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

Lanyun Zhu, Tianrun Chen, Deyi Ji, Peng Xu, Jieping Ye, *Fellow, IEEE*, and Jun Liu

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

***Index Terms*—Few-shot segmentation, large language models.**

I. Introduction

Image segmentation is a fundamental task in computer vision with broad applications. The advent of deep learning algorithms trained by expansive datasets has brought significant progress to this domain. For example, trained on over 1 billion high-quality annotated images, the Segment Anything Model (SAM) [29] achieves class-agnostic segmentation with strong generalization capabilities. However, annotating pixel-level segmentation labels on such a large scale is extremely resource-intensive. Consequently, few-shot segmentation, a more resource-efficient learning paradigm that requires fewer annotated samples, has garnered increasing interest within the academic community and holds immense practical value.

In few-shot segmentation (FSS), a model is required to recognize and segment a novel class based on very few annotated examples, known as the support images. Motivated by the success of few-shot classification, the majority of FSS approaches typically adopt a support-feature-guided mechanism. This mechanism involves extracting representative features from the support images to assist in segmenting an unlabeled image, referred to as the query image. Several methods have been proposed to boost the effectiveness of this mechanism, focusing on enhancing the extraction method of support features [34], [57], [49] or improving how these features assist in segmenting the query images [23], [83], [92]. Although these methods have made some incremental improvements, their segmentation performance is still far from satisfactory. A critical factor contributing to this underperformance issue is their heavy dependence on a very limited set of support images, which can only provide a narrow, incomplete, and possibly biased set of information. Consequently, frameworks that depend exclusively on such restricted data are inherently limited by informational constraints, thus incapable of achieving sufficiently high accuracy. To address this issue, some recent methods [8], [68] employ foundation models, such as CLIP [60] and SAM [29], to enhance FSS performance by leveraging their pretrained image–text alignment or segmentation feature extraction abilities. However, due to the relatively limited number of model parameters for CLIP and the lack of class awareness of SAM, these foundation models are still unable to provide sufficient auxiliary information and lack inherent few-shot learning capabilities. In light of these limitations, we believe that the further advancement of FSS requires the development of an entirely new framework—one that can leverage a richer and more comprehensive set of information while offering enhanced few-shot learning capabilities, thus breaking through the limitations of existing paradigms and enabling superior performance.

In our LLaFS [103] published on CVPR 2024, we delved into the large language models (LLMs) and found them to be an effective foundation for developing such a brand-new framework with more information gotten involved. Specifically, we identified two properties of LLMs that motivated us to leverage them as a few-shot segmenter: (1) LLMs' extensive pretraining on broad textual corpora equips them with a vast amount of prior knowledge, which can effectively supplement the insufficient information in support images, thus providing enhanced guidance. (2) Furthermore, LLMs

Lanyun Zhu is with Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore, 487372 (e-mail: lanyun_zhu@mymail.sutd.edu.sg)

Tianrun Chen is with College of Computer Science and Technology, Zhejiang University, China, 310027 (email: tianrun.chen@zju.edu.cn)

Deyi Ji, Peng Xu and Jieping Ye are with Alibaba Cloud Computing, China, 310023 (email: jideyi.jdy@alibaba-inc.com, pengxin.xp@alibaba-inc.com, yejieping.ye@alibaba-inc.com)

Jun Liu is with School of Computing and Communications, Lancaster University, UK (e-mail: j.liu81@lancaster.ac.uk)

This work is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-08-006).

Corresponding author: Jun Liu.

have been demonstrated to be effective few-shot learners in NLP [4]. This success naturally inspires us to further extend their capabilities to few-shot tasks in other modalities. Drawing on these motivations, in [103], we introduced LLaFS, an innovative framework that pioneered applying LLMs to FSS and achieved SOTA results. Unlike some previous FSS methods that also use language models (LMs) but only for auxiliary purposes, such as utilizing LMs to extract text features [94], [74], our LLaFS is the first to leverage the more powerful large language models and directly employs LMs to generate segmentation results. This approach elevates LMs from a supportive position to a central role, making them no longer work as only auxiliary tools but unlocking their complete potential to perform complex vision tasks in an end-to-end manner. In this way, we provide a pioneering exploration towards creating a generalized framework that empowers LLMs to address few-shot learning challenges in other modalities beyond NLP.

In the research, we find that utilizing LLMs for FSS presents a lot of significant, non-trivial challenges that must be overcome. A primary issue is how to adapt LLMs, which are designed to produce text-based outputs, to the requirements of image segmentation that demands to output pixel-level binary masks. Drawing inspiration from previous work [76], we tackle this challenge by representing segmentation results as the vertices of 16-sided polygons and crafting an instruction within the LLM's input to explicitly define this format. This approach provides LLM with a clear hint about the task's definition and requirements, thereby prompting it to handle image segmentation more effectively and robustly. Another crucial challenge lies in how to effectively combine the visual information from support images with the textual information from LLMs to guide the segmentation of query images. Leveraging the LLMs' strong capacity for in-context learning, we treat support images as demonstration exemplars and introduce a region-attribute corresponding table as a more fine-grained multi-modal guidance. This table details specific attributes of the target class alongside their corresponding regions on the support image, thereby instructing the LLM to execute segmentation in a more fine-grained and human-like manner. Moreover, we notice the issue of training difficulty caused by the limited data and propose a pseudo-sample generation strategy to tackle it with a curriculum learning mechanism to facilitate optimization. By incorporating these innovative designs, our proposed LLaFS framework presents excellent effectiveness in handling few-shot segmentation.

