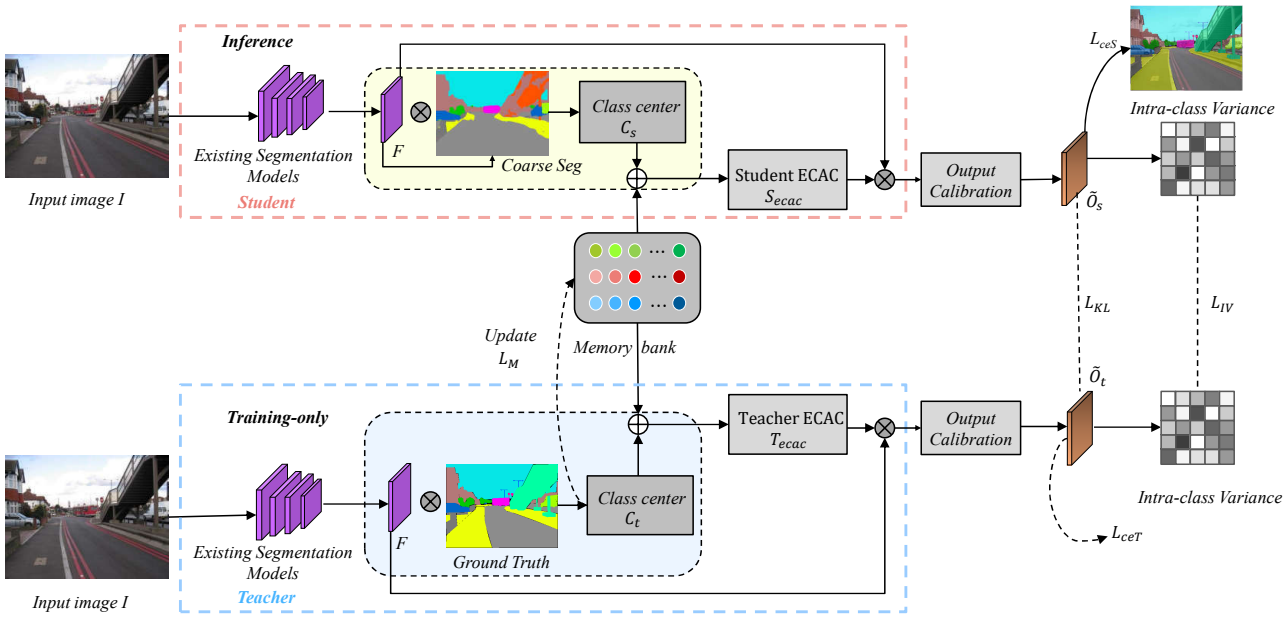


Figure 4.2: **The overview of ECAC. The memory bank $\mathcal{M}$ stores the dataset-level category information. Combined with the class center, we obtain an extended context-aware classifier. A calibration is adopted to mitigate the imbalanced issue. A teacher-student network is adopted to transfer comprehensive contextual information extracted by the ground-truth label to further enhance the classifier. (● ● ●) represents the mean features of each class.**

Truth (GT) labels. To tackle this, we use a teacher-student framework inspired by [43] with knowledge distillation. The teacher, guided by GT labels, creates a GT-based class center, while the student mimics it using image-level class centers from the memory bank. These are concatenated to form an enhanced classifier, approximating GT-level performance during inference when GT labels are unavailable, with learning supervised by KL divergence. We also add an intra-variance loss to transfer intra-class feature variance from teacher to student, boosting segmentation performance under imbalanced conditions. At last, we add a calibration stage that adjusts classification scores using a learnable linear transformation. This aligns the classifier output with the true class distribution, dynamically refining logits to improve fairness and accuracy.

4.2.2 Memory Bank

Given an input image $I \in \mathbb{R}^{3\times H\times W}$, an encoder network (*i.e.*, FCN [4] and PSPNet [19]) first maps pixels into a non-linear embedding space as $F \in \mathbb{R}^{d\times h\times w}$, where d, h, and w represent

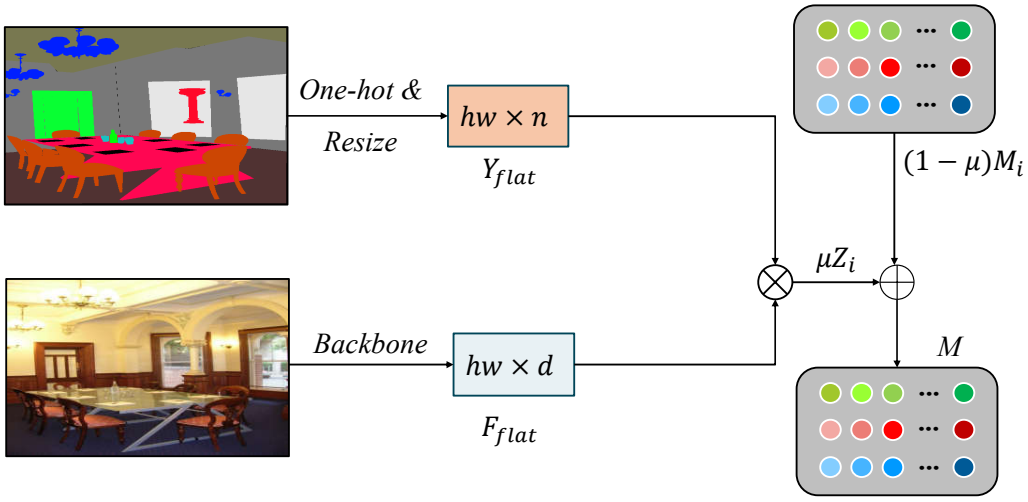


Figure 4.3: **The process of memory bank updating. The memory bank $\mathcal{M}$ is initialized as an empty structure of size $n \times d$. We update the memory bank using a momentum-based approach (Eq. 4.2), where $i-th$ class representation $\mathcal{M}_i$ is refined by blending the previous memory bank state with the newly computed class-specific features Z_i.**

the feature dimension, height and width, respectively. Then, we create an empty memory bank $\mathcal{M}$ of size $n \times d$ and initialize it by using the values in the masked feature map Z as shown in Figure 4.3, where n is the number of classes. To update the i-th item in the memory bank, we conduct average pooling across the designated region, guided by the segmentation mask corresponding to the i-th class, as follows:

$$Z = \frac{Y_{flat} F_{flat}^T}{K_i} \tag{4.1}$$

where K_i is the number of pixels belonging to the i-th class, Y_{flat} is a resized one-hot segmentation ground truth with the size of $n \times hw$ and $Z \in \mathbb{R}^{n \times d}$ is a masked feature map.

During the training phase, we compute the average feature vector for every class to set up the memory bank $\mathcal{M}$. The memory bank is updated after each training iteration using the following equation:

$$\mathcal{M} = (1-\mu)\mathcal{M}_i + \mu Z_i \tag{4.2}$$

where $\mu \in [0, 1]$ is the momentum parameter. $\mathcal{M}_i$ denotes the i-th class representation and Z_i is corresponding masked class center.

4.2.3 Knowledge Distillation

The memory bank stores dataset-level contextual information, acting as a global classifier, but it lacks the precision of ground truth labels. To address this, we adopt a teacher-student framework inspired by [43], [44] for knowledge distillation, designing a Teacher Extended Context-Aware Classifier (Teacher-ECAC) and a Student-ECAC. During training, Teacher-ECAC leverages individual image ground truth segmentation to refine the classifier with precise class information. However, since ground truth is unavailable during inference, we use a coarse segmentation from a pre-trained model for Student-ECAC to mimic the teacher's output. This architecture enables effective knowledge transfer from the teacher model, enhancing the student's performance in real-world scenarios.

Teacher-ECAC (T_{ecac})

To form ECAC, we propose to combine the memory bank $\mathcal{M}$ with the local (image-level) class center. Despite the superior performance of recently proposed methods based on the class center, they cannot be compared with ground truth labels. Therefore, we propose to use the teacher-student network, which leverages additional ground truth to calculate the class center during training for transferring comprehensive contextual information to the student network.

We first apply one-hot encoding on the ground-truth label Y and resize it as $Y_{flat} \in \mathbb{R}^{n \times hw}$. Then, we calculate the teacher's class center as follows:

$$C_t = \frac{Y_{flat} \times F_{flat}^T}{\sum_{i=1}^{hw} Y|(y_i = c_i)} \tag{4.3}$$

where Y_{flat} of size hw stores the ground truth category labels. Then, we concatenate C_t and $\mathcal{M}$ to get the teacher extended context-aware classifier:

$$T_{ecac} = \xi(C_t \oplus \mathcal{M}) \tag{4.4}$$

where ξ is a projector head consisting of two linear layers to reduce the channel dimension. T_{ecac} is of size $n \times d$. The final output of the teacher network is given by:

$$O_t = T_{ecac} \cdot F \tag{4.5}$$

where O_t provides more precise information to transfer knowledge to the student network.

It is worth noting that utilizing Ground Truth to generate C_t ensures the reliability of the supervision signal, specifically during the early training stages when the feature encoder is not

yet fully optimized. Unlike methods that rely on unstable pseudo-labels or prediction-based prototypes, the GT-guided C_t accurately represents the centroid of the true class distribution within the current feature space. This prevents the student network from aligning with noisy or incorrect class centers, thereby providing a consistent and trustworthy optimization direction throughout the entire learning process. Although the memory bank can learn the mean features of different classes, the imbalanced distribution issue of different datasets still exists. To tackle this issue, we adopt a unified distribution alignment strategy [87] to calibrate the classifier's output by aligning it with a reference distribution of classes, promoting balanced predictions.

Student-ECAC (S_{ecac})

For Student-ECAC, F is first divided into n class regions (coarse segmentation result) as $F_{coarse} \in \mathbb{R}^{n\times h\times w}$. Next, we get the flattened pixel representations $F_{flat} \in \mathbb{R}^{d\times hw}$ and flattened coarse segmentation result $F'_{coarse} \in \mathbb{R}^{n\times hw}$. We calculate the class center as follows:

$$C_s = \text{softmax}(F'_{coarse}) \times F^T_{flat} \tag{4.6}$$

Here, $C_s \in \mathbb{R}^{n\times d}$ is the average features of all pixel features to a category. Then, we concatenate C_s and $\mathcal{M}$ to obtain student extended context-aware classifier:

$$S_{ecac} = \xi(C_s \oplus \mathcal{M}) \tag{4.7}$$

where ξ is also a projector head a projector head consisting of two linear layers to reduce the channel dimension. S_{ecac} is of size $n \times d$. $\mathcal{M}$ is the memory bank of size $n \times d$ storing the dataset's mean features of different classes. n is the number of classes. Note that S_{ecac} serves as a more comprehensive classifier, yielding improved prediction results during inference, as shown in Equation 4.8.

$$O_s = S_{ecac} \cdot F \tag{4.8}$$

where Z_s is the final output of the student network.

Output Calibration for Enhancing Segmentation

We apply a calibration step to refine the final outputs of the teacher and student networks, addressing class imbalance and improving segmentation accuracy. For the teacher network, this calibration is defined as follows:

$$\tilde{O}_{t,i} = \gamma_i \cdot O_{t,i} + \delta_i, \quad \forall i \in \mathcal{C} \tag{4.9}$$

where $\tilde{O}_{t,i}$ represents the logits produced by the teacher classifier T_{ecac} (Equation 4.5). $\mathcal{C}$ represents the set of all class indices. γ_i is a learnable class-specific scaling factor that adjusts the magnitude of the logits for each class i, and δ_i is a learnable class-specific offset that shifts the decision boundary, both of which will be learned from data. Similarly, for the student network, the output is calibrated as shown:

$$\tilde{O}_{s,j} = \gamma_j \cdot O_{s,j} + \delta_j, \quad \forall j \in \mathcal{C} \tag{4.10}$$

Here, $\tilde{O}_{s,j}$ denotes the logits from the student classifier *Secac* (Equation 4.8). The learnable parameters γ_j and δ_j are specifically tailored for the student model and differ from those used in the teacher model. This output calibration serves as a critical refinement step following the teacher-student distillation process. By applying class-specific adjustments to the logits with parameters unique to the student, it better mitigates the bias toward majority classes that often arises due to imbalanced class distributions in semantic segmentation datasets.

Discussion on the Efficacy and Scope of Linear Calibration. The linear transformation $\tilde{O}_i = \gamma_i \cdot O_i + \delta_i$ is chosen for its parameter efficiency and targeted effectiveness against systemic prediction biases. Its efficacy stems from two class-specific mechanisms: the scaling factor γ_i corrects the confidence magnitude mismatch by boosting the logits of minority classes, thus directly counteracting class imbalance bias, while the offset δ_i translates the decision boundary for class i, which is essential for precise threshold tuning and logit alignment during distillation. This approach is highly effective for correcting errors caused by **data distribution biases** or **model capacity differences**.

Discussion on Prototype Strategy. While multi-prototype approaches [88] can capture intra-class variations, we adhere to a single-prototype design for two strategic reasons rooted in our ECAC architecture:

1. **Feature Alignment and Efficiency:** Our framework relies on the direct fusion of global memory $\mathcal{M}$ and local class centers C_s (or C_t). Since the local aggregation of the current input image inherently generates a single, coherent feature vector for each present class (*e.g.*, a $1 \times d$ vector per class), maintaining a corresponding single global prototype ensures a direct one-to-one feature alignment. This avoids the computational overhead and ambiguity of attention-based selection mechanisms required to match a local center against multiple global sub-centers.

2. **Dynamic Compensation:** Although we use a single global prototype, our classifier remains dynamic. The fused classifier adapts to the current image's specific mode (captured by C_s) calibrated by the global average ($\mathcal{M}$). This effectively addresses intra-class variance on-the-fly without the need to explicitly store and update a complex bank of multi-modal prototypes.

4.2.4 Loss function

Knowledge Distillation

Knowledge distillation enhances the performance of small models by leveraging knowledge transferred from large models [89]. Using knowledge distillation, the student model attains remarkable accuracy without compromising efficiency. Inspired by this, we utilize KL divergence to force the student classifier T_{ecac} to mimic the teacher classifier S_{ecac}, where the loss is defined as:

$$L_{KL} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\tilde{O}_t^i \cdot log(\tilde{O}_s^i) \tag{4.11}$$

where N denotes the total number of pixels in the image and C represents the number of classes. Following previous work [43], we incorporate entropy H into the KL loss to capture the distributional information of each pixel:

$$L_{KL} = -\frac{1}{C}\sum_{k=1}^{C}\frac{\sum_{i=1}^{N}\sum_{c=1}^{C} B_k^i H^i \tilde{O}_t^{\,i} \cdot log(\tilde{O}_s^i)}{\sum_{i=1}^{N} B_k^i H^i} \tag{4.12}$$

where B_k is a binary mask representing the presence of the k-th class.

Re-weighting Cross-Entropy Loss

To tackle the issue of imbalanced distribution, we apply a re-weighting strategy [90] to the cross-entropy loss. Specifically, we propose a calibration strategy that aligns the model's predictions with a reference class distribution designed to encourage balanced predictions. Then, the re-weighting loss can be written as:

$$L_{rce} = w_c \cdot L_{ce} \tag{4.13}$$

where w_c is the class weight and L_{ce} is cross-entropy loss. We formulate the reference weights as a distribution derived from the empirical class frequencies $\mathbf{f} = [f_1, \ldots, f_K]$, calculated using

the training set.

$$w_c = \frac{(1/f_c)^\rho}{\sum_{k=1}^{K}(1/f_k)^\rho}, \quad \forall c \in \mathcal{C} \tag{4.14}$$

where ρ serves as a scaling hyper-parameter, offering greater flexibility in representing the class prior.