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

We conduct extensive experiments across multiple datasets, and the results validate the outstanding performance of LLaFS and LLaFS++. On PASCAL-5^i, COCO-20^i, and FSS-1000, LLaFS++ achieves improvements of 3.6%, 8.3%, and 3.2% respectively compared to the previously reported best results. Moreover, LLaFS++ outperforms LLaFS by 3.7%, 3.6%, and 0.9% on these three datasets, highlighting the effectiveness of the proposed new techniques and improvements in this extended work. We also carry out comprehensive ablation studies to demonstrate the effectiveness and rationality of each module and design within our LLaFS++ framework. In summary, the main contributions of this paper are as follows:

- We propose LLaFS (and its extension LLaFS++), the first framework to address few-shot segmentation using large language models.
- We propose various innovative designs to make better use of LLMs in few-shot segmentation, including a task-tailored instruction, a fine-grained in-context instruction serving as multi-modal guidance, a pseudo-sample-based curriculum pretraining mechanism, and a novel inference method to mitigate prediction mistakes.
- Our approach achieves state-of-the-art performance on multiple datasets, with extensive experiments demonstrating the effectiveness of our designs.

As an extension of our previous conference paper [103], this paper introduces significant enhancements in four key aspects. *First*, we propose an improved methodology for the creation of the region-attribute corresponding table, which not only offers enhanced effectiveness but also requires reduced computational costs. (Sec.III-C3) *Second*, a new inference method is proposed to mitigate the issue identified in the original LLaFS framework, where inaccuracies in predicting localized regions instead of the whole object are observed. (Sec.III-F2) *Third*, we conduct more comprehensive experiments including additional datasets and more ablation studies. (Sec.IV) *Fourth*, the applicability of our method is assessed across more tasks, including few-shot object detection that extends beyond segmentation. The excellent performance across diverse tasks demonstrates the generalization of LLaFS++. (Sec.IV-G)

II. Related Work

A. Image Segmentation

Image segmentation is a fundamental task in the field of computer vision. Over the last decade, deep learning has brought significant advancements to this field, with techniques based on neural networks showcasing outstanding achievements across various sub-domains, such as semantic segmentation [7], [88], [106], [12], [36], [1], [104], [81], instance segmentation [17], [33], [27], [79], [87], and panoptic segmentation [70], [28], [10], [9], [51], [39]. A notable example is the Deeplab series [5], [6], [7], which employs atrous convolution to increase the size of the receptive field, enabling richer semantic information to be captured and preserving the high-resolution of feature maps to avoid boundary blurring. More recently, transformer-based approaches [88], [11], [12], [24], [91] have pushed the boundaries of segmentation performance even further. For instance, SegFormer [88] introduces an innovative pipeline that combines a hierarchical transformer encoder with an MLP-based decoder. Mask2Former [11] introduces masked attention to achieve faster convergence by constraining cross-attention within predicted mask regions. Despite remarkable results, their success heavily relies on the extensive segmentation annotations for training across all classes. Our research, different from the previously mentioned methods, focuses on the task of few-shot segmentation, which allows to segment a query image on a novel class using only a very small number of annotated support images, thus avoiding the huge cost for the extensive data annotation.

B. Few-Shot Segmentation

Few-shot segmentation (FSS) [75], [93], [43], [26], [47], [3], [31], [57], [32], [13], [71], [48], [105] has gained significant attention in recent years due to its ability to work well with only limited data, which is highly practical in real-world applications. [64] proposes the pioneering method in this field, where a feature is extracted from the labeled support images to generate a head that is then used to segment unlabeled query images. Building upon this framework, many existing methods [14], [31], [93], [55], [34], [23], [13], [82] adopt a prototype-guided strategy. These techniques employ masked average pooling (MAP) to derive global or local average prototypes from support image features, which then guide the segmentation of query images through various approaches such as feature fusion [34], [31], [46], distance measurement [49], [18], or attention-based mechanisms [59]. To avoid information loss caused by the prototype generation process, some more recent methods [98], [97], [73], [22], [89], [84] do not compress features into prototypes but instead retain the complete feature maps for per-pixel processing. For example, [92] proposes a self-calibrated cross-attention network with pixel-wise correlation extraction to solve the background mismatch and foreground-background entanglement issues. Other approaches [49], [66], [18], [90] try to extract the relationship between support and query image features in even greater detail. For instance, [49] proposes a hypercorrelation squeeze network that leverages efficient 4D convolutions to extract multi-level feature correlations. [80] formulates the segmentation task as an in-context coloring problem to improve the model's few-shot capability. While these methods have achieved some success, they can only leverage a limited amount of information extracted from a very small number of support images. Such a constraint may lead to suboptimal performance and decreased robustness. To mitigate this issue, [94] employs word embeddings from a language model as a more comprehensive source of class information to aid in segmentation, [8] leverages the image-text alignment capability of CLIP [60] to enhance segmentation performance. While these approaches introduce some enhancements, they remain limited by the relatively weak capabilities of small language models and lack an in-depth exploration of how to combine textual information with support image information more effectively for the improved guidance. In this work, we are the first to apply LLMs to few-shot segmentation by using our carefully designed instructions, which offer a more comprehensive and effective multimodal guidance system. Moreover, we leverage the LLM to directly output segmentation results rather than merely using the features of a language model as done in [94], [8]. This novel method introduces a brand-new paradigm for few-shot segmentation, exploring new possibilities for using LLMs in this research domain.

C. Large Language Models and Their Applications in Image Segmentation

The advent of large language models (LLMs) such as GPT [4] and Llama [72] has marked the beginning of a new era in artificial intelligence. Thanks to their significantly increased model parameters and training data, these LLMs contain rich prior knowledge and can be efficiently finetuned for specific tasks or application requirements through methods such as prompts [44], adapters [20] and LoRA [21]. Recently, researchers have started exploring visual large language models [35], [41], [76], [67], [2], [102] to establish a unified framework for multimodal data processing, aiming to override the restriction of LLMs being solely applicable to language data. However, most of these methods can only handle image-level understanding tasks, such as image captioning, visual question answering, etc., but are incapable of performing the more fine-grained segmentation tasks. Some more recent research [30], [62], [100], [76], [96], [61] has begun exploring how to extend the capabilities of LLMs to the field of image segmentation. For example, [30] proposes a large language instructed segmentation assistant to produce segmentation masks by incorporating an additional segmentation token into the existing vocabulary. [62] proposes an LLM-based segmentation model with a lightweight pixel decoder and a comprehensive segmentation codebook. [100] introduces a novel framework to handle unified segmentation by first generating mask proposals and then using LLMs to classify them. In our approach, we follow the strategy proposed by VisionLLM [76], which empowers LLMs to produce segmentation masks by generating the vertices of enclosing polygons. While all of the above methods are capable of performing image segmentation, our method differs from them significantly since none of them is designed specifically for few-shot segmentation. In contrast, LLaFS++

is the first LLM-based few-shot segmentation framework with several novel and task-tailored designs including: *(a) LLM's inputs.* Two novel instructions serving as LLM's inputs are proposed to extract rich information from the annotated support images in few-shot scenarios. *(b) Training data.* A novel method for synthesizing pseudo samples is proposed to solve the insufficient training data issue in few-shot segmentation. *(c) Optimization approaches.* A curriculum learning strategy is implemented to overcome slow convergence challenges. Incorporating these novel designs, LLaFS++ constitutes a brand-new framework that can effectively leverage information from both the annotated images and language priors to achieve high-quality few-shot segmentation.

D. Vision Foundation Models and Their applications for Zero-Shot Image Segmentation

Benefiting from the expansion of data scale and the advancement of computational power, vision foundation models, such as vision-language models [60], [41] and large-scale segmentation models [29], have made rapid progress in recent years. For example, CLIP [60] aligns text and image representations into the same space through multimodal contrastive learning. SAM [29] achieves highly generalized segmentation capabilities by training on extensive segmentation datasets. These progresses have further propelled the development of zero-shot and open-vocabulary segmentation [40], [95], [65], [91], [101], [25], [100], which aim to segment novel classes unseen during training. Some methods [40], [95], [91], [101], [25] leverage the pretrained vision-language alignment capabilities of CLIP to enable zero-shot segmentation. For example, OVSeg [40] segments unseen categories by aligning image features from cropped regions with text features of class names. Open-Vocabulary SAM [95] further improves performance by combining CLIP's alignment ability with SAM's powerful segmentation capabilities. More recently, researchers have begun exploring the use of the more effective large language models (LLMs) for zero-shot segmentation [65], [100]. For instance, LLMFormer [65] enhances segmentation performance by incorporating LLM-generated image descriptions as additional information. Our proposed LLaFS++ differs from these approaches by extending the setting from zero-shot to few-shot segmentation, where additional guidance from annotated support images can provide richer and more comprehensive information to further improve segmentation performance. To fully leverage the information in support images, our work further introduces a novel instruction design incorporating a carefully constructed region-attribute corresponding table, which effectively enhances the foundation model's segmentation capability and improves model performance.

III. Method

A. Overview

This paper aims to construct an LLM-based framework for few-shot segmentation, i.e., to segment a query image I_q based on N_s support images $\{I_s^n\}_{n=1}^{N_s}$ and their ground truth maps $\{G_s^n\}_{n=1}^{N_s}$.[1] As shown in Figure 1, the overall framework of LLaFS++ can be divided into three key components: (1) a feature extractor that extracts image features and generates visual tokens; (2) a task-tailored instruction that combines visual tokens, target categories, and task requirements to provide task-related information and support guidance; and (3) an LLM that predicts segmentation masks based on the input instruction and segmentation embeddings, followed by a refinement network to optimize the results. For the feature extractor, we adopt the approach in Blip2 [35] by using an image encoder followed by a Q-former and a fully-connected layer to generate a set of visual tokens. We use a frozen ResNet as the image encoder, which generate tokens from each input image. The Q-former [35] further compresses the number of such tokens and performs an initial interaction between the visual features and the textual information of the class name. For the instruction, we carefully design it as the combination of two parts: segmentation task instruction (Sec.III-B) and fine-grained in-context instruction (Sec.III-C) to provide comprehensive and detailed guidance. The instruction is concatenated with a set of learnable segmentation embeddings $\{\mathbf{P}_n\}_{n=1}^{N}$ (Sec.III-D) for inputting into the LLM. For the LLM, we employ CodeLlama [63] with 7 billion parameters that have been finetuned through instruction tuning. Note that compared to vanilla Llama, we empirically find that CodeLlama finetuned with code generation datasets exhibits higher accuracy and stability in generating structured information like the segmentation result in our task. We equip CodeLlama with LoRA for finetuning. All these components work together within the LLaFS++ framework to achieve high-quality few-shot segmentation.