Intra-Class Variance

Increasing intra-class compactness can lead to a wider margin between classes, thereby yielding features that are easier to distinguish. To reduce intra-class variance, we use the map of output (O_s, O_t) similarity between pixels and their class centers. First, we compute the class center for each class by taking the mean of the output (O_s, O_t) for all pixels in the class. Then, we calculate the cosine similarity between each pixel and its corresponding class center. The intra-class variance of student and teacher is defined as:

$$D_{intraS} = \cos\left(\tilde{O}_s, CenterSeg_s\right) \tag{4.15}$$

$$D_{intraT} = \cos\left(\tilde{O}_t, CenterSeg_t\right) \tag{4.16}$$

where Seg_s and Seg_t denote the outputs of the student and teacher classifier, and *cos* denotes the cosine similarity function. $CenterSeg_s$ and $CenterSeg_t$ stand for the class center of output as follows:

$$CenterSeg_s = \frac{1}{Nc}\sum_{i=1}^{N_c}\tilde{O}_{s,i} \tag{4.17}$$

$$CenterSeg_t = \frac{1}{Nc}\sum_{i=1}^{N_c}\tilde{O}_{t,i} \tag{4.18}$$

where N_c represents the number of pixels sharing the same label. To address large discrepancies between the intra-class variance of the student and teacher outputs, we employ a straightforward yet effective Mean Squared Error loss. The intra-class variance loss is formulated as:

$$L_{IV} = f_{mse}(D_{intraS}, D_{intraT}) \tag{4.19}$$

Apart from L_{KL} and L_{iv}, cross-entropy loss is applied to Seg_s and Seg_t, defined as L_{ceS} and L_{ceT}.

Memory Bank Updating

We propose a memory loss for updating the memory bank:

$$L_{\mathcal{M}} = \frac{1}{hw} \sum_{k=1}^{hw} -Y_m[k] \cdot log(\text{softmax}(C_m)) \tag{4.20}$$

where Y_m is the ground-truth map. To ensure robust initialization of the memory bank, especially for rare or initially absent classes, we delay the computation of $L_{\mathcal{M}}$ until the first **1,000 training iterations** are completed. This design choice is motivated by two key factors:

- **Feature Stability:** In early training stages, model parameters and feature representations are rapidly evolving. Delaying $L_{\mathcal{M}}$ allows the network to learn coarse yet stable features before optimizing the memory bank, avoiding premature updates based on noisy representations.
- **Class Coverage:** For long-tailed datasets, some classes may appear infrequently or not at all in initial batches. By postponing $L_{\mathcal{M}}$, the memory bank accumulates sufficient samples for all classes through multiple updates (Eq. (4.2)), ensuring balanced feature initialization.

Hence, the overall loss is defined as:

$$L = L_{rceS} + L_{rceT} + L_{\mathcal{M}} + \alpha L_{KL} + \beta L_{iv} \tag{4.21}$$

where α is set to 1 and β is set to 50. $L_{\mathcal{M}}$ is activated only after 1,000 iterations.

4.2.5 Integration with other methods

As shown in Figure 4.2, ECAC seamlessly integrates with Existing Segmentation Models, including FCN [4], PSPNet [19], DeeplabV3+ [18], and UperNet [69], achieving significant performance improvements. While these methods emphasize image-level context aggregation, ECAC introduces dataset-level contextual information, further enhancing segmentation performance.

To integrate our module into existing methods, we compute class centers derived from the feature outputs of the current image, capturing the per-image category-specific information. These image-specific class centers are computed for both the student model and the teacher model. The student's class centers represent the current image's learned feature representations for each class, while the teacher's class centers, informed by ground truth, provide a more

accurate reference for those classes. Simultaneously, a memory bank maintains feature centers that aggregate category-specific information across the entire dataset, forming a global (dataset-level) contextual representation. By combining the image-specific class centers with the memory bank's feature centers, we construct a robust dataset-level classifier that leverages both local (image-level) and global (dataset-level) contextual information. The teacher-ECAC, guided by ground truth, facilitates knowledge distillation, enabling the student model to refine its class centers by learning from the teacher's more precise representations. This teacher-student framework enhances the student's ability to capture nuanced class-specific features, improving overall classification performance across diverse datasets.

4.3 Experiments

This section begins by introducing the benchmark datasets and outlining the implementation details of our method. Subsequently, we present the ECAC results across various semantic segmentation datasets. Finally, we conduct ablation studies to evaluate the influence of key components in ECAC.

4.3.1 Datasets

ADE20K. ADE20K [1] is a challenging segmentation dataset, which contains more than 20,000 images. The dataset comprises 150 semantic categories, *i.e.*, water, buildings, vegetation, and discrete objects such as animals, bicycles, and furniture. It is divided into 20k/2k/3k images for training, validation, and testing.

COCO-Stuff10k. The COCO-Stuff10k [70] dataset includes 171 semantic classes, with 81 thing classes and 91 stuff classes. It contains 9k training images and 1k testing images.

PASCAL-Context. The PASCAL-Context [71] is a widely used semantic segmentation dataset with 10,100 images, including 4,996 for training and 5,104 for testing, labeled across 59 categories.

4.3.2 Implementation Details

Model Settings. We conduct all the experiments on PyTorch. We employ ImageNet pre-trained ResNet and Swin-Transformer as backbones for our method. Each model is trained on

dual NVIDIA L40 GPUs with 48GB memory.

Training Settings. We optimize the network using stochastic gradient descent (SGD) with a momentum of 0.9, a polynomial learning rate scheduler defined as $(1 - \frac{iter}{iter_{max}})^{0.9}$, and synchronized batch normalization (SyncBN). During testing, we employ multi-scale ratios from 0.5 to 1.75 and incorporate data augmentation techniques such as flipping and random cropping. Detailed settings are provided for each benchmark.

- ADE20K: The initial learning rate is set to 0.01, with a crop size of 512 × 512 and a weight decay of 0.0005. By default, training runs for 130 epochs with a batch size of 16.
- COCO-Stuff10K: The initial learning rate is 0.001, crop size is 512 × 512, and weight decay is 0.0001. Unless specified, we train for 110 epochs with a batch size of 16.
- PASCAL-Context: We use an initial learning rate of 0.001, crop size of 480 × 480, and a weight decay of 0.0001. If not specified, training consists of 110 epochs with a batch size of 16.

4.3.3 Comparisons with State-of-the-art Methods

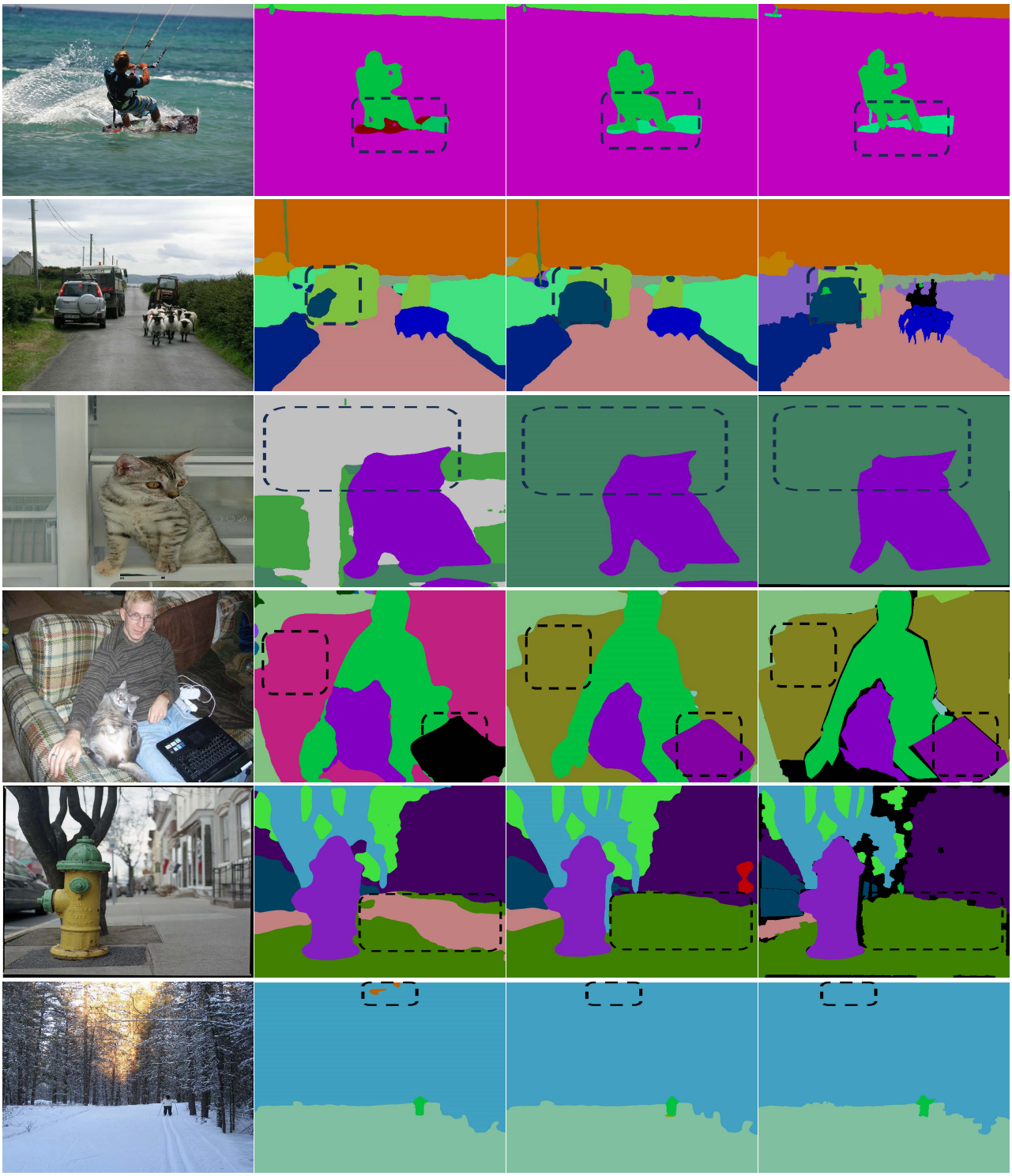
Table 4.1 summarizes the performance of state-of-the-art image segmentation methods on the ADE20K validation set, COCO-Stuff10K test set, and Pascal-Context test set, with mean Intersection over Union (mIoU) as the evaluation metric. The table is divided into two main sections based on backbone architecture: CNN-based and Transformer-based methods. Each section lists methods with their respective backbones (*e.g.*, ResNet-101 [74], Swin-Large [75]) and mIoU scores across the three datasets. We selected DeepLabV3plus [18] and UperNet [69] as the base architectures for integrating our proposed ECAC module due to their established effectiveness and complementary strengths in image segmentation tasks. The CNN-based section starts with PSPNet and progresses to our DeepLabV3plus + ECAC. In contrast, the Transformer-based section begins with StructToken [94] and culminates in our UperNet + ECAC, showcasing the impact of our approach across different architectures. Within the Transformer section, methods using the Swin-Large backbone are grouped to highlight performance trends, including GSS-FT [83], TSG [85], CoT [98], and our UperNet + ECAC. Notably, our UperNet + ECAC with Swin-Large achieves the highest scores among Transformer-based methods, with mIoU values of 54.70% on ADE20K, 50.49% on COCO-Stuff10K, and 64.50% on Pascal-Context.

	Method	Backbone	mIoU (%)		
			ADE20K	COCO-Stuff10k	Pascal-Context
CNN backbones	PSPNet [19]	ResNet-101	43.29	38.86	53.50
	CPNet [32]	ResNet-101	46.27	-	53.90
	DeeplabV3plus [18]	ResNet-101	46.35	39.00	54.67
	CSFLM [91]	ResNet-101	46.65	41.30	55.90
	PCAA [81]	ResNet-101	46.74	-	55.60
	ISNet [12]	ResNet-101	47.31	41.60	-
	CPT-M [92]	ResNet-101	47.00	-	56.30
	CAA [93]	ResNet-101	47.45	41.00	55.57
	DeeplabV3plus + ECAC (*ours*)	ResNet-50	47.67	40.93	54.99
		ResNet-101	**48.93**	**42.53**	**57.44**
Transformer backbones	StructToken [94]	ViT-Base	51.85	-	-
	SETR-MLA [95]	ViT-Large	50.28	-	55.83
	MCIBI [37]	ViT-Large	50.80	44.89	-
	SegNeXt [96]	MSCAN-L	51.28	44.77	59.49
	UperNet [86]	InternImage-B	51.30	45.30	61.34
	SegNeXt + ECAC (*ours*)	MSCAN-L [96]	51.74	45.72	60.19
	UperNet + ECAC (*ours*)	InternImage-B [86]	**52.21**	**46.32**	**61.75**
	SemCL [97]	Swin-Base	48.68	-	-
	UperNet+CAC [43]	Swin-Base	53.52	-	62.16
	GSS-FT-W [83]	Swin-Large	48.54	-	-
	CoT [98]	Swin-Large	54.16	-	57.21
	TSG [85]	Swin-Large	54.20	-	63.30
	UperNet + SSA [44]	Swin-Large	54.54	50.25	64.25
	UperNet + ECAC (*ours*)	Swin-Base	53.69	48.19	63.64
		Swin-Large	**54.70**	**50.49**	**64.50**

Table 4.1: **Comparisons with state-of-the-art methods are conducted using multi-scale and flipped testing on the ADE20K val set, COCO-Stuff10K test set, and Pascal-Context test set, with mIoU as the evaluation metric.**

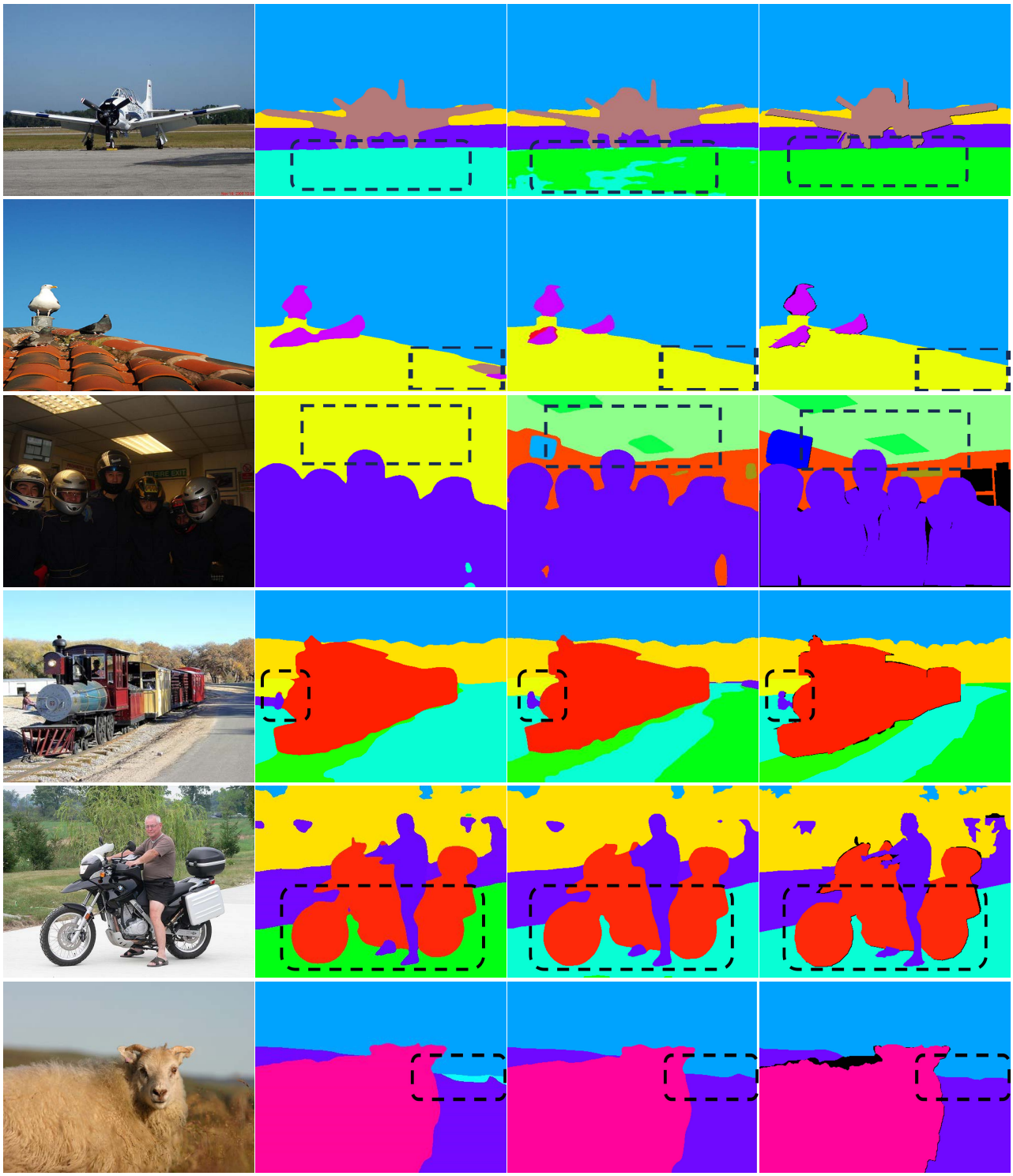
(a) Image (b) DLV3plus (c)DLV3plus+ECAC (d) Ground truth