As the input of LLM, the instruction is the most crucial component in our framework that makes LLM possible to handle few-shot segmentation. To provide comprehensive information, we design two instructions, namely segmentation task instruction and fine-grained in-context instruction, to respectively provide the LLM with detailed task definitions and fine-grained multi-modal guidance. These two instructions are integrated to formulate the complete instruction as shown in Figure 1. In the following Sec.III-B and Sec.III-C, we introduce these two instructions in detail.

B. Segmentation Task Instruction

The LLMs trained on massive text contents have gained strong reasoning capabilities and a vast amount of world knowledge. Language instructions have shown to be a powerful tool for leveraging this knowledge and capability to handle complex tasks [58]. To achieve better results, the instructions need to be sufficiently clear and detailed, whereas those using only simple terminologies such as 'performing image segmentation' are too abstract for LLMs to comprehend. Thus, we design a structured instruction to explicitly provide more task details such as the expected input and output formats of few-shot segmentation. Specifically, in our instruction, we follow [76] by representing the pixel-wise segmentation output

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

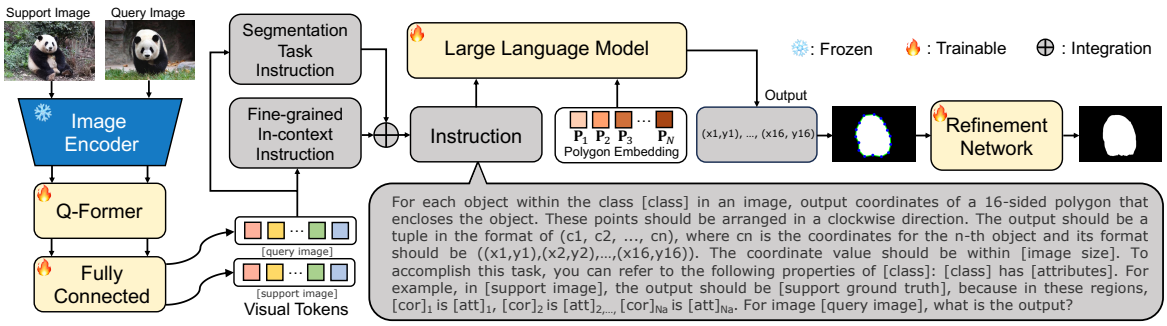


Fig. 1. **Overview of LLaFS++.** The image encoder and Q-former extract image features and generate a set of visual tokens. Subsequently, a segmentation task instruction and fine-grained in-context introduction are introduced to provide detailed and comprehensive information. These two instructions are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

as a set of 16-sided polygons that enclose the target objects [42]. Note that it is hard for LLMs to directly generate pixel-wise segmentation masks due to LLM's limited number of output tokens. Our alternative solution of generating polygon vertices provides a token-efficient method for using LLMs to achieve pixel-level segmentation.

Furthermore, the language-focused design of LLMs poses a challenge for their precise interpretation of visual information. This issue is particularly severe in few-shot image segmentation, where the availability of training images is extremely limited. To address this problem, inspired by the success of in-context learning in NLP [50], [15], we propose a novel strategy that encodes the support image along with its ground truth as a visual demonstration example. This example is then incorporated into the instruction, providing the LLM with a clear and intuitive reference that instructs the LLM on how to accurately segment a specific class within an image.

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

C. Fine-grained In-context Instruction

1) Motivation: The above task instruction makes segmenting a class possible by leveraging LLM's knowledge of the class. In the instruction, the class to be segmented is indicated by the [class] token, which is typically a single noun. However, considering that LLMs are language-based models mainly trained on text corpus, it is challenging for them to directly align this abstract noun with an image region that may possess a complex internal structure. To address this issue, we drew inspiration from human brains and found that when classifying an unseen new class, the human cognitive

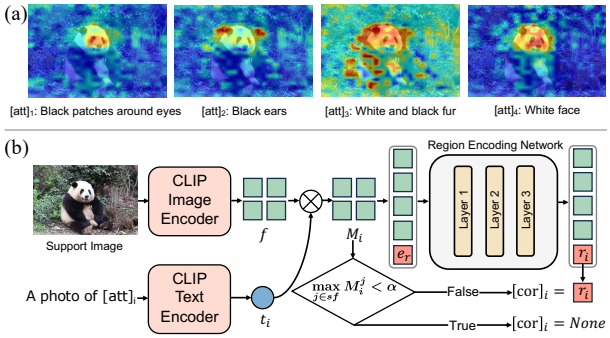


Fig. 2. (a) Examples of similarity maps M_i computed from the support image and class attributes. (b) Illustration of how to construct the region-attribute corresponding table for the i-th attribute $[\text{att}]_i$. sf refers to all pixels in support foreground. Note that the spatial shape of f and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

system follows a mechanism of '*from general to detailed, from abstract to concrete*' [85], [54]. Specifically, given an unseen class represented by a *general* noun, the human brain first decomposes it into *detailed* attributes based on the acquired knowledge. For example, in the case of an unseen class 'panda', a person can first gather information from references to learn about the panda's attributes such as 'black and white fur' and 'black ears'. Subsequently, it can search the image for *concrete* regions that match these *abstract* attributes to determine the presence of the class.

Motivated from the above discussion, we propose a fine-grained in-context instruction that leverages support images to simulate such a human cognitive mechanism. Specifically, we first instruct the LLM to extract detailed attributes of the target class (Sec.III-C2). Subsequently, we locate regions within support images that match these attributes and create a corresponding table accordingly (Sec.III-C3). This table, together with the extracted attributes, constitutes an in-context instruction (Sec.III-C4), which is then fed into the LLM to serve as a demonstration example that guides the LLM on how to recognize image classes in a more human-like and fine-grained manner. This approach effectively mitigates the limitations of LLMs in performing segmentation tasks based solely on generic class names. Furthermore, we also present

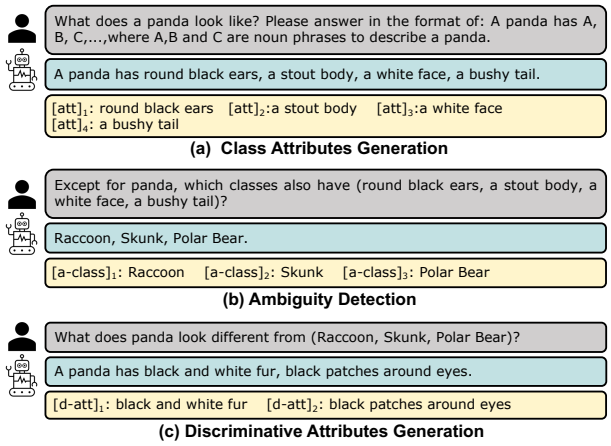


Fig. 3. Examples of using LLM for (a) class attributes generation, (b) ambiguity detection and (c) discriminative attributes generation.

an LLM-checking framework to refine the produced instructions (Sec.III-C5). In the following sections, we introduce the method for generating and refining the instruction in detail.

2) Attributes Extraction: We first simulate the step of '*from general to detailed*' to extract class attributes. Specifically, as shown in Figure 3(a), we construct a prompt '*What does a* [class] *look like? Please answer in the format of: A* [class] *has A, B, C,..., where A, B, and C are noun phrases to describe a* [class].', and instruct the LLM in LLaFS++ to extract phrases-based attributes that describe the fine-grained details of this class. These attributes are denoted as [attributes] = $\{[\text{att}]_i\}_{i=1}^{N_a}$. For each $[\text{att}]_i$, we utilize '*A photo of* $[\text{att}]_i$' as a prompt to extract an embedding t_i from the CLIP's text encoder. In this way, we get $\{t_i\}_{i=1}^{N_a}$ from $\{[\text{att}]_i\}_{i=1}^{N_a}$.

3) Region-attribute Corresponding Table: Considering that many attributes describe the locally regional characteristics of a category, for example, 'black ear' for 'panda', to obtain a more refined support guidance, we further simulate the second step of '*from abstract to concrete*' by identifying specific local regions within the support image that can be aligned with these class attributes, and then employing the alignments to construct a fine-grained in-context demonstration example. To implement this alignment, we introduce a simple yet effective method. Specifically, we first feed the support image into an enhanced CLIP image encoder proposed by [53] to produce a feature map $f \in \mathbb{R}^{H\times W\times C}$, where H, W and C represent the height, width, and channel number of f, respectively. Benefiting from its patch-level contrastive pretraining [53], this enhanced CLIP encoder excels in aligning text with specific local image regions. We then compute the cosine similarity between each pixel f^j within f and each attribute embedding t_i to produce a similarity map $M_i \in \mathbb{R}^{H\times W}$. As shown in Figure 2(a), it is encouraging to observe that this similarity map, although derived through a simple and straightforward method without complex post-processing, already exhibits a good level of attribute awareness, with regions corresponding to the attribute typically exhibiting higher similarities than the other areas. We further observe that some attributes, such as 'black and white fur' for 'panda', describe the wide-level properties of a class rather than specific details in local regions. In this case, M_i can still capture the presence of such attributes effectively, with a wide range of pixels across the entire foreground showing a high degree of similarity.

The next challenge involves how to encode the attribute-corresponding region captured by M_i into a format that the LLM can receive as input. To tackle this issue, we introduce a lightweight region encoding network (REN) designed to convert M_i into an implicit feature. As shown in Figure 2(b), the REN is structured with three serial transformer layers, with the input being the concatenation of the similarity map M_i and a learnable region embedding e_r. e_r's hidden state r_i at the output of the transformer is utilized as a feature to represent $[\text{att}]_i$'s corresponding region in the support image. Note that in the transformer, we employ masked attention [11] rather than the vanilla self-attention to focus REN on the support foreground area that belongs to the target class. During the training process, REN is optimized end-to-end in sync with the LLM. We also notice that not every attribute extracted through Sec.III-C2 can find a corresponding region on the support foreground, mainly due to the variations in camera angles and instances of occlusion. To prevent introducing misleading information, we use a simple thresholding approach to filter out feature r calculated from these support-non-corresponding attributes. In this way, we establish region-attribute correspondence $[\text{cor}]_i$ for each attribute $[\text{att}]_i$ by:

$$[\text{cor}]_i = None \text{ if } \max_{j\in sf} M_i^j < \alpha \text{ else } r_i, \tag{1}$$

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

It is crucial to highlight that the aforementioned technique for creating the region-attribute corresponding table is simpler yet more effective than the method utilized in LLaFS [103]. As mentioned in Sec.I, the approach used in LLaFS requires the extraction of CLIP features from cropped images, which compromises the region-attribute alignment accuracy due to the loss of context information. In contrast, the improved approach presented in this paper eliminates this need for image cropping and achieves better performance as demonstrated by the experimental results shown in Table VI.