Figure 4.4: **Qualitative comparisons on ADE20K val [1]. The dotted boxes emphasize areas with significant improvements achieved through the application of the proposed ECAC. DLV3 is the abbreviation for DeeplabV3plus [18].**



(a) Image (b) DLV3plus (c)DLV3plus+ECAC (d) Ground truth

Figure 4.5: **Qualitative comparisons on COCO-Stuff10K test [70]. The dotted boxes emphasize areas with significant improvements achieved through the application of the proposed ECAC. DLV3 is the abbreviation for DeeplabV3plus [18].**



(a) Image (b) DLV3plus (c)DLV3plus+ECAC (d) Ground truth

Figure 4.6: **Qualitative comparisons on Pascal Context test [71]. The dotted boxes emphasize areas with significant improvements achieved through the application of the proposed ECAC. DLV3 is the abbreviation for DeeplabV3plus [18].**

Baseline	ECAC	L_{rce}	L_{mem}	L_{KL}	L_{mse}	mIoU(%)
✓						45.76
✓		✓				46.56
✓	✓	✓				49.10
✓	✓	✓	✓			50.21
✓	✓	✓	✓	✓		50.43
✓	✓	✓	✓	✓	✓	**50.68**

Table 4.2: **Ablation study about various loss combinations on Pascal-Context *val*.** L_{rce}**,** L_{KL}**,** L_{mem} **and** L_{mse} **denote re-weighting cross-entropy loss, Kullback-Leibler (KL) divergence, memory loss and Mean Squared (L2) loss.**

Overall, the integration of our proposed ECAC module consistently enhanced the performance of both CNN and Transformer backbones, with the most notable gains observed in the Pascal-Context dataset, where our methods achieved the highest mIoU scores. Qualitative comparisons are presented in Figure 4.4, 4.5 and 4.6.

4.3.4 Ablation Study

Here, we present ablation studies on the Pascal-Context to investigate the effectiveness of our proposed method. ResNet-50 is selected as the backbone for the FCNs [4] baseline due to its lower computational cost compared to ResNet-101, ViT, and Swin-Transformer. Unless otherwise specified, all experiments in this section are performed using **single-scale testing** for simplicity and fair comparison, which differs from the Table 4.1.

Ablation Study on Loss function

Our ECAC framework incorporates a combination of re-weighting cross-entropy loss, memory loss, KL divergence loss, and MSE loss to enhance segmentation performance. To investigate the contribution of each loss component, we conduct an ablation study, and the results are summarized in Table 4.2.

We achieve 0.8% mIoU improvement using the re-weighting cross-entropy loss. Utilizing ECAC without memory loss, we achieve 49.10% mIoU. When memory loss is included, the performance improves significantly, reaching a mIoU of 50.21%. Further, adding KL loss alongside memory loss leads to a mIoU of 50.43%. The KL divergence loss helps align the distributions of the teacher and student models, enabling more efficient knowledge transfer and improving the

Loss	mIoU(%)
$L_{rceS} + L_{rceT} + L_M$	50.21
$L_{rceS} + L_{rceT} + L_M + 0.1 \cdot L_{KL}$	50.22
$L_{rceS} + L_{rceT} + L_M + 1 \cdot L_{KL}$	**50.43**
$L_{rceS} + L_{rceT} + L_M + 10 \cdot L_{KL}$	50.30
$L_{rceS} + L_{rceT} + L_M + L_{KL}$	50.43
$L_{rceS} + L_{rceT} + L_M + L_{KL} + 50 \cdot L_{iv}$	**50.68**
$L_{rceS} + L_{rceT} + L_M + L_{KL} + 100 \cdot L_{iv}$	50.58
$L_{rceS} + L_{rceT} + L_M + L_{KL} + 200 \cdot L_{iv}$	50.45

Table 4.3: **Ablation study about KL loss weight α and Mean Squared loss weight β on Pascal-Context *val*.**

segmentation accuracy. This result highlights the complementary nature of memory loss and KL loss in refining the learned representations. Finally, combining memory loss, KL loss, and MSE loss achieves the highest mIoU of 50.68%. The combination of these three losses allows ECAC to leverage both inter-class relationships and intra-class consistency effectively.

We additionally present experiments on Pascal-Context to verify the appropriate loss weight for the KL loss α and MSE loss β in Table 4.3. The results show that the best performance, with a mIoU of 50.68%, is achieved when $\alpha = 1$ and $\beta = 50$.

Update strategy of memory bank

Table 4.4 shows the results of different update strategies on the memory bank. Our memory bank is updated by employing ground-truth masks calculated class center. Compared with the memory bank update strategy in MCIBI [37], which uses a simple cosine similarity strategy, our method improves the mIoU from 50.10% to 50.68%. Therefore, we adopt the masks calculated class center for updating the memory bank.

The improvement demonstrates that using ground-truth mask-calculated class centers allows for a more precise and consistent representation of each class within the memory bank. Unlike the cosine similarity strategy, which relies solely on feature alignment, our approach benefits from directly incorporating spatial and semantic information from the ground-truth, leading to enhanced model performance.

Method	Backbone	mIoU(%)
Baseline	ResNet-50	45.76
Baseline + ECAC(*cosine*)	ResNet-50	50.10
Baseline + ECAC(*ours*)	ResNet-50	**50.68**

Table 4.4: **Ablation study on the test set of Pascal-Context *val* about the update strategy of the memory bank.**

Method	Backbone	mIoU(%)
DeeplabV3plus	ResNet-50	50.73
DeeplabV3plus + ECAC	ResNet-50	53.06
DeeplabV3plus + ECAC + Calibration	ResNet-50	**53.56**

Table 4.5: **Ablation study on the test set of Pascal-Context *val* about the output calibration.**

Ablation about output calibration

We also conduct an ablation study on output calibration in Table 4.5. As we mentioned in section 4.2.2, the calibration strategy aligns the output with a reference class distribution, ensuring more balanced and accurate predictions. We gain 0.5% mIoU improvement compared with our ECAC.

Computational Complexity

Table 4.6 presents the efficiency of ECAC integrated with existing segmentation frameworks on ADE20K using a ResNet-50 backbone. The results are derived from an input size $512 \times 64 \times 64$ (an $8\times$ down-sampling of 512×512). Compared with MCIBI [37], we achieve a significant improvement in accuracy with only a marginal increase in the number of parameters. For instance, incorporating ECAC with PSPNet, we achieve a mIoU of 45.47%, with 1.05M extra parameters and 1.13G more FLOPs. MICBI [37] achieves 43.77% mIoU but increases 13.19M parameters and 54.04 GFLOPs.

Method	Backbone	Params(M)	FLOPs(G)	mIoU (%)
FCN	ResNet-50	11.87	47.72	36.10
+ MCIBI	ResNet-50	16.41(↑ 4.54)	67.22(↑ 19.50)	42.84(↑ 6.74)
+ MCIBI++ [38]	ResNet-50	15.45(↑ 3.58)	64.44(↑ 16.72)	43.39(↑ 7.29)
+ ECAC	ResNet-50	**12.90**(↑ 1.03)	**49.17**(↑ 1.45)	**43.12**(↑ 7.02)
PSPNet	ResNet-50	23.14	77.71	42.48
+ MCIBI	ResNet-50	36.33(↑ 13.19)	131.75(↑ 54.04)	43.77(↑ 1.29)
+ MCIBI++ [38]	ResNet-50	33.99(↑ 10.85)	122.16(↑ 44.45)	43.89(↑ 1.41)
+ ECAC	ResNet-50	**24.19**(↑ 1.05)	**78.84**(↑ 1.13)	**45.47**(↑ 2.99)
DeeplabV3plus	ResNet-50	17.74	65.97	43.95
+ MCIBI	ResNet-50	29.68(↑ 11.94)	117.34(↑ 51.37)	44.34(↑ 0.39)
+ MCIBI++ [38]	ResNet-50	28.59(↑ 10.85)	110.42(↑ 44.45)	44.85(↑ 0.9)
+ ECAC	ResNet-50	**18.76**(↑ 1.02)	**67.22**(↑ 1.25)	**46.32**(↑ 2.37)

Table 4.6: **Comparison of inference computational complexity and accuracy on ADE20K *val.***

Integration with Various Semantic Segmentation Frameworks

ECAC fits effortlessly into current segmentation frameworks. To validate the performance and stability of our strategy, we integrate ECAC into four popular segmentation frameworks: FCN [4], PSPNet [19], UperNet [69] and DeepLabVplus [18]. Table 4.7 illustrates the comparative performance analysis on three segmentation datasets: ADE20k, COCO-Stuff10K, and Pascal-Context. To ensure fairness, we perform single-scale testing. Integrating ECAC into the vanilla segmentation framework (e.g., FCN) significantly improves mIoU by 7.02% on ADE20K, 2.58% on COCO-Stuff, and 4.92% on Pascal-Context. With the stronger baseline DeeplabV3plus, ECAC boosts mIoU from 43.95% to 46.32% on ADE20K, 36.92% to 40.19% on COCO-Stuff, and 50.73% to 53.56% on Pascal-Context. These results highlight the flexibility and efficacy of ECAC.

Per-Class Analysis

We analyze the performance of DeepLabV3plus + ECAC on the ADE20k dataset to assess its impact on class imbalance, as shown in Table 4.8. To clearly reflect the long-tailed distribution, we include the *Ratio* of each category in the table, where Ratio denotes the average percentage of pixels belonging to that category in the training set. Based on these Ratios, we

Method	Backbone	ADE20K	COCO-Stuff	Pascal-Context
FCN [4]	ResNet-50	36.10	34.10	45.76
FCN + ECAC	ResNet-50	43.12 (↑ 7.02)	36.68 (↑ 2.58)	50.68 (↑ 4.92)
UperNet [69]	ResNet-50	42.05	35.80	48.90
UperNet + ECAC	ResNet-50	45.29 (↑ 1.76)	37.61 (↑ 1.81)	53.13 (↑ 4.23)
PSPNet [19]	ResNet-50	42.48	36.33	50.21
PSPNet + ECAC	ResNet-50	45.47 (↑ 2.99)	38.32 (↑ 1.99)	53.47 (↑ 3.26)
DeeplabV3plus [18]	ResNet-50	43.95	36.92	50.73
DeeplabV3plus + ECAC	ResNet-50	46.32 (↑ 2.37)	40.19(↑ 3.27)	53.56 (↑ 2.83)

Table 4.7: **The performance of ECAC's integration into mainstream frameworks on various benchmarks. We adopt single-scale testing for the sake of convenience and simplicity, which is different from Table 4.1.**

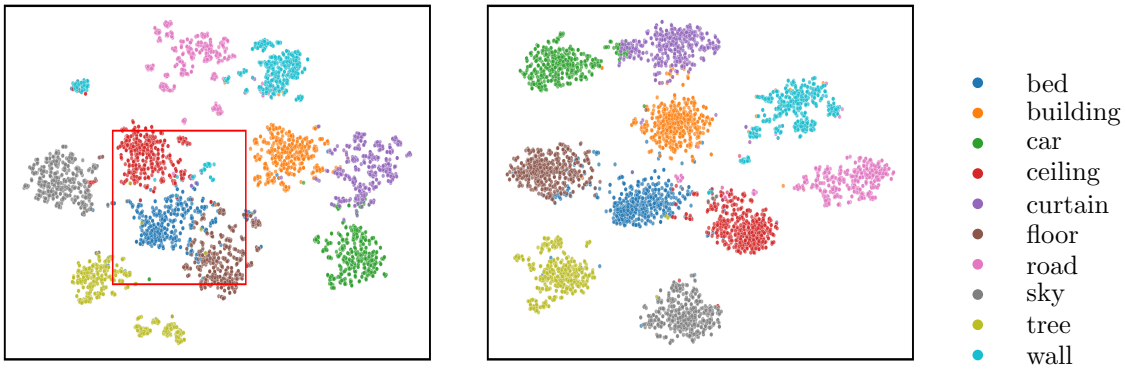
divide the categories into three representative groups: the 10 major classes (highest Ratios), 10 moderate classes (medium Ratios), and 10 minor classes (lowest Ratios). This grouping allows us to evaluate the model across head, middle, and tail classes. For the major categories, DeepLabV3plus [18] achieves an mIoU of 78.18%, which improves to 79.02% with ECAC. Per-class results reveal significant improvements in categories like "windowpane" (60.06% to 61.73%) and "grass" (65.05% to 69.79%), though dominant classes such as "sky" show a slight decline (94.09% to 93.99%). For moderate categories, the mIoU increases from 46.06% to 51.59% with ECAC, with notable gains in "arcade machine" (29.48% to 47.38%) and "palm" (50.87% to 53.74%). For minor categories, the mIoU improves substantially from 26.75% to 31.91%, with significant gains in "shower" (0.0% to 12.20%) and "pier" (25.11% to 32.58%), though "bulletin board" declines (35.98% to 27.58%). Per-class analysis indicates that ECAC enhances feature representation for minor classes by leveraging contextual cues, as seen in the improved performance for rare categories. However, inconsistencies in certain minor classes, such as "bulletin board," suggest potential overfitting or sensitivity to specific class characteristics. Overall, ECAC improves segmentation performance for both moderate and minor classes.

Visualization of Learned Features

To further examine the quality of the learned feature representations, we visualize the feature distributions of 10 randomly selected ADE20K classes using t-SNE, as shown in Figure 4.7. In

Method (Major Class)	wall(15.8%)	building(10.7%)	sky(8.8%)	floor(6.2%)	tree(4.8%)	ceiling(4.5%)	road(4.0%)	bed(2.3%)	windowpane(2.0%)	grass(1.8%)	mIoU(%)
DeeplabV3plus [18]	75.53	82.18	94.09	79.11	73.61	82.63	82.0	87.52	60.06	65.05	78.18
DeeplabV3plus + ECAC	76.02	82.21	93.99	79.43	73.81	82.79	82.68	87.79	61.73	69.79	79.02
Method (Moderate Class)	bench(0.13%)	countertop(0.12%)	stove(0.12%)	palm(0.12%)	kitchen(0.12%) island(0.12%)	computer(0.11%)	swivel chair(0.1%)	boat(0.09%)	bar(0.09%)	arcade machine(0.09%)	mIoU(%)
DeeplabV3plus [18]	38.59	54.02	74.64	50.87	43.74	55.5	46.83	42.83	24.12	29.48	46.06
DeeplabV3plus + ECAC	40.44	59.91	75.84	49.31	42.88	55.46	41.73	76.45	26.49	47.38	51.59
Method (Minor Class)	pier(0.03%)	crt screen(0.03%)	plate(0.03%)	monitor(0.03%)	bulletin board(0.03%)	shower(0.03%)	radiator(0.03%)	glass(0.02%)	clock(0.02%)	flag(0.02%)	mIoU(%)
DeeplabV3plus [18]	25.11	0.33	42.83	55.34	35.98	0.0	41.73	9.42	22.7	34.1	26.75
DeeplabV3plus + ECAC	32.58	0.37	47.32	73.60	27.58	12.20	46.00	16.74	29.81	32.90	31.91

Table 4.8: **Comparison of each category in ADE20k. We include the *Ratio* of each category, which represents the average percentage of pixels belonging to that class in the training set. Based on these Ratios, we select 10 major (high-Ratio), 10 moderate (medium-Ratio), and 10 minor (low-Ratio) categories to provide a representative evaluation across head, middle, and tail classes.**



(a) DeeplabV3plus (b) DeeplabV3plus+ECAC

Figure 4.7: **Visualization of feature distributions learned with DeeplabV3plus [18] (left) and our ECAC (right).**

the left plot (a), representing the baseline DeepLabV3plus [18], the clusters appear relatively loose and partially intermingled. Such dispersion and class-to-class intrusion indicate that the baseline model struggles to form compact and well-separated feature groups, thereby limiting its ability to discriminate between different semantic categories.

In contrast, the right plot (b) presents the feature space produced by our DeepLabV3plus+ECAC. The clusters become noticeably tighter and more clearly separated, demonstrating that ECAC effectively enhances the model's capability to organize features into more coherent and discriminative structures. This improvement confirms that ECAC reduces class ambiguity and yields more reliable feature representations.

Nevertheless, a small amount of overlap can still be observed, suggesting that while ECAC significantly improves cluster compactness and class separability, there remains potential for further refinement toward achieving fully optimal feature distinction across the dataset.

4.4 Conclusion

This chapter tackles the critical problem of improving pixel-wise classification accuracy in semantic segmentation, particularly under the challenge of class imbalance, which often leads to poor performance on rare or underrepresented categories. While existing methods tend to focus on local features within a single image, they often fail to leverage the global statistical information across the dataset or integrate contextual knowledge in a class-aware manner. Our motivation lies in addressing this limitation by effectively incorporating both dataset-level and

image-specific contextual cues to guide semantic segmentation.

To this end, we introduce a novel module named Extended Context-Aware Classifier (ECAC). ECAC enhances segmentation frameworks through three core components: (1) a memory bank that maintains dataset-level class feature centers, providing a global semantic prior; (2) dynamic extraction of image-specific class centers from each input sample, ensuring the model adapts to image-specific context; (3) a teacher-student framework, in which a ground truth-guided teacher ECAC supervises the student model, refining the class center estimation through knowledge distillation. This design ensures the learned class centers are not only statistically robust but also semantically aligned with the ground truth.

The strength of ECAC lies in its context fusion strategy, which merges global consistency and local adaptability. The memory bank offers strong regularization and stabilizes class representations across training, while the teacher-student mechanism further enhances representation quality by aligning the student's predictions with the teacher's more accurate, ground-truth-informed outputs. Notably, ECAC is computationally efficient and can be seamlessly integrated into existing semantic segmentation architectures as a plug-in module, with minimal inference overhead.

Extensive experiments on multiple challenging segmentation benchmarks demonstrate the significant gains in segmentation accuracy, especially for long-tail and minority classes. ECAC not only boosts overall performance but also helps reduce misclassification between semantically similar categories—an issue that plagues conventional segmentation models.

Chapter 5

Open Vocabulary Image Semantic Segmentation

In the previous chapters, we have addressed the problem of semantic segmentation under the closed-set assumption, where all categories present in the test images are known and fixed. To tackle the challenges arising from complex intra-class variation and class imbalance, two key modules were proposed: the Class-Aware Affinity (CAA) module, which captures both pixel-to-pixel and pixel-to-class relationships for better boundary discrimination; and the Extended Context-Aware Classifier (ECAC), which leverages both global and local contextual information to dynamically adjust classification, thereby alleviating the impact of class distribution imbalance.

However, real-world scenarios often go beyond the limitations of closed-set assumptions. In practice, models frequently encounter novel or unseen categories that were not included during training. To address this more challenging and practical setting, the focus of this chapter shifts to open-vocabulary semantic segmentation. This task not only demands dense pixel-level predictions but also requires the ability to generalize to a broad set of categories defined by arbitrary text inputs, often outside the predefined label space.

Beginning with this chapter, we explore methodologies for bridging the gap between vision and language, aiming to improve the alignment between visual features and textual representations. By leveraging pre-trained vision-language models and integrating contextual cues from both modalities, our goal is to enable more flexible and generalizable segmentation systems that can

perform well even in open-set scenarios.

5.1 Introduction

Conventional semantic segmentation [18], [25], [30], [37], [61], [93] is limited to the pre-defined closed-set categories, making it difficult for them to recognize new or unseen categories during inference. Recently, open-vocabulary semantic segmentation (OVSS) [47], [48], [51] aims to classify each pixel in an image into its most relevant class from a potentially unlimited set of semantic categories, where the model identifies and labels pixels based on arbitrary or descriptive text input. Such kind of segmentation capability closely aligns with real-world applications.

Existing strategies to improve alignment fall into three main categories: (1) **Refine region-level visual-text alignment.** Methods like [46]–[49] employ a category-agnostic mask generator to derive region-level representations that closely resemble image-level features. However, these methods primarily achieve region-level alignment, incurring substantial computational costs and inefficient memory usage. (2) **Refine pixel-level visual-text alignment.** Cutting-edge works [53], [99], [100] propose to leverage the Segment Anything Model (SAM) [101] or feature aggregation to enhance rich pixel-level visual representation, thereby compensating for the spatial deficiencies of CLIP features. These methods effectively complement the spatial information deficiencies inherent in CLIP features, which are trained via image-level contrastive learning and thus excel at global context but struggle with localized pixel semantics. Yet, these efforts often overlook a critical component: the quality of textual representations. Simple text prompts, such as '`a photo of a [class]`,' lack the semantic richness needed to disambiguate complex objects, like distinguishing a flower from a tulip based on petal structure or color. Moreover, CLIP's text encoder can struggle with lexical ambiguities, limiting its discriminative power for fine-grained segmentation tasks. (3) **Attribute-level visual-text alignment.** Emerging studies have shown that enhancing the quality of text templates by incorporating attributes [102], [103] or class-specific details [57], [104] can improve CLIP's performance. However, these efforts are typically designed for general contexts and seldom customized to meet the specific requirements of open-vocabulary semantic segmentation (OVSS).

Although existing methods have achieved promising performance, there are still some issues to be addressed: (1) **Text Prompts**: We argue that the quality of textual representations is equally important to achieving precise visual-text alignment in OVSS. Simplistic prompts

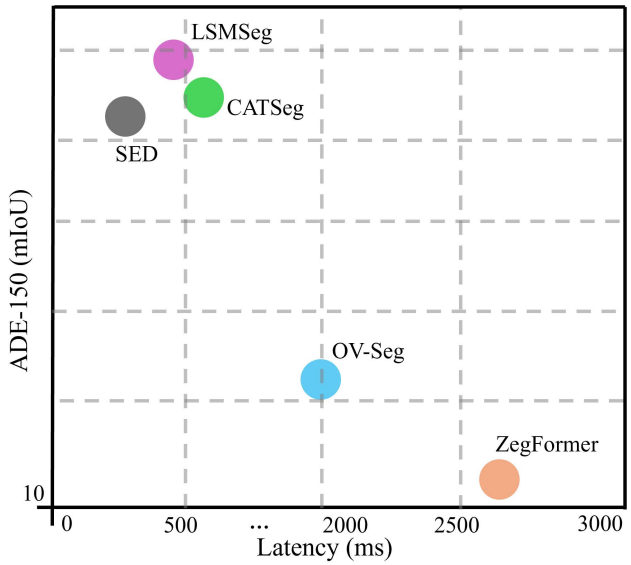


Figure 5.1: **Segmentation Performance and inference latency on PC-459. Our LSMSeg outperforms ZegFormer [49], OV-Seg [47], CATSeg [100], and SED [53], achieving a new-state-of-the-art mIoU of 19.7% while maintaining reduced latency.**

fall short in three key aspects: First, they lack the detailed semantic information required for fine-grained segmentation tasks, such as differentiating a flower species based on its intricate petal structure and color. Second, the discriminative power of generated text embeddings depends on the CLIP text encoder, which may fail to distinguish between meanings if there are lexical ambiguities. For instance, the word '`bat`' could refer to either '`a flying mammal`' or '`a piece of sports equipment used in baseball`,' and simply encoding the class name will not be enough to differentiate between these two concepts. Third, they fail to leverage multi-modal information, which is crucial for capturing the nuances of complex categories, thereby limiting the model's adaptability to diverse and fine-grained visual contexts. (2) **Visual Spatial Information**: Prior works [105], [106] have demonstrated that intermediate layers of CLIP effectively capture dense features. However, in the context of open-vocabulary semantic segmentation (OVSS), these dense features often fall short in delivering the fine-grained, spatially precise information required for accurate pixel-level classification. The primary limitation stems from CLIP's training paradigm, which emphasizes image-level contrastive learning and prioritizes global context over localized semantics. As a result, the extracted features lack the spatial granularity necessary to distinguish intricate object boundaries or small category differences in complex scenes. This limitation is especially noticeable when segmenting unseen classes, where accurate alignment of visual and textual data is vital. In this work, we propose

LSMSeg, a novel framework for OVSS that jointly considers text prompts and visual spatial information. Our method consists of four parts: (a) *Text Prompt Generation*. We leverage GPT-4 to generate a set of candidate attributes, which are then used to prompt GPT-4 for detailed sentence descriptions tailored to each category. (b) *Visual Feature Fusion*. We utilize a frozen Segment Anything Model (SAM) image encoder [101] to augment the spatial information in CLIP features, integrating them through a learnable weight fusion strategy to effectively combine CLIP and SAM visual embeddings. (c) *Category Filtering Module*. We introduce a category filtering module to eliminate irrelevant classes, refining the cost map and lowering computational complexity. By pruning low-relevance categories from the initial cost map, the module improves segmentation accuracy and efficiency. As illustrated in Figure 5.1, LSMSeg achieves a new state-of-the-art for both efficiency and accuracy. (d) *Feature Refinement Module*. We improve segmentation performance by refining features at spatial and class levels using a Swin-Transformer block and a linear Transformer block.

Our main contributions can be summarized as follows:

- We introduce LSMSeg, a pioneering framework that leverages large language models (LLMs) to create detailed, attribute-enriched text prompts, significantly improving text-visual alignment for OVSS.
- We propose a comprehensive approach combining optimized text prompts, visual feature fusion, category filtering, and feature refinement, enhancing both accuracy and efficiency in segmentation tasks.
- Extensive experiments across multiple benchmarks demonstrate that LSMSeg achieves state-of-the-art performance in open-vocabulary semantic segmentation.

5.2 Proposed Method

5.2.1 Preliminaries

Problem Definition. Open-vocabulary semantic segmentation aims to partition an image $I \in \mathbb{R}^{H \times W \times 3}$ into distinct semantic regions based on text descriptions, including classes that were not seen during training. In training, only pixel-level annotations of the seen classes C_{train} are used, with knowledge of their existence and quantity (*i.e.*, how many and what classes are

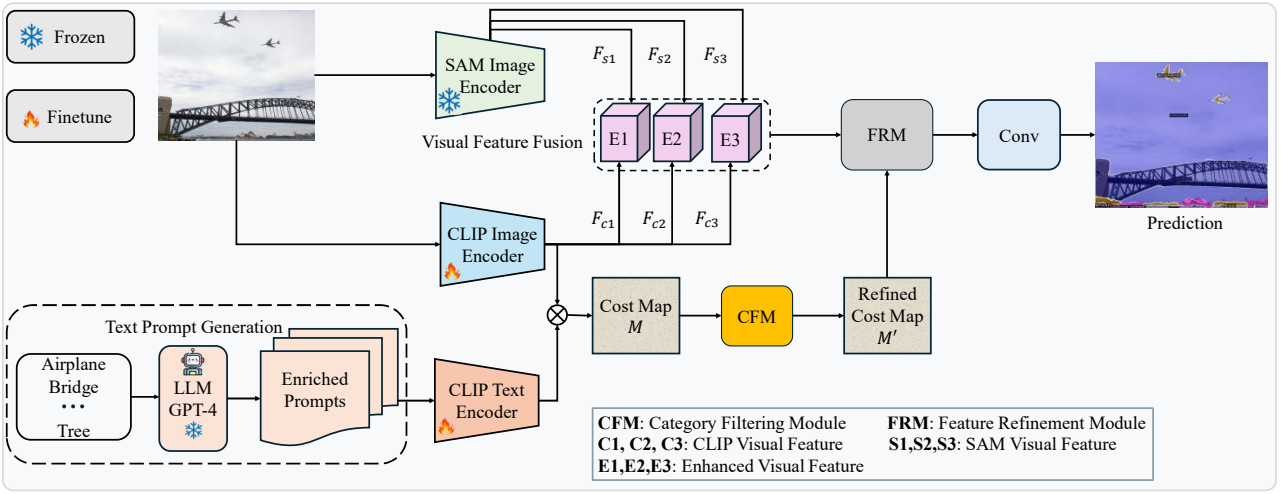


Figure 5.2: **Overall architecture of our proposed LSMSeg. We first utilize GPT-4 to generate enhanced text prompts. SAM visual feature is then used to compensate the lack of spatial information of CLIP visual feature through a visual feature fusion strategy. Next, we propose a category filter module to eliminate irrelevant classes, yielding a refined cost map and reducing computational complexity. Finally, we adopt feature refinement to enhance the filtered cost map at spatial and class level.**

present). Annotations for unseen categories are replaced with an "ignored" label. In an open-vocabulary setting, C_{test} may include new categories that were not encountered during training, meaning $C_{train} \neq C_{test}$. In inference, both seen and unseen classes need to be segmented.

5.2.2 Architecture Overview

Figure 5.2 illustrates the overall architecture of our proposed LSMSeg, comprising four main components: (a) The *Text Prompts Generation* leverages the GPT-4 model to first select appropriate attributes and then generate descriptive sentences based on those attributes, which are subsequently processed by the CLIP text encoder to obtain text features. (b) The *Visual Feature Fusion* integrates SAM features with CLIP visual features through a learnable weighted fusion strategy to enhance spatial information representation. (c) The *Category Filter Module* is proposed to reduce computational parameters and accelerate training by filtering irrelevant classes. (d) The *Feature Refinement Module* is introduced to effectively integrate and enhance both spatial-level and class-level feature information, enabling more precise and comprehensive feature representation for improved model performance.

5.2.3 Text Prompts Generation

Candidate Attribute Generation

To initiate the generation of comprehensive linguistic prompts, we leverage GPT-4 to identify candidate attributes that enhance text representations for open-vocabulary semantic segmentation (OVSS). As depicted in Figure 5.3, we begin by querying GPT-4 with: "What visual attributes are most relevant for generating descriptive text prompts to enhance pixel-level semantic segmentation?" In response, GPT-4 provides nine visual attributes-color, shape, size, texture, material, position or location, pattern, action or state, and contextual relationships-selected as they represent common visual characteristics applicable across diverse object categories. To substantiate the utility of these attributes, we conduct an additional experiment by providing GPT-4 with three diverse images (a cat, a boat, and a bridge) as shown in Figure 5.4. Each image was accompanied by the query: "Describe this image in five separate sentences." GPT-4 generated five sentences per image, resulting in 15 descriptions in total. These sentences, color-coded in Figure 5.4 to highlight the nine attributes, collectively encompass all proposed attributes, demonstrating their natural relevance to describing visual phenomena in segmentation tasks.

Enriched Text Prompt Generation

Building upon the candidate attributes from Section 5.2.3, we generate enriched text prompts tailored to specific categories. For each attribute, we query GPT-4 with: "Describe what a {class name} generally looks like in terms of attribute?" This transforms the attribute list into concrete descriptions. For example, for the class "cat" and the attribute "color," GPT-4 might generate: "A cat is often black, white, orange, or gray, sometimes with mixed fur colors."

This process is repeated across all nine attributes for each category, resulting in a set of attribute-rich text prompts. These prompts are designed to be more informative than the simplistic templates (*e.g.,* "a photo of a {class name}") used in prior OVSS methods, providing the CLIP text encoder with detailed semantic input. The output of this stage is a comprehensive pool of descriptive sentences, which we subsequently evaluate and refine.