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

target class. With such an example as reference, the LLM can be taught how to understand and segment an image class in a fine-grained and human-like manner.

5) Instruction Refinement: The above-introduced instruction, which is constructed by the extracted attributes $\{[\text{att}]_i\}_{i=1}^{N_a}$ and table $\{[\text{cor}]_i\}_{i=1}^{N_a}$, can be directly fed into LLM for guidance. However, we have identified potential issues that directly combining the attributes derived from Sec.III-C2 may introduce class ambiguities due to the shared attributes across different classes. For example, the combination of attributes 'wheels, windows, doors' might be extracted for the 'train' class but could also refer to other classes such as 'bus' and 'car'. Furthermore, since attributes not corresponding to the support image have been filtered out through Eq.1, the generated table $\{[\text{cor}]_i\}_{i=1}^{N_a}$ may represent regions for only a subset of attributes within $\{[\text{att}]_i\}_{i=1}^{N_a}$. The combination of these partial attributes is consequently more susceptible to class ambiguities, and thus making the resultant instruction to be confusing and misleading.

To alleviate the aforementioned issue, we propose an LLM-checking framework to refine the instruction. This framework identifies potential ambiguous classes for the existing attributes, and subsequently extracts additional attributes with higher class discrimination ability to mitigate the ambiguity problem. Specifically, the instruction refinement is implemented through the following three steps: ***1) Ambiguity Detection.*** As shown in Figure 3(b), we instruct the LLM to identify potential ambiguous classes in the obtained table $\{[\text{cor}]_i\}_{i=1}^{N_a}$. Specifically, we denote the set of all attributes with a non-empty $[\text{cor}]_i$ as [valid-att] and ask the LLM *'Except for* [class]*, which classes also have* [valid-att]*?'*[2] In this way, we obtain a set of ambiguous classes denoted as [a-classes]=$\{[\text{a-class}]_i\}_{i=1}^{N_{ac}}$ from LLM's feedback. ***2) Discriminative Attributes Generation.*** As shown in Figure 3(c), to avoid being misled by these ambiguous classes, we use *'What does* [class] *look different from* [a-classes]*?'* as a text prompt, enabling the LLM to generate attributes that are more discriminative from the ambiguous classes. The obtained attributes $\{[\text{d-att}]_i\}_{i=1}^{N_d}$ are added to [attributes] for updating. ***3) Table and Instruction Refinement.*** Finally, using the updated attributes, we generate a refined table by reperforming Eq.1. The updated attributes and table are reassembled through the way in Sec.III-C4 to obtain a refined instruction.

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

[2]We also add a format control prompt for asking the LLM. Please see Appendix.A-B for details.

D. Segmentation Prediction

We integrate segmentation task instruction and fine-grained in-context instruction to formulate the complete instruction as shown in Figure 1. With this instruction as input, the LLM can predict the vertex coordinates of 16-sided polygons that surround the target objects. Specifically, as shown in Figure 1 and inspired by MaskFormer [12], we introduce N sets of polygon embeddings $\{\mathbf{P}_n\}_{n=1}^{N}$, each consisting of 33 learnable embeddings $(\mathbf{x}_n^1, \mathbf{y}_n^1, \mathbf{x}_n^2, \mathbf{y}_n^2, ..., \mathbf{x}_n^{16}, \mathbf{y}_n^{16}, \mathbf{v}_n)$. Each $\mathbf{P}_n$ is concatenated with the instruction and fed into the LLM. The LLM's outputs for $(\mathbf{x}_n^1, \mathbf{y}_n^1, \mathbf{x}_n^2, \mathbf{y}_n^2, ..., \mathbf{x}_n^{16}, \mathbf{y}_n^{16})$ represent the x- and y-coordinates of the polygon's 16 vertices, and the hidden states of $\mathbf{v}_n$ from the LLM's final layer are passed through a fully connected layer f_v to produce a validity score v_n. During training, we first apply bipartite matching to align all the LLM-predicted polygons with the ground truth mask. For polygons matched to the ground truth mask, the corresponding v_n is optimized to be as large as possible; while for unmatched polygons, v_n is optimized to be as small as possible. Through this training process, the validity score v_n learns to reflect the likelihood that the polygon produced by $\mathbf{P}_n$ can accurately represent the target object in the query image. Please refer to Appendix.A-F for the detailed optimization loss. Moreover, to rectify the imprecision caused by the polygon representation of object edges, we introduce a refinement network that comprises a pixel decoder and a mask transformer to generate a refined segmentation mask by using these polygons as the initial masks. Please see Appendix.A-C for the detailed structures of this network. Note that this refinement network is only an optional component that can further improve performance. Excluding it from LLaFS++ and directly using the LLM-generated polygons as the final segmentation mask is completely acceptable, and it can still achieve the SOTA performance.