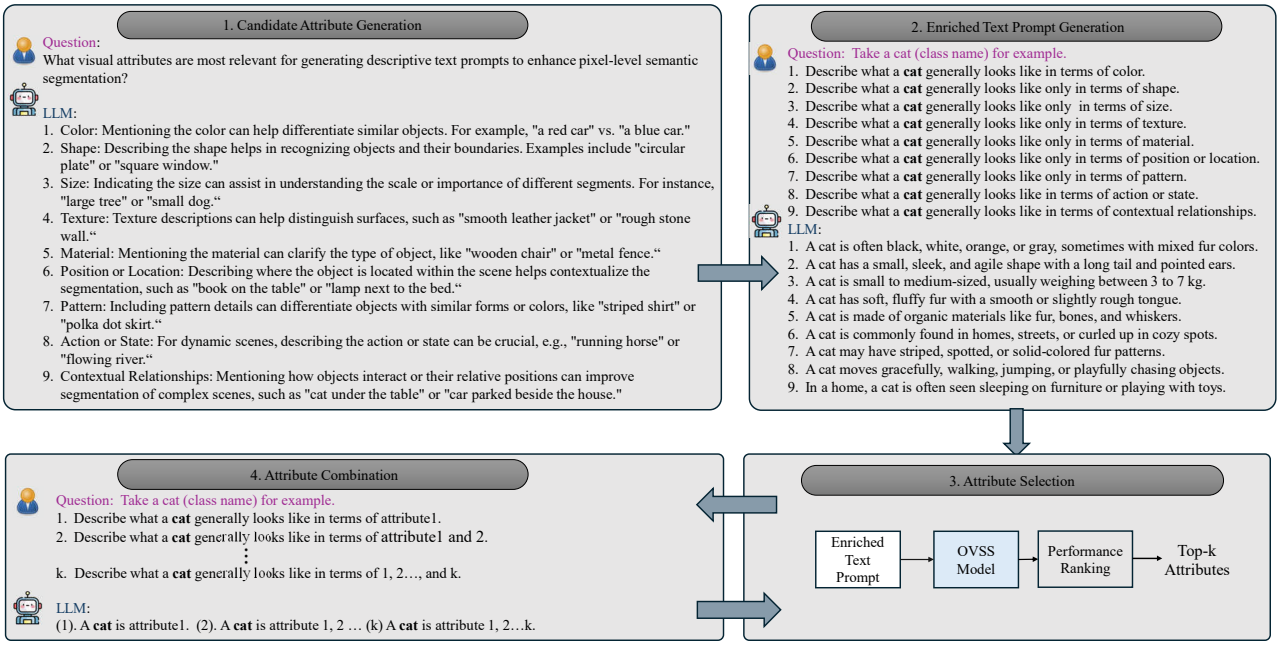


Figure 5.3: **The pipeline of generating a comprehensive linguistic prompt. (1) Candidate Attribute Generation, where we query large language models (LLMs) to identify visual attributes (*e.g.,* color, shape, size) most relevant for semantic segmentation; (2) Enriched Text Prompt Generation, where the identified attributes are used to guide LLMs in generating detailed, category-specific descriptions (*e.g.,* for the class "cat"); and (3) Attribute Selection, where the generated prompts are evaluated using the existing OVSS model to select the top-k attributes based on their performance rankings in terms of mean Intersection over Union (mIoU); (4) Attribute Combination, where the selected attributes are systematically combined (*e.g.,* size + shape + texture + color) to form comprehensive prompts, with further experiments conducted to determine the optimal combination for enhancing segmentation performance.**

Attribute Selection

To optimize the enriched text prompts, we evaluate each attribute's contribution to segmentation performance using OVSS model such as CAT-Seg [100] without its spatial and class aggregation modules, serving as a lightweight baseline to isolate the impact of text prompts. We input the attribute-specific prompts from Section 5.2.3 and measure performance via mIoU on the COCO-Stuff validation set, assessing both seen and unseen classes. The results, detailed in Section 5.3.4 (Table 5.3), reveal varying performance across the nine attributes. Attributes such as "color," "size," "shape," and "texture" consistently outperform others, achieving higher average mIoU scores. In contrast, attributes like "action or state" and "contextual relationships"

yield lower scores, suggesting they may introduce noise or lack sufficient discriminative power for pixel-level tasks. Thus, we select the top four attributes-color, size, shape, and texture-for their superior discriminative power and efficiency in enhancing visual-text alignment.

1. The image features an orange tabby cat with a slightly turned head, gazing at the camera with a calm yet curious expression.
2. Its fur is thick and well-groomed, displaying a rich, warm shade of orange with subtle stripes.
3. The cat's green eyes contrast with its fur, adding depth to its striking appearance.
4. Its ears are perked up, and its whiskers are long and slightly curved.
5. The blurred background suggests an outdoor setting, possibly a garden or patio, with natural lighting illuminating the cat's face.

1. The image features a traditional sailing vessel with two large, cream-colored sails fully unfurled.
2. The boat has a dark wooden hull and an open deck with people on board.
3. The water is calm with gentle ripples, and the sky is partly cloudy with patches of blue.
4. In the background, there is a shoreline with trees and buildings visible.
5. The vessel appears to be in motion, likely propelled by the wind filling its sails.

1. The image captures two airplanes flying in the sky above a large steel bridge.
2. The bridge is the Sydney Harbour Bridge, a well-known landmark in Australia.
3. Below the bridge, the Sydney Opera House is visible in the distance, showcasing its iconic white sail-like design.
4. The sky is overcast, with a grayish-white cloud cover dominating the background.
5. The perspective of the image emphasizes the scale of the bridge and the airplanes soaring high above it.

Different colors are used to indicate different attributes : color, shape, size, texture, material, position or location, pattern, action or state, and contextual relationship.

Figure 5.4: **Examples of GPT-4 Generated Descriptions to Validate Candidate Attributes for Semantic Segmentation. GPT-4 generates five sentences for each image, highlighting attributes (color-coded: color, shape, size, texture, material, position or location, pattern, action or state, contextual relationships) to show their relevance for segmentation.**

Attribute Combination

Having identified the top-performing attributes in Section 5.2.3, we explore their combinations to construct comprehensive linguistic prompts, aiming to maximize segmentation performance. We start with "size"-the highest individual performer as a baseline and incrementally add "shape," "texture," and "color." For each combination, we concatenate attribute-specific descriptions into a single prompt (*e.g.*, "A cat has a small, sleek, and agile shape with a long tail and pointed ears, is small to medium-sized, weighing between 3 to 7 kg, has soft, fluffy fur with a smooth or slightly rough tongue, and is often black, white, orange, or gray, sometimes with mixed fur colors.") and evaluate it using the CAT-Seg [100] setup from Section 5.2.3 on the COCO-Stuff validation set.

Results show progressive improvement: "size" (47.4%), "size + shape" (47.5%), "size + shape + texture" (47.6%), and "size + shape + texture + color" (47.8%). The full combination

outperforms others by providing a holistic description that captures both structural (size, shape) and surface (texture, color) properties, reducing ambiguity in text-visual alignment. Compared to the baseline's fixed prompt mIoU of 46.8% (Section 5.3.4), this optimal combination yields a 1.0% gain, underscoring the efficacy of attribute-enriched prompts. Further experiments confirm this quartet as the most effective balance of richness and complexity for OVSS tasks.

5.2.4 Visual Feature Fusion

The CLIP [45] model is trained by image-level contrastive learning and struggles with precisely localizing pixel-level visual features. Its embeddings focus on global visual context instead of the pixel-level semantics within the image. This can be a problem for segmentation, which requires understanding the local context of each pixel about its neighbors. To address this, we further propose leveraging a frozen SAM [101] image encoder to enhance and supplement the spatial information. As shown in Figure 5.2, we input image I into SAM image encoder and extract image features $F_s \in \mathbb{R}^{B\times H_s\times W_s\times D_s}$ from the last three global attention blocks. We also input I into the CLIP image encoder to extract image features $F_c \in \mathbb{R}^{B\times H_c\times W_c\times D_c}$.

For the visual feature fusion, we propose a learnable weight fusion strategy. We utilize a linear layer to align with the channel dimension F_s to that of F_c.

$$E = w \times F_c + (1 - w) \times F_s \tag{5.1}$$

where w is the learnable , and E is the fused visual feature. Then, we follow the work [100] to calculate the correlation between visual and text features.

5.2.5 Category Filtering Module (CFM)

Given an input image I, we obtain dense visual features $E \in \mathbb{R}^{B\times H\times W\times D}$ from CLIP and SAM image encoder, where B, H, W, and D represent the batch size, height, width and channel. Given a set of class names C, we leverage LLMs to generate comprehensive linguistic prompts. We obtain text embeddings $T \in \mathbb{R}^{B\times T\times D}$ by feeding these prompts into the CLIP text encoder, where T and D represent the number of class and channel. By computing the cosine similarity between the visual feature E and text embedding T, we derive the cost map embedding M as:

$$M_{(i,j,n)} = \frac{E_{(i,j)} \cdot T_n}{\|E_{(i,j)}\|\|T_n\|}. \tag{5.2}$$

where i, j denotes the spatial positions, and n indicates the text embedding index. Thus, the cost map embedding M has the dimension of $B \times T \times d \times H \times W$, where d is the channel of cost map embedding.

To reduce computational overhead and suppress noisy or uninformative text tokens, we apply a top-k token selection when the number of text tokens exceeds a predefined padding threshold q. Specifically, we compute the maximum correlation across spatial dimensions and visual prompts:

$$\mathbf{A} = \max_{h,w,d} (\mathbf{M}), \quad \mathbf{A} \in \mathbb{R}^{B \times T}, \tag{5.3}$$

Then, we select the indices of the top-k highest responding tokens:

$$\mathcal{I}_k = \texttt{TopK}(\mathbf{A},\ k = q). \tag{5.4}$$

These selected token embeddings are gathered and re-normalized:

$$\mathbf{T}' = \texttt{Gather}(\texttt{Norm}(\mathbf{T}),\ \mathcal{I}_k), \quad \mathbf{T}' \in \mathbb{R}^{B \times k \times D} \tag{5.5}$$

where $\texttt{Norm}(\cdot)$ denotes ℓ_2 normalization along the feature dimension. $\texttt{Gather}(\cdot, \mathcal{I})$ retrieves the top-k token embeddings along the token dimension based on the selected indices $\mathcal{I}_k$. For a tensor $\mathbf{T} \in \mathbb{R}^{B \times T \times D}$ and index set $\mathcal{I}_k \in \mathbb{R}^{B \times k}$, this operation selects k tokens per sample in the batch and preserves the prompt and feature structure. Then, we recompute the refined cross-modal cost map via:

$$M'_{(i,j,n)} = \frac{E_{(i,j)} \cdot T'_n}{\|E_{(i,j)}\| \|T'_n\|}. \tag{5.6}$$

5.2.6 Feature Refinement Module

As a segmentation task, it is intuitive to further explore spatial-level and class-level information. As shown in Figure 5.5, we first utilize the Swin-Transformer block [75] to process the fused visual features for enriching spatial feature information as in [53], [100]. Then, we perform class-level feature refinement to map textual information onto each pixel, achieving more precise alignment. We feed the text embedding into a linear transformer block generated from the comprehensive prompts through the CLIP text encoder. Finally, we leverage the fused feature again to up-sample the enhanced feature representations. The overall process of feature refinement is summarized as follows:

$$M''_{(i,j,n)} = S\left([M'_{(i,j,n)}; E]\right), \tag{5.7}$$

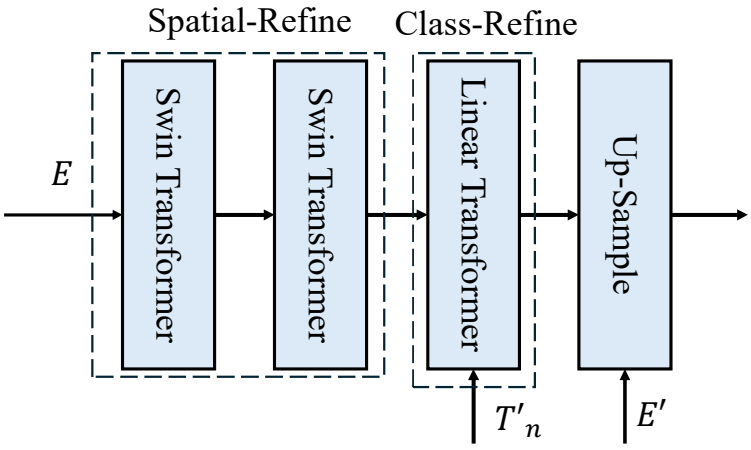


Figure 5.5: **The Feature Refinement Module. We first perform spatial-level feature enhancement and then aggregate class-level features.**

$$M'''_{(i,j,n)} = C\left([M''_{(i,j,n)}; T'_n]\right), \tag{5.8}$$

$$O = Up\left([M'''_{(i,j,n)}; E']\right), \tag{5.9}$$

where S and C represent spatial- and class-level refinements, E' is the intermediate visual feature layer, Up denotes up-sampling, and O is the final prediction.

5.3 Experiments

5.3.1 Dataset and evaluation protocol

We train our model on COCO Stuff [70] dataset, and conduct evaluation on ADE20k-847 [1] ADE20k-150 [1], Pascal Context-459 [71], Pascal Context-59 [71], and Pascal VOC [107]. COCO-Stuff dataset contains 171 annotated classes and includes 118k training, 5k validation, and 41k test images.

ADE20K [1] is a large-scale benchmark for scene understanding, comprising 20k training images, 2k validation images, and 3k testing images. It includes two sets of annotated classes: ADE20K-150 with 150 classes and ADE20K-847 with 847 classes, although both use the same images.

Pascal Context [71] extends Pascal VOC 2010, offering 4,998 training and 5,005 validation images, with annotations available in two configurations: PC-59 (59 classes) and PC-459 (459 classes).

Pascal VOC [71] consists of 11,185 training images and 1,449 validation images across 20 object classes.

Evaluation Metric: Following previous work [51], [100], [108], Mean Intersection over Union (mIoU) is used as the evaluation metric across all experiments. This metric represents the average intersection-over-union values calculated for each class across all classes.

5.3.2 Implementation Details

Our experiments utilize the pre-trained CLIP model from OpenAI [45], specifically the ViT-B/16 and ViT-L/14 variants. We fine-tune the CLIP image and text encoder and the total training iteration is set as 80k. The initial learnable fusion weight is empirically set as 0.5 for balance. We use 2 NVIDIA-L40 GPUs for training with a batch size of 4 and the AdamW optimizer with an initial learning rate of 2×10^{-4}. The weight decay is 1×10^{-4} for our model. During training, the input image resolution is 384×384 for ViT-B/16. For ViT-L/14, the resolution is 336×336.

5.3.3 Comparisons with State-of-the-art Methods

We compare our method with existing state-of-the-art approaches across six datasets in Table 5.1, including the vision-language model (VLM) and training dataset. Apart from SPNet [109] and ZS3Net [110], most methods are developed using VLM as a foundation.

To ensure a fair comparison, the results using the same vision-language model are grouped together. Existing open-vocabulary semantic segmentation methods have explored various strategies to bridge the gap between vision and language, yet they often struggle with accurately segmenting unseen classes. In contrast, our approach achieves notable performance in accurately segmenting both seen and unseen classes. With ViT-B/16 as the vision-language model, our LSMSeg outperforms SAN [51], SED [53], and CATSeg [100] by 5.7%, 1.6%, and 1.4% on A-150. On PC-459, our method exceed SED [53], EBSeg [99], and CAT-Seg [100] by 1.3%, 2.6%, and 0.9%. When using a larger model ViT-L, our LSMSeg also attains notable performance on all six datasets. For instance, on ADE-150, our LSMSeg outperforms FC-CLIP [108], SED [53] and CAT-Seg [100] by 4.4%, 3.3% and 0.6%. Our method achieves favorable performance with both base and large models. The results for the seen and unseen classes of VOCb are presented in Table 5.2. As can be seen, our method consistently outperforms CATSeg [100] on both seen

Method	VLM	Training Dataset	A-847	PC-459	A-150	PC-59	VOC	VOCb
SPNet [109]	-	PASCAL VOC	-	-	-	24.3	18.3	-
ZS3Net [110]	-	PASCAL VOC	-	-	-	19.4	38.3	-
Lseg+ [111]	ALIGN EN-B7	COCO-Stuff	3.8	7.8	18.0	46.5	-	-
OpenSeg [46]	ALIGN EN-B7	COCO Panoptic	8.1	11.5	26.4	44.8	-	70.2
ZegFormer [49]	CLIP ViT-B/16	COCO-Stuff	5.6	10.4	18.0	45.5	89.5	65.5
DeOP [112]	CLIP ViT-B/16	COCO-Stuff-156	7.1	9.4	22.9	48.8	91.7	-
OVSeg [47]	CLIP ViT-B/16	COCO-Stuff	7.1	11.0	24.8	53.3	92.6	-
SAN [51]	CLIP ViT-B/16	COCO-Stuff	10.1	12.6	27.5	53.8	94.0	-
SCAN [52]	CLIP ViT-B/16	COCO-Stuff	10.8	13.2	30.8	58.4	**97.0**	-
EBSeg [99]	CLIP ViT-B/16	COCO-Stuff	11.1	17.3	30.0	56.7	94.6	-
SED [53]	ConvNeXt-B	COCO-Stuff	11.4	18.6	31.6	57.3	94.4	-
CAT-Seg [100]	CLIP ViT-B/16	COCO-Stuff	<u>12.0</u>	<u>19.0</u>	<u>31.8</u>	<u>57.5</u>	94.6	<u>77.3</u>
LSMSeg (*ours*)	CLIP ViT-B/16	COCO-Stuff	**12.8**	**19.9**	**33.2**	**59.4**	<u>95.2</u>	**80.8**
SimSeg [48]	CLIP ViT-L/14	COCO-Stuff	7.1	10.2	21.7	52.2	92.3	-
MaskCLIP [113]	CLIP ViT-L/14	COCO Panoptic	8.2	10.0	23.7	45.9	-	-
OVSeg [47]	CLIP ViT-L/14	COCO-Stuff	9.0	12.4	29.6	55.7	94.5	-
ODISE [50]	CLIP ViT-L/14	COCO-Stuff	11.1	14.5	29.9	57.3	-	-
SAN [51]	CLIP ViT-L/14	COCO-Stuff	12.4	15.7	32.1	57.7	94.6	-
EBSeg [99]	CLIP ViT-L/14	COCO-Stuff	13.7	21.0	32.8	60.2	96.4	-
SCAN [52]	CLIP ViT-L/14	COCO-Stuff	14.0	16.7	33.5	59.3	97.2	-
FC-CLIP [108]	ConvNeXt-L	COCO Panoptic	14.8	18.2	34.1	58.4	95.4	81.8
SED [53]	ConvNeXt-L	COCO-Stuff	13.9	22.6	35.2	60.6	96.1	-
MAFT+ [114]	CLIP ViT-L/14	COCO-Stuff	15.1	21.6	36.1	59.4	96.5	-
DPSeg[115]	ConvNeXt-L	COCO-Stuff	14.9	23.5	36.4	62.0	**97.4**	-
CAT-Seg [100]	CLIP ViT-L/14	COCO-Stuff	<u>16.0</u>	<u>23.8</u>	<u>37.9</u>	<u>63.3</u>	97.0	<u>82.5</u>
LSMSeg (*ours*)	CLIP ViT-L/14	COCO-Stuff	**16.8**	**25.5**	**38.5**	**63.4**	<u>97.2</u>	**83.6**

Table 5.1: **Comparison with state-of-the-art methods. We present the mIoU(mean Intersection over Union) results on six commonly used test sets for open-vocabulary semantic segmentation. The highest results are highlighted in bold, and the second highest are underlined. Compared with other methods, our proposed LSMSeg demonstrates superior performance across all six test sets.**

Method	Seen Classes																Unseen Classes						
	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	person	sheep	train	pottedplant	mIoU	aeroplane	diningtable	motorbike	sofa	tvmonitor	background	mIoU
CAT-Seg [100]	44.8	89.7	81.3	84.6	98.0	78.4	95.5	49.9	97.6	95.0	94.8	92.9	95.9	91.1	69.2	83.9	95.7	44.8	92.8	61.4	84.6	93.8	78.8
LMSeg (*ours*)	47.7	90.3	81.4	86.3	98.1	79.7	96.5	51.7	97.7	96.0	95.0	93.0	96.0	93.1	65.8	84.5	96.4	53.5	93.3	60.7	81.3	94.4	79.9

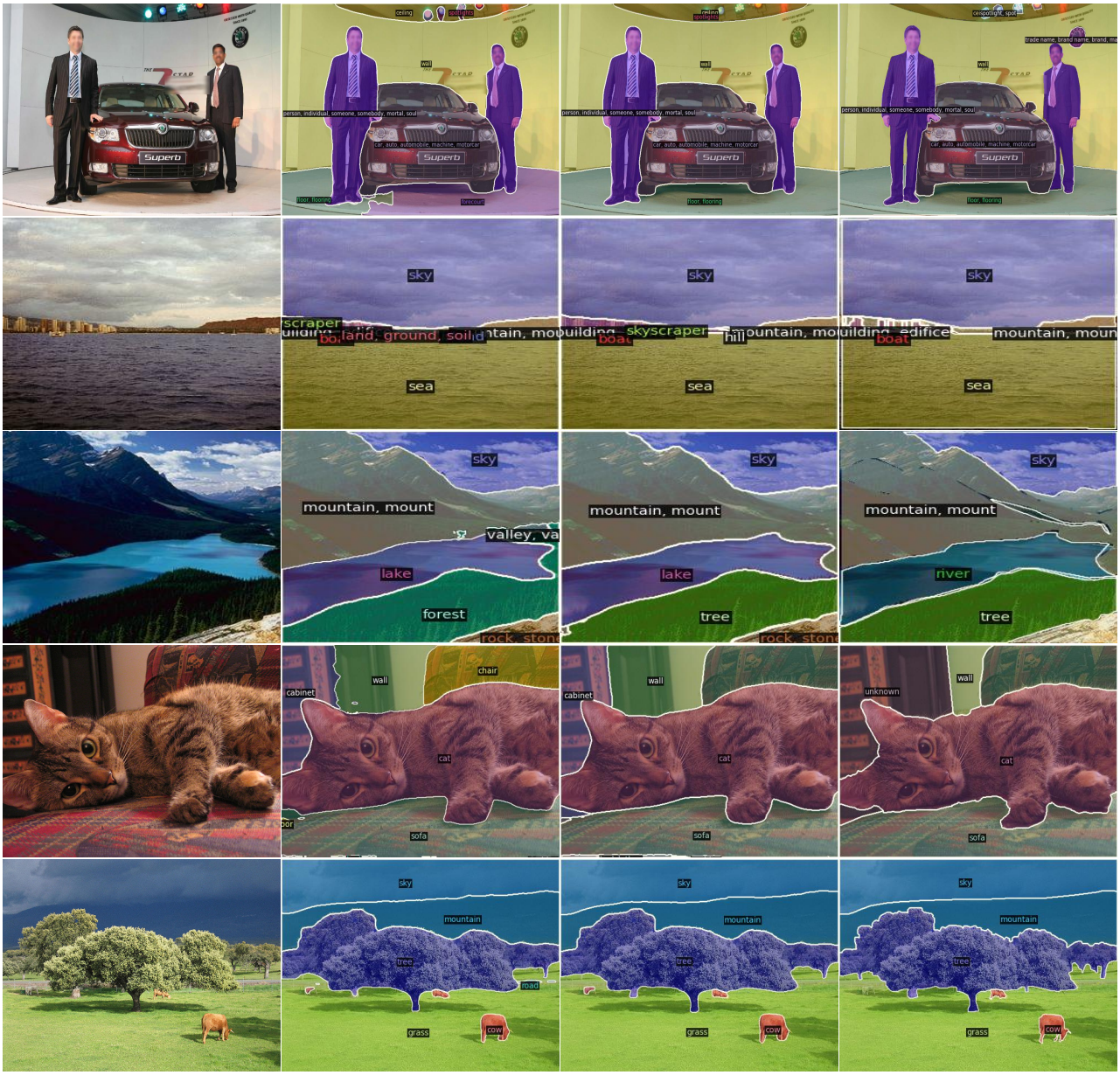
Table 5.2: **Comparison of each category in VOCb. We divided the classes into seen and unseen categories. mIoU (mean Intersection over Union) measures the average overlap between predicted and ground-truth segments across all classes.**

(a) Image (b) CAT-Seg (c) LSMSeg (ours) (d) Ground truth

Figure 5.6: **Qualitative comparisons on PC-459. From left to right: input images, results of CAT-Seg, results of our LSMSeg, and ground truth.**

(a) Image (b) CAT-Seg (c) LMSeg (ours) (d) Ground truth

Figure 5.7: **Qualitative comparisons on A-150. From left to right: input images, results of CAT-Seg, results of our LMSeg, and ground truth.**



(a) Image (b) CAT-Seg (c) LMSeg (ours) (d) Ground truth

Figure 5.8: **Qualitative comparisons on A-847. From left to right: input images, results of CAT-Seg, results of our LMSeg, and ground truth.**

Methods	A-847	PC-459	A-150	PC-59	VOC	VOCb	avg.
Baseline	11.0	17.9	28.4	54.6	94.6	74.3	46.8
Color	11.2	17.8	28.3	55.3	94.5	76.5	47.3
Shape	11.2	18.4	28.2	54.7	94.9	77.1	47.4
Size	11.6	18.2	28.3	55.8	94.3	76.2	47.4
Texture	11.3	18.2	28.4	55.5	94.8	76.2	47.4
Material	11.0	18.2	27.7	55.9	93.6	75.4	47.0
Positation or Location	11.4	18.0	28.0	55.0	93.1	75.7	46.9
Pattern	10.9	16.9	28.2	55.2	94.8	76.2	47.0
Action or Satae	11.2	18.1	13.9	42.2	94.9	76.4	42.8
Contextual Relationship	7.1	17.4	16.5	29.8	94.6	73.9	39.9

Table 5.3: **Analysis of different prompts. We conduct an ablation study on each visual attribute individually to verify the positive and negative attributes.**

and unseen classes. Additionally, we present qualitative comparisons on Pascal Context (459 categories) and ADE20k (150 and 847 categories) in Figure 5.6, 5.7 and 5.8, demonstrating the superior effectiveness of our proposed LSMSeg approach relative to the cutting-edge method.

5.3.4 Ablation Study

Analysis of different prompts.

As mentioned in Section 5.2.3, we have obtained different visual attributes *e.g.* color, shape, size, texture, material, position or location, pattern, action or state, and contextual relationship. Here, we utilize ViT-B/16 as the VLM and training on the COCO-Stuff dataset using CATSeg [100] without spatial and class aggregation. To identify the attributes most essential for the segmentation task, we conduct an extensive series of experiments, with the findings concisely presented in Table 5.3. Due to the instability of results on some datasets, we calculated the average performance across all datasets to determine the best approach. The baseline method [100], relying on fixed hand-crafted prompts, achieves an average mIoU of 46.8%. Attributes perform differently: color (47.3%), shape (47.4%), texture (47.4%), and size (47.4%) lead, followed by material (47.0%), pattern (47.0%), position (46.9%), action/state (42.8%),

Methods	A-847	PC-459	A-150	PC-59	VOC	VOCb	avg.
Size	11.6	18.2	28.3	55.8	94.3	76.2	47.4
Size + Shape	11.3	18.2	28.2	55.3	95.1	76.8	47.5
Size + Shape +Texture	11.4	18.5	28.7	54.4	95.0	77.5	47.6
Size + Shape +Texture + Color	11.6	18.4	28.9	55.6	94.9	77.1	**47.8**
Size + Shape +Texture + Color + Material	11.6	18.3	28.7	55.4	94.9	76.3	47.5

Table 5.4: **Ablation Study on Attribute Combinations. We conduct an ablation study on different combinations of different attributes and identify the optimal combination.**

and contextual relationship (39.9%). The top performers are color, size, shape, and texture.

Ablation Study on Attribute Combinations.

Subsequently, we conduct a detailed analysis of various combinations of visual attributes, as shown in Table 5.4. We gradually incorporate attributes such as *size*, *shape*, *texture*, *color*, and *material* into the generated prompts and examine their impact on segmentation accuracy. The results show a steady improvement: an average mIoU of 47.5% is achieved with size and shape, which increases to 47.6% when texture is added, and further reaches 47.8% with the inclusion of color. However, when material is also included, the performance drops slightly back to 47.5%.

These findings suggest that certain attributes contribute more effectively to the segmentation task. In particular, size, shape, texture, and color provide complementary information that helps the model distinguish between different categories more accurately. On the other hand, adding material does not lead to further improvement and may even introduce noise, possibly because material is harder to describe visually and less consistent across categories.

Overall, this experiment highlights the importance of selecting meaningful attribute combinations. Carefully chosen attributes can enrich the textual descriptions and enhance the alignment between text and visual features, leading to better pixel-level classification results.

Ablation study for CFM.

We investigate the effect of the filtered class number in CFM. Table 5.5 illustrates the impact of varying the class number k during training on the COCO-Stuff dataset, showing that $k = 32$,

K	A-847	PC-459	A-150	PC-59	VOC	VOCb	avg.
16	11.7	19.0	30.9	58.3	95.0	79.9	49.1
32	12.8	19.7	32.1	58.0	94.8	80.0	49.6
48	12.6	19.6	32.0	58.1	95.2	79.9	49.6
64	12.6	19.9	32.0	58.4	94.8	79.7	49.6
96	12.6	19.8	32.2	58.4	94.8	79.6	49.6

Table 5.5: **Ablation Study on class number k. It shows that choosing a k value that's too high or too low can hinder the model's ability to effectively capture contextual information.**

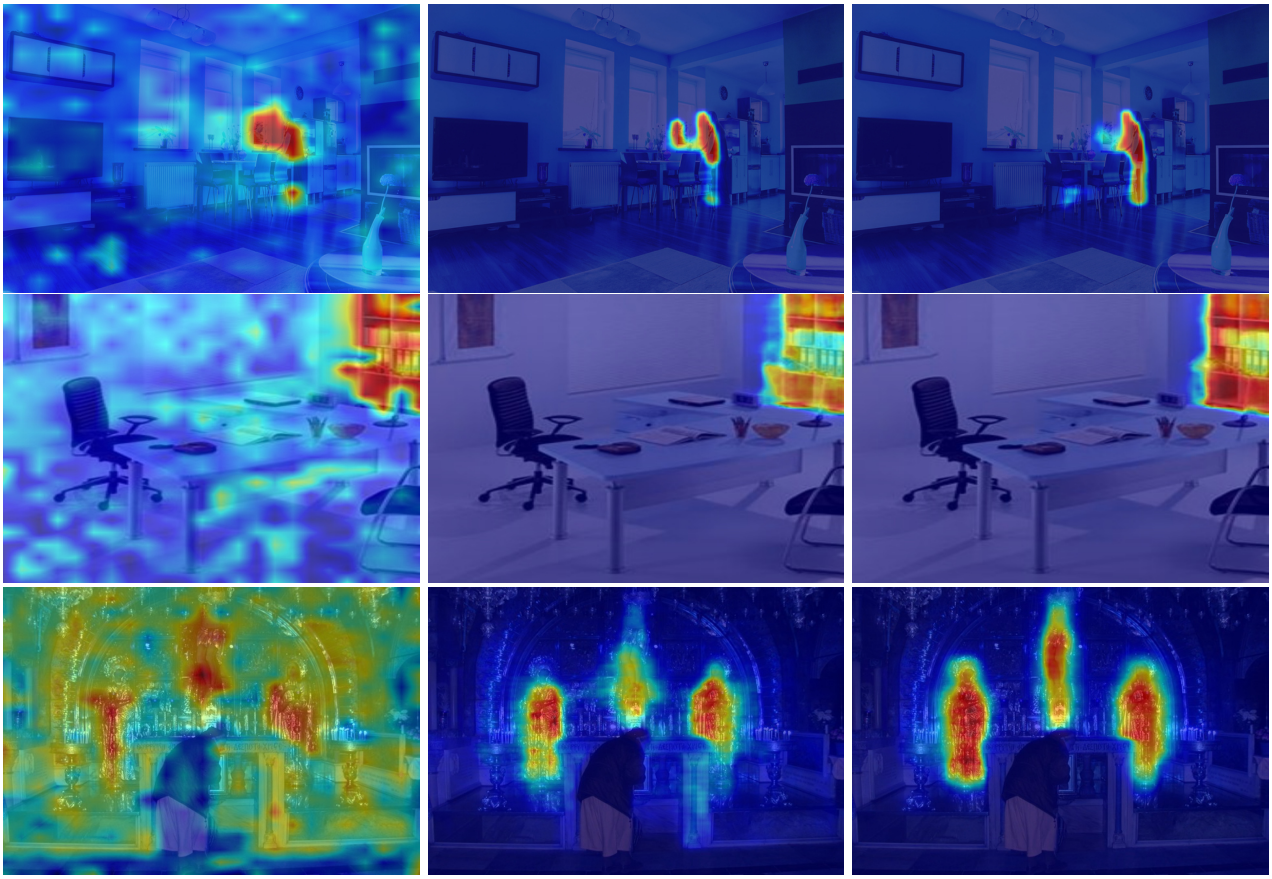
Methods	A-847	PC-459	A-150	PC-59	VOC	VOCb
CATSeg†	11.9	18.5	31.7	57.6	94.6	77.0
LSMSeg(w/o SAM)	12.8	19.7	32.1	58.0	94.8	80.0
LSMSeg(w/ Dinov2-B)	12.2	18.8	31.1	57.2	94.9	78.7
LSMSeg(w/ SAM-B)	12.5	20.3	32.1	58.8	95.2	80.2
LSMSeg(w/ SAM-L)	12.8	19.9	33.2	59.4	95.2	80.8