E. Curriculum Pretraining with Pseudo Samples

1) Motivation: After carefully designing the model structure and instruction format, the next challenge is how to train LLaFS effectively to achieve high-quality segmentation results. Previous work [41] has highlighted that the success of LLMs typically relies on the training on extensive data. However, due to the challenge of acquiring pixel-annotated labels, the datasets for training in segmentation often have a limited number of images. To mitigate this limitation, we propose an innovative solution that generates pseudo support-query pairs for pretraining the LLM. The LLM's ability to handle few-shot segmentation can thus be enhanced by seeing more visual samples with segmentation annotations.

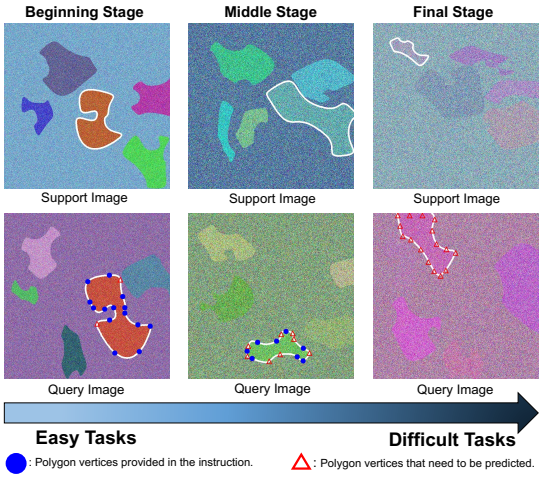


Fig. 4. Examples of pseudo samples generated at different pretraining stages. Foreground regions are marked by white contours. As pretraining progresses, pseudo images have reduced intra-image foreground-background differences and greater support-query foreground differences. Meanwhile, the number of polygon vertex coordinates provided in the instruction decreases, while the predicted vertex count increases. These changes gradually increase the pretraining difficulty. (Best viewed in color)

2) Pseudo Sample Generation: Specifically, we propose a method to generate pseudo support-query pairs with the following three steps: ***1) Pseudo foreground-background partition.*** We first use bezier curves to generate a random contour within a black image. The area surrounded by this contour is considered as the foreground within the target class, while the regions outside the contour are treated as the background. ***2) Noise filling for pseudo support generation.*** We fill the foreground with Gaussian noise that has a random mean value. For background, we first randomly divide it into multiple subregions, aiming to simulate the complex backgrounds in real images. Each subregion is then filled with Gaussian noise that has a random mean value different from the foreground noise. The resulting image is utilized as the support image. ***3) Pseudo query generation.*** We use the same approach to generate a query image. Note that in this process, the contour and the mean value of the foreground noise are not entirely random but are instead adjusted according to those for creating the support image. This is done to ensure that the foreground regions of both the support and query images exhibit similar contour shapes and internal characteristics, so that they can represent the same category. Please refer to Appendix.A-D2 for more pseudo support-query generation details.

3) Curriculum Pretraining: The synthetic support-query pairs can be directly used for pretraining. However, this straightforward method is observed to yield a slow rate of convergence. One potential explanation for this issue is that the LLM, given its language-based nature, may face difficulties in optimizing for a complex image processing task. To address this issue, we propose a progressive pretraining approach inspired by the success of curriculum learning [78], in which we initiate the model's pretraining with a simple task and gradually increase the task's difficulty until it ultimately reaches the requirements of segmentation.

Specifically, as show in Figure 4, during pretraining, we incrementally raise the task's difficulty from the following two aspects: ***1) Image understanding.*** During pretraining, by controlling the difference between mean values of different filled noise, we gradually increase the difference in foreground between the synthetic support and query, while reducing the internal difference between foreground and background within each image. This strategy incrementally increases the challenge for the LLM to execute few-shot guidance and distinguish between foreground and background areas as pretraining progresses. ***2) Polygon generation.*** Generating a polygon represented by a combination of vertex coordinates is observed to be another challenge for the LLM. Therefore, we also apply a progressive strategy to this aspect. Specifically, instead of pretraining the model to directly predict the coordinates of a polygon's all 16 vertices, we randomly provide the coordinates of K vertices in the instruction, leaving the LLM to predict the coordinates of the remaining $16-K$ vertices. During pretraining, we gradually reduce K from 15 to 0. This incremental reduction means that the model receives fewer hints and is required to predict more vertex coordinates as pretraining progresses. Consequently, the pretraining difficulty gradually increases, ultimately reaching the task of predicting all 16 vertices for segmentation. Experimental results show that this curriculum learning approach allows the model to converge better and achieve higher results. Please see Appendix.A-D3 for more technical details on how we increase the difficulty in image understanding and polygon generation.

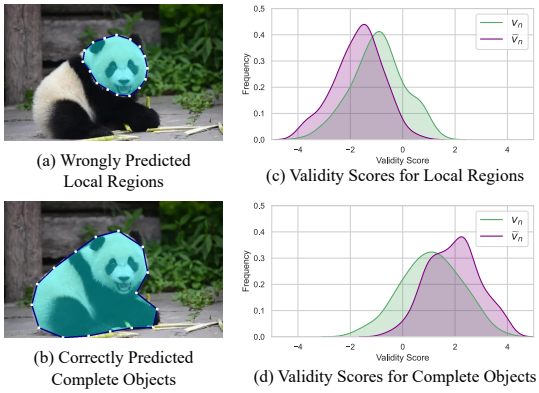
Ultimately, the model is trained on the realistic few-shot segmentation dataset after completing the aforementioned pretraining process. We will illustrate the detailed training procedures in the following section.

F. Training and Inference

After introducing our innovative designs within the LLaFS++ framework, in this section, we further elaborate on the complete process of model training and inference methods. Specifically, we follow previous works by using a multi-stage training strategy (Sec.III-F1), and propose a novel inference method to address potential hallucination issues when executing the LLaFS++ framework (Sec.III-F2).

1) Training: Following the method of Blip2 [35], which commences by training the Q-former independently before jointly training it with the LLM, we train the LLaFS++ framework using three stages, with distinct components targeted at each stage. In the first stage, we freeze the LLM, pretrain the Q-former and fully-connected layers for 100K steps (1 epoch) with a batch size of 128 using the image captioning datasets[3] and methods in Blip2 [35], with the aim of enabling the LLM to acquire the capability to process visual images. Note that incorporating image captioning datasets into the training process is a strategy widely adopted by a lot of LLM-based segmentation methods such as LISA [30] and VisionLLM [76]. Thus, we employ the same method as well. Another point worth mentioning is that we found models trained directly with the original image captioning dataset exhibit poor spatial locality awareness, which is detrimental to image

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



(a) Wrongly Predicted Local Regions
(c) Validity Scores for Local Regions
(b) Correctly Predicted Complete Objects
(d) Validity Scores for Complete Objects

Fig. 5. Illustration of the oversegmentation hallucination problems (a) and the distribution of v_n (green lines) and $\widetilde{v}_n$ (purple lines) for local regions (c) and complete objects (d). Best viewed in color.

segmentation. To address this issue, we employ a simple data augmentation method, which adds noise to a random region of each training image, and then modifies the image's caption to include description of the noisy region's spatial location. Training with such augmented data helps improve the spatial awareness of the model and thus improving segmentation performance. Please see Appendix.A-E for more details of this augmentation method. In the second stage, we freeze the Q-former, equip the LLM with LoRA, and pretrain the fully-connected layers, LLM and refinement network using the pseudo-sample-based curriculum learning method (Sec.III-E) for 60k steps (1 epoch) with a batch size of 32. In the third stage, we fine-tune the fully-connected layers, LLM and refinement network on the realistic few-shot segmentation dataset (25 epochs for PASCAL-5^i and FSS-1000, 3 epochs for COCO-20^i) with a batch size of 32. The number of epochs is set based on experimental results shown in Appendix.C6. The loss functions are detailed in Appendix.A-F.

2) Inference with Hallucination Mitigation: During the inference stage, when segmenting a target class, we follow the same setting as previous methods [48], [45], [31], [92], treating an annotated image of this class as the support image and the test image to be segmented as the query image. We feed these images into the LLaFS++ framework as shown in Figure 1, using the generated polygons with validity scores $v_n > 0$ as the initial segmentation results and the refined mask from the refinement network as the final results. While this straightforward method typically produces satisfactory outcomes, as shown in Figure 5(a), we occasionally face a hallucination issue where the model incorrectly segments incomplete local regions of the target object rather than capturing its entirety. This problem may arise from the utilization of the fine-grained in-context instruction as described in Sec.III-C, where some locally regional characteristics of the target class and their corresponding local regions within the support image are input into the LLM for guidance, which could potentially mislead the LLM to focus on segmenting only local regions of the target class, thus resulting in the hallucination issue. To address this problem, we introduce a contrastive prediction approach for model inference. Specifically, consider the LLM ϕ with L layers; denote the first $L-1$ layers of the LLM as $\phi_{[1:L-1]}$ and the final layer as $\phi_{[L]}$. Within $\phi_{[L]}$, we use self-attention masks to block all hidden states of [class] tokens and [support ground truth] tokens, thereby preventing other tokens from seeing these tokens describing the target class's global information. In this way, we create a locally-biased layer denoted as $\hat{\phi}_{[L]}$. For the n-th polygon embedding $\mathbf{P}_n$, we denote its validity score (refer to Sec.III-D for details) computed from $[\phi_{[1:L-1]}, \phi_{[L]}]$ as v_n and that from $[\phi_{[1:L-1]}, \hat{\phi}_{[L]}]$ as $\hat{v}_n$. During the inference stage, instead of using the original v_n, we calculate another score $\widetilde{v}_n$ as follows to verify the validity of the polygon:

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

Subsequently, polygons with $\widetilde{v}_n < 0$ are excluded. This procedure incorporates a contrastive element $(v_n - \hat{v}_n)$ during the inference phase, a strategy drawing inspiration from the success of contrastive decoding [38] in NLP. Specifically, within the LLM's input instructions, the [class] tokens denote the class name, while the [support ground truth] tokens describe the entire area belonging to the target class within the support image. These tokens collectively represent the holistic or global information of the target class. When such global tokens are masked out within $\hat{\phi}_{[L]}$, the remaining information is primarily related to the localized attributes of the target class. Relying solely on such locally-focused information inherently increases the validity scores of the predicted local regions, while reducing those for polygons that enclose the entire target object. Therefore, a higher contrastive value $(v_n - \hat{v}_n)$ can reflect a greater possibility that the n-th polygon represents the entire object. By combining this contrastive element with v_n and using the combined score $\widetilde{v}_n$ to assess the polygons, we can filter out those that represent just local regions