Table 5.6: **Ablation study on Enriched Prompt and SAM. We conduct an ablation study to verify the effectiveness of our proposed Enriched Prompt and SAM. † Represents the results of our replication. In this experiment, we adopt CLIP ViT-B/16 as the backbone.**

$k = 48$, $k = 64$, and $k = 96$ yield similar performance. Given this, $k = 32$ is selected as the optimal choice when considering overall computational complexity. Notably, while increasing k marginally improves the model's flexibility to capture fine-grained semantics, the performance gain remains limited. This indicates that a relatively small subset of informative classes is sufficient to guide the model's learning process. Therefore, our choice of $k = 32$ achieves a good trade-off between segmentation performance and computational efficiency.

Ablation Study for Visual Feature Fusion.

To evaluate the effectiveness of our enriched text prompts and SAM-based visual guidance, we integrate them into the CAT-Seg [100] framework, which serves as our baseline due to its



(a) CLIP (b) CAT-Seg (c) LSMSeg (ours)

Figure 5.9: **Visualization of the cost map for different methods. The cost map represents the alignment between image and text features. The first row indicates the seen class 'person,' and the last two rows indicate the unseen classes 'bookcase' and 'sculpture.'**

strong performance in open-vocabulary semantic segmentation (OVSS). The baseline CAT-Seg employs fixed, hand-crafted prompts. In contrast, our LSMSeg enhances CAT-Seg in two key ways: (1) replacing simplistic prompts with enriched linguistic descriptions generated by GPT-4, incorporating the optimal combination of size, shape, texture, and color; and (2) integrating visual priors from SAM to guide feature alignment. Table 5.9 shows that enriched prompts alone improve performance (e.g., +0.9 mIoU on A-847, +1.2 on PC-459), while adding SAM-B or SAM-L further boosts results, with LSMSeg(w/ SAM-L) achieving the best performance. Substituting SAM with DINOv2-B yields limited gains. This improvement is particularly evident on challenging datasets with diverse categories, such as A-150 and PC-59, where richer prompts and spatial priors enhance pixel-text alignment for both seen and unseen classes. We additionally presents an ablation study on visual feature fusion strategies combining CLIP and

Methods	A-847	PC-459	A-150	PC-59	VOC	VOCb	avg.
Baseline	12.0	19.0	31.8	57.5	94.6	77.3	48.7
Concat	12.1	19.3	32.0	57.5	95.0	77.0	48.8
Attn	12.0	19.5	32.0	57.6	95.0	76.6	48.8
Weighted	12.4	19.3	32.2	57.7	95.2	77.0	**49.0**

Table 5.7: **Ablation study on different fusion strategies. We conduct an ablation study by utilizing different fusion strategies, including concatenation, attention-based, and learnable weighted methods.**

SAM features without using enriched text prompts in Table 5.7. The reason why the learnable weighted fusion outperforms the attention-based fusion might be due to the heterogeneity of SAM and CLIP features. SAM features are optimized for segmentation tasks, while CLIP features are aligned with text information, leading to different feature distributions. Attention mechanisms might struggle to effectively integrate these diverse features, as they could average out the importance of each feature type. In contrast, learnable weights can adaptively adjust the contributions of each feature source, effectively addressing redundancy and conflicts and thereby enhancing overall performance. Qualitative comparisons in Figure 5.9 further illustrate this advantage, with LSMSeg producing more precise segmentation.

Model efficiency.

Based on the efficiency comparison presented in Table 5.8, LSMSeg outperforms the baseline method CAT-Seg on PC-459 dataset. LSMSeg achieves 156.4M Params, 362.8ms Latency, 446 minutes of training Time, and 2122.02 GFLOPs, while CAT-Seg records 156.4M Params, 535.2ms Latency, 693.7 minutes of training Time, and 3459.7 GFLOPs. This indicates that LSMSeg reduces training time by approximately 247 minutes and enhances computational efficiency with a lower GFLOPs requirement. Additionally, LSMSeg exhibits a higher mIoU performance compared to CAT-Seg, further demonstrating its superiority in both efficiency and accuracy. The reduced latency of LSMSeg (362.8ms vs. 535.2ms) also highlights its advantage in processing speed, positioning it as a more effective solution across multiple performance metrics.

Methods	Params (M)	Latency (ms)	Time (min)	GFLOPs
ZegFormer	531.2	2700	1148.3	19,425.6
OVSeg	532.6	2000	-	19,345.6
CAT-Seg	156.4	535.2	693.7	3459.7
SED	180.8	358.9	1702	1068.9
LSMSeg(w/o SAM)	**156.4**	362.8	**446**	2122.0

Table 5.8: **Efficiency comparison. All results are measured with Nvidia-L40 GPU. Time stands for training time.**

Methods	A-847	PC-459	A-150	PC-59	VOC	VOCb
Qwen2.5	12.4	19.6	32.1	58.4	**96.0**	79.7
GPT-4	**13.1**	**20.3**	**33.3**	**59.7**	95.4	**81.1**

Table 5.9: **Ablation study on different LLMs. We conduct an ablation study to verify the effectiveness of different LLMs.**

5.4 Ablation study on different LLMs

Table 5.9 reports the ablation results comparing different large language models used for generating attribute-enriched descriptions. We evaluate two representative LLMs: Qwen2.5 [116] and GPT-4 [55] under six benchmarks. Across all datasets, GPT consistently outperforms Qwen, achieving higher mIoU on both large-scale (A-847, PC-459) and small-scale (A-150, PC-59) benchmarks. Notably, GPT also yields competitive or superior performance on VOC and VOCb. These results indicate that higher-quality language descriptions generated by stronger LLMs can further enhance the semantic alignment between visual and textual features, thereby improving segmentation performance.

Ablation study on fine-tuning the encoder of LSMSeg.

Table 5.10 presents an ablation study on fine-tuning components of CLIP. Due to computational cost constraints, the study did not fine-tune the SAM model, focusing instead on the specified CLIP components. When we freeze the CLIP encoder, the lowest result is achieved across six datasets. The best fine-tuning strategy for CLIP is to fine-tune query and value projections

only, with an average result of 50.02%.

Methods	A-847	PC-459	A-150	PC-59	VOC	VOCb	avg.
Freeze	8.1	13.3	25.9	46.9	83.4	61.6	39.87
CLIP_{qk}	11.6	18.2	30.7	56.2	94.7	78.5	48.32
CLIP_{kv}	12.7	19.9	32.2	59.0	95.0	80.2	49.83
CLIP_{qv}	12.8	19.9	33.2	59.4	95.2	80.8	50.02

Table 5.10: **Ablation study on fine-tuning encoder of LSMSeg. We conduct an ablation study on fine-tuning the CLIP encoder during training.** q, k**, and** v **of CLIP are query, key, and value projections.**

Training on various datasets.

To evaluate the generalization capability of our LSMSeg in Table 5.11. We further conduct experiments on different scale datasets, including A-150 and PC-59. Our LSMseg achieves the best performance across all datasets compared with ZegFormer [49], SimSeg [48], and CAT-Seg [100] when training on COCO-Stuff and A-150. When training on PC-59, our LSMseg achieves the best results on A-847, PC-459, A-150, VOCb and PC-59. However, it falls just slightly short of CAT-Seg on VOC. Based on the observation, our method demonstrates generalization capability across different datasets, as well as more precise pixel-level recognition ability.

Examples of Class Descriptions

We present a selection of detailed class descriptions generated using the GPT-4 model by OpenAI. For each class in the COCO-Stuff vocabulary, we provide four descriptions in terms of color, shape/size, texture/material, and comprehensive combination.

Generated descriptions for "bicycle"

- `A bicycle has a two-wheeled frame with handlebars and a seat, is medium-sized at around 1 to 1.5 meters in length, has a smooth metal frame, rubber tires, and textured handle grips, and is often red, blue, black, or metallic with shiny or matte finishes.`

Methods	Training dataset	A-847	PC-459	A-150	PC-59	VOC	VOCb
ZegFormer [49]	COCO-Stuff	5.6	10.4	18.0	45.5	89.5	65.5
SimSeg [48]	COCO-Stuff	7.0	9.0	20.5	47.7	88.4	67.9
CAT-Seg [100]	COCO-Stuff	12.0	19.0	31.8	57.5	94.6	77.3
LSMSeg (ours)	COCO-Stuff	**12.8**	**19.9**	**33.2**	**59.4**	**95.2**	**80.8**
ZegFormer [49]	PC-59	3.8	8.2	13.1	48.7	86.5	66.8
SimSeg [48]	PC-59	3.0	7.6	11.9	54.7	87.7	71.7
CAT-Seg [100]	PC-59	9.6	16.7	27.4	63.7	**93.5**	79.9
LSMSeg (ours)	PC-59	**11.1**	**18.3**	**29.1**	**64.2**	93.4	**80.1**
ZegFormer [49]	A-150	6.8	7.1	33.1	34.7	77.2	53.6
SimSeg [48]	A-150	7.6	7.1	40.3	39.7	80.9	61.1
CAT-Seg [100]	A-150	14.4	16.2	47.7	49.9	91.1	73.4
LSMSeg (ours)	A-150	**15.3**	**17.4**	**49.2**	**53.6**	**92.8**	**74.5**

Table 5.11: **Ablation study of training on different datasets. We train our model on different scale datasets to demonstrate the generalization capabilities.**

Generated descriptions for "car"

- `A car has a boxy or sleek aerodynamic shape with four wheels, varies in size from compact to SUVs and large sedans, has a smooth metal body, rubber tires, and leather or fabric seats, and is usually white, black, red, or blue with a glossy finish.`

Generated descriptions for "airplane"

- `An airplane has a long fuselage with two wings and a tail fin, is very large, ranging from small private jets to massive airliners, has a smooth metal surface with rivets and windows, and is typically white, gray, or silver, sometimes with colorful airline logos.`

Generated descriptions for "bench"

- `A bench has a long, rectangular seat with a flat or slightly curved surface, is medium to large, seating two to four people, has a smooth wooden surface or a textured metal or stone finish, and is often brown, gray, or green, blending into outdoor environments.`

Generated descriptions for "snowboard"

- `A snowboard is medium to large, usually around 140-170 cm, with a long slightly curved shape, rounded ends, and a smooth glossy underside with a rough grippy top surface, often black, white, or brightly colored with artistic designs.`

5.5 Conclusion

This study addresses the challenge of **open-vocabulary semantic segmentation (OVSS)**, which demands accurate pixel-level alignment between visual content and an open set of textual descriptions, including *unseen or novel categories*. Existing OVSS methods often rely on static, hand-crafted text templates that lack expressiveness and fail to reflect the *diverse visual attributes* present in real-world images. These limitations result in poor generalization to unseen categories and reduced segmentation precision.

To overcome these issues, we propose **LSMSeg**, a novel and effective framework that leverages the complementary strengths of *large-scale models* in vision and language. At its core, LSMSeg utilizes **Large Language Models (LLMs)**, such as GPT-4, to automatically generate *rich, attribute-aware text prompts* for each class. These prompts incorporate fine-grained visual details—*color*, *shape*, *size*, and *texture*—to better guide the semantic interpretation of visual content. These enhanced prompts are then used in conjunction with the **Segment Anything Model (SAM)** and **CLIP**, forming a unified pipeline where the text guides CLIP's visual representation and SAM refines spatial boundaries at the pixel level.

The key innovation behind LSMSeg lies in its **multi-modal synergy**: (i) the LLM-generated prompts capture category-specific nuances, vastly improving text-visual alignment; (ii) CLIP leverages this enriched textual information to produce more semantically relevant visual embeddings; and (iii) SAM ensures spatial precision, resulting in segmentation outputs that are both *semantically accurate* and *spatially coherent*.

Despite its modularity, LSMSeg remains **computationally efficient**, requiring no end-to-end

training and allowing integration with various backbone architectures. Its plug-and-play nature makes it suitable for practical deployment across domains with *limited annotations* or *dynamic category sets*.

We validate LSMSeg's effectiveness through extensive experiments on **six OVSS benchmarks**, including A-847, PC-459, and Pascal VOC, where it consistently *outperforms prior state-of-the-art methods*, especially in *generalizing to unseen categories* and capturing fine object boundaries. These results highlight the potential of LLM-driven enhancements in addressing the open-vocabulary segmentation problem.

Nonetheless, LSMSeg also opens up new research directions. Challenges remain in: (i) handling *ambiguous or abstract categories* that lack well-defined visual features; (ii) designing *adaptive prompt generation* mechanisms tailored to specific visual scenes; and (iii) optimizing the framework for *real-time applications*, particularly in low-resource environments.

Future work will explore integrating *prompt refinement modules* that dynamically adjust text based on visual feedback, as well as *lightweight model variants* to reduce computation while preserving accuracy. Additionally, applying LSMSeg to tasks like *zero-shot instance segmentation*, *multi-modal retrieval*, or *interactive segmentation* could further validate its generalizability.

In conclusion, LSMSeg represents a significant step forward in open-vocabulary segmentation by **bridging vision-language understanding** through large-scale models. Its innovative design and strong empirical performance suggest broad applicability and offer a foundation for future exploration in multimodal, scalable segmentation systems.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

This thesis has addressed the critical challenges of improving the generalizability and robustness of deep learning models for image semantic segmentation. By exploring innovative techniques across multiple levels—image, dataset, and open-vocabulary—this work has contributed novel solutions that enhance segmentation performance in real-world scenarios, where data is often noisy, imbalanced, or incomplete.

The key contributions of this thesis are summarized as follows:

- **Image-level contextual information for semantic segmentation**: We proposed a Class-Aware Affinity Module (CAA) that explicitly models pixel-to-pixel and pixel-to-class semantic relationships within each image. By learning intra-class and inter-class pixel affinities and refining class center representations, the CAA module enhances local semantic coherence and boundary precision. Integrated into standard segmentation architectures, CAA consistently improves performance across multiple datasets, including ADE20K, COCO-Stuff, and Pascal-Context, demonstrating its broad applicability and robustness.

- **Dataset-level contextual information for semantic segmentation**: To complement local information with global priors, we introduced the Extended Context-Aware Classifier (ECAC), which maintains a memory bank of dataset-level class features and combines them with image-specific features. The ECAC employs a teacher-student learning frame-

work to guide the learning of class centers and mitigate the effects of class imbalance and annotation noise. This approach ensures robust and balanced representation learning across frequent and rare categories, leading to improved generalization in diverse and challenging environments.

- **Open-vocabulary semantic segmentation**: We developed LSMSeg, a novel framework for open-vocabulary segmentation that leverages Large Language Models (LLMs) to generate visually descriptive text prompts enriched with attributes such as color, size, shape, and texture. These prompts provide fine-grained supervision to guide CLIP-based visual encoding, while the Segment Anything Model (SAM) ensures accurate spatial alignment. The synergy between textual richness and visual precision enables LSMSeg to achieve state-of-the-art results on six benchmark datasets, demonstrating its effectiveness in aligning vision and language for open-category understanding.

Collectively, the contributions of this thesis provide a unified perspective on enhancing semantic segmentation through hierarchical context modeling—from within-image spatial relationships, to dataset-level semantics, to cross-modal generalization in open-vocabulary settings. These insights lay a strong foundation for future research in scalable and adaptive segmentation systems.

6.2 Future Work

The advancements presented in this thesis mark significant progress in improving the accuracy, generalization, and flexibility of image semantic segmentation models. Nevertheless, several open challenges remain, and there are multiple promising directions for future exploration that can extend the impact of this work:

- **Lightweight Context Modeling**: While affinity modeling and context-aware mechanisms offer notable improvements, they can introduce computational overhead. Future research could investigate more efficient designs, such as dynamic sparse affinity propagation, low-rank attention, or token pruning techniques to capture long-range dependencies without compromising real-time performance. These lightweight approaches will be particularly important for deploying segmentation models on edge devices or in latency-sensitive applications.

- **Spatiotemporal Context for Video Segmentation**: Extending the current methods to video data introduces new challenges and opportunities. Future work could incorporate temporal consistency, motion cues, and object tracking into the pixel-to-class affinity framework to enhance robustness over time. Learning temporally coherent features that align with evolving semantic concepts can significantly improve video semantic segmentation and enable real-world applications such as autonomous driving or video editing.

- **Multi-modal Fusion for Richer Contextual Understanding**: Incorporating complementary modalities—such as depth maps, optical flow, or even audio signals in multimodal video data—can provide additional cues for disambiguating visually similar classes and detecting occluded regions. Future models may benefit from learning joint representations across modalities using cross-attention or contrastive objectives, leading to more robust and context-aware segmentation systems.

- **Open-vocabulary Generalization and Prompt Adaptation**: While LSMSeg demonstrates strong performance with static prompts, further improvements could be achieved through adaptive prompt generation conditioned on scene content. Additionally, exploring self-supervised or few-shot learning techniques to enhance open-vocabulary generalization under limited supervision remains an open and impactful direction.

- **Scalability and Real-world Applications**: Finally, applying the proposed modules in real-world settings—such as medical imaging, remote sensing, or industrial inspection—may require additional adaptations. Investigating domain adaptation, label-efficient training, and user-interactive refinement will be key to transitioning from research prototypes to practical systems.

In summary, this thesis lays the groundwork for context-aware, scalable, and open-vocabulary segmentation models. Continued research in the directions outlined above holds great promise for advancing the field toward more intelligent and adaptive visual understanding systems.

Bibliography

[1] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.

[2] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," *Advances in neural information processing systems*, vol. 17, 2004.

[3] Z. Wu, D. Lin, and X. Tang, "Deep markov random field for image modeling," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, Springer, 2016, pp. 295–312.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[5] O. Ronneberger, P. Fischer, and T. 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.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *International Conference on Learning Representations*, 2021.

[9] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[10] A. G. Howard, M. Zhu, B. Chen, *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[11] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng, "Strip pooling: Rethinking spatial pooling for scene parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4003–4012.

[12] Z. Jin, B. Liu, Q. Chu, and N. Yu, "Isnet: Integrate image-level and semantic-level context for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7189–7198.

[13] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 603–612.

[14] C. Gao, H. Ye, F. Cao, C. Wen, Q. Zhang, and F. Zhang, "Multiscale fused network with additive channel–spatial attention for image segmentation," *Knowledge-Based Systems*, vol. 214, p. 106 754, 2021.

[15] X. Liu, L. Jiao, L. Li, X. Tang, and Y. Guo, "Deep multi-level fusion network for multi-source image pixel-wise classification," *Knowledge-Based Systems*, vol. 221, p. 106 921, 2021.

[16] H. Tang, Y. Zhao, Y. Jiang, Z. Gan, and Q. Wu, "Class-aware contextual information for semantic segmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, pp. 1–5.

[17] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, *arXiv preprint arXiv:1706.05587*, 2017.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[20] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2393–2402.

[21] H. Zhang, K. Dana, J. Shi, *et al.*, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[22] J. Fu, J. Liu, Y. Wang, *et al.*, "Adaptive context network for scene parsing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6748–6757.

[23] Z. Lin, M. Feng, C. N. d. Santos, *et al.*, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[24] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291–4308, 2020.

[25] J. Fu, J. Liu, H. Tian, *et al.*, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[26] B. Lu, Q. Hu, Y. Wang, and G. Hu, "Rcanet: Row-column attention network for semantic segmentation," in *ICASSP*, IEEE, 2022, pp. 2604–2608.

[27] Z. Zhong, Z. Q. Lin, R. Bidart, *et al.*, "Squeeze-and-attention networks for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 065–13 074.

[28] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*, PMLR, 2019, pp. 7354–7363.

[29] F. Zhang, Y. Chen, Z. Li, *et al.*, "Acfnet: Attentional class feature network for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6798–6807.

[30] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *European conference on computer vision*, Springer, 2020, pp. 173–190.

[31] Z. Li, Y. Sun, L. Zhang, and J. Tang, "Ctnet: Context-based tandem network for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[32] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang, "Context prior for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 416–12 425.

[33] G. Sun, W. Wang, J. Dai, and L. Van Gool, "Mining cross-image semantics for weakly supervised semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, 2020, pp. 347–365.

[34] I. Alonso, A. Sabater, D. Ferstl, L. Montesano, and A. C. Murillo, "Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8219–8228.

[35] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7303–7313.

[36] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, "Rethinking semantic segmentation: A prototype view," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2582–2593.

[37] Z. Jin, T. Gong, D. Yu, *et al.*, "Mining contextual information beyond image for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7231–7241.

[38] Z. Jin, D. Yu, Z. Yuan, and L. Yu, "Mcibi++: Soft mining contextual information beyond image for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5988–6005, 2022.

[39] L. Yu, Z. Li, M. Xu, Y. Gao, J. Luo, and J. Zhang, "Distribution-aware margin calibration for semantic segmentation in images," *International Journal of Computer Vision*, vol. 130, pp. 95–110, 2022.

[40] Y. Wang, J. Fei, H. Wang, *et al.*, "Balancing logit variation for long-tailed semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 561–19 573.

[41] B. Han, Q. Xu, Z. Yang, *et al.*, "Aucseg: Auc-oriented pixel-level long-tail semantic segmentation," in *Advances in Neural Information Processing Systems*, 2024.

[42] C. Liang, W. Wang, J. Miao, and Y. Yang, "Gmmseg: Gaussian mixture based generative semantic segmentation models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 360–31 375, 2022.

[43] Z. Tian, J. Cui, L. Jiang, *et al.*, "Learning context-aware classifier for semantic segmentation," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023.

[44] X. Ma, Z. Ni, and X. Chen, "Ssa-seg: Semantic and spatial adaptive pixel-level classifier for semantic segmentation," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 93 690–93 713.

[45] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[46] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *European Conference on Computer Vision*, Springer, 2022, pp. 540–557.

[47] F. Liang, B. Wu, X. Dai, *et al.*, "Open-vocabulary semantic segmentation with mask-adapted clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7061–7070.

[48] M. Xu, Z. Zhang, F. Wei, *et al.*, "A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model," in *European Conference on Computer Vision*, Springer, 2022, pp. 736–753.

[49] J. Ding, N. Xue, G.-S. Xia, and D. Dai, "Decoupling zero-shot semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 583–11 592.

[50] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello, "Open-vocabulary panoptic segmentation with text-to-image diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2955–2966.

[51] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai, "Side adapter network for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2945–2954.

[52] Y. Liu, S. Bai, G. Li, Y. Wang, and Y. Tang, "Open-vocabulary segmentation with semantic-assisted calibration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3491–3500.

[53] B. Xie, J. Cao, J. Xie, F. S. Khan, and Y. Pang, "Sed: A simple encoder-decoder for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3426–3436.

[54] S. Roy and A. Etemad, "Consistency-guided prompt learning for vision-language models," *arXiv preprint arXiv:2306.01195*, 2023.

[55] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[56] H. Touvron, T. Lavril, G. Izacard, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[57] S. Pratt, I. Covert, R. Liu, and A. Farhadi, "What does a platypus look like? generating customized prompts for zero-shot image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 691–15 701.

[58] M. U. Khattak, M. F. Naeem, M. Naseer, L. Van Gool, and F. Tombari, "Learning to prompt with text only supervision for vision-language models," *arXiv preprint arXiv:2401.02418*, 2024.

[59] K. Roth, J. M. Kim, A. Koepke, O. Vinyals, C. Schmid, and Z. Akata, "Waffling around for performance: Visual classification with random words and broad concepts," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 746–15 757.

[60] X. Lai, Z. Tian, Y. Chen, *et al.*, "Lisa: Reasoning segmentation via large language model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9579–9589.

[61] J. Wang, K. Sun, T. Cheng, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[62] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 472–480.

[63] H. Zhang, C. Wu, Z. Zhang, *et al.*, "Resnest: Split-attention networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2736–2746.

[64] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[65] B. Zhan, E. Song, H. Liu, X. Xu, W. Li, and C.-C. Hung, "Segmenting medical images via explicit–implicit attention aggregation," *Knowledge-Based Systems*, vol. 279, p. 110 932, 2023.

[66] J. Chen, Y. Chen, W. Li, G. Ning, M. Tong, and A. Hilton, "Channel and spatial attention based deep object co-segmentation," *Knowledge-Based Systems*, vol. 211, p. 106 550, 2021.

[67] H. Zhao, Y. Zhang, S. Liu, *et al.*, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 267–283.

[68] S. Bai and C. Wang, "Information aggregation and fusion in deep neural networks for object interaction exploration for semantic segmentation," *Knowledge-Based Systems*, vol. 218, p. 106 843, 2021.

[69] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 418–434.

[70] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1209–1218.

[71] R. Mottaghi, X. Chen, X. Liu, *et al.*, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 891–898.

[72] M. Contributors, *MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark*, `https://github.com/open-mmlab/mmsegmentation`, 2020.

[73] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[75] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[76] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[77] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 548–557.

[78] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8950–8959.

[79] D. Shen, Y. Ji, P. Li, Y. Wang, and D. Lin, "Ranet: Region attention network for semantic segmentation," *NIPS*, vol. 33, pp. 13 927–13 938, 2020.

[80] Q. Song, J. Li, C. Li, H. Guo, and R. Huang, "Fully attentional network for semantic segmentation," in *AAAI*, vol. 36, 2022, pp. 2280–2288.

[81] S.-A. Liu, H. Xie, H. Xu, Y. Zhang, and Q. Tian, "Partial class activation attention for semantic segmentation," in *CVPR*, 2022, pp. 16 836–16 845.

[82] Y. Huang, D. Kang, L. Chen, *et al.*, "Car: Class-aware regularizations for semantic segmentation," in *European Conference on Computer Vision*, Springer, 2022, pp. 518–534.

[83] J. Chen, J. Lu, X. Zhu, and L. Zhang, "Generative semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7111–7120.

[84] B. Shi, D. Jiang, X. Zhang, *et al.*, "A transformer-based decoder for semantic segmentation with multi-level context mining," in *European Conference on Computer Vision*, Springer, 2022, pp. 624–639.

[85] H. Shi, M. Hayat, and J. Cai, "Transformer scale gate for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3051–3060.

[86] W. Wang, J. Dai, Z. Chen, *et al.*, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 408–14 419.

[87] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun, "Distribution alignment: A unified framework for long-tail visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2361–2370.

[88] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020.

[89] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[90] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.

[91] X. He, J. Liu, J. Fu, X. Zhu, J. Wang, and H. Lu, "Consistent-separable feature representation for semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 1531–1539.

[92] Q. Tang, C. Liu, F. Liu, *et al.*, "Rethinking feature reconstruction via category prototype in semantic segmentation," *IEEE Transactions on Image Processing*, vol. 34, pp. 1036–1047, 2025.

[93] H. Tang, Y. Zhao, C. Du, M. Xu, and Q. Wu, "Caa: Class-aware affinity calculation add-on for semantic segmentation," *Knowledge-Based Systems*, p. 112 097, 2024.

[94] F. Lin, Z. Liang, S. Wu, J. He, K. Chen, and S. Tian, "Structtoken: Rethinking semantic segmentation with structural prior," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 10, pp. 5655–5663, 2023. DOI: 10.1109/TCSVT.2023.3252807.

[95] S. Zheng, J. Lu, H. Zhao, *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[96] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu, "Segnext: Rethinking convolutional attention design for semantic segmentation," *Advances in neural information processing systems*, vol. 35, pp. 1140–1156, 2022.

[97] S. Quan, M. Hirano, and Y. Yamakawa, "Semantic information in contrastive learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5686–5696.

[98] Y. Shao, L. Sun, L. Jiao, *et al.*, "Cot: Contourlet transformer for hierarchical semantic segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, pp. 132–146, 2025.

[99] X. Shan, D. Wu, G. Zhu, Y. Shao, N. Sang, and C. Gao, "Open-vocabulary semantic segmentation with image embedding balancing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 412–28 421.

[100] S. Cho, H. Shin, S. Hong, A. Arnab, P. H. Seo, and S. Kim, "Cat-seg: Cost aggregation for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4113–4123.

[101] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[102] A. Yan, Y. Wang, Y. Zhong, *et al.*, "Learning concise and descriptive attributes for visual recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3090–3100.

[103] P. Kaul, W. Xie, and A. Zisserman, "Multi-modal classifiers for open-vocabulary object detection," in *International Conference on Machine Learning*, PMLR, 2023, pp. 15 946–15 969.

[104] O. Saha, G. Van Horn, and S. Maji, "Improved zero-shot classification by adapting vlms with text descriptions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 542–17 552.

[105] J. Wang, C. Yan, and G. Kang, "Rethinking the global knowledge of clip in training-free open-vocabulary semantic segmentation," *arXiv preprint arXiv:2502.06818*, 2025.

[106] T. Shao, Z. Tian, H. Zhao, and J. Su, "Explore the potential of clip for training-free open vocabulary semantic segmentation," in *European Conference on Computer Vision*, Springer, 2024, pp. 139–156.

[107] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[108] Q. Yu, J. He, X. Deng, X. Shen, and L.-C. Chen, "Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip," *Advances in Neural Information Processing Systems*, vol. 36, pp. 32 215–32 234, 2023.

[109] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata, "Semantic projection network for zero-and few-label semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8256–8265.

[110] M. Bucher, T.-H. Vu, M. Cord, and P. Pérez, "Zero-shot semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[111] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[112] C. Han, Y. Zhong, D. Li, K. Han, and L. Ma, "Open-vocabulary semantic segmentation with decoupled one-pass network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1086–1096.

[113] Z. Ding, J. Wang, and Z. Tu, "Open-vocabulary universal image segmentation with maskclip," *arXiv preprint arXiv:2208.08984*, 2022.

[114] S. Jiao, H. Zhu, J. Huang, Y. Zhao, Y. Wei, and S. Humphrey, "Collaborative vision-text representation optimizing for open-vocabulary segmentation," in *European Conference on Computer Vision*, 2024.

[115] Z. Zhao, X. Li, L. Shi, N. Imanpour, and S. Wang, "Dpseg: Dual-prompt cost volume learning for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2025, pp. 25 346–25 356.

[116] S. Bai, K. Chen, X. Liu, *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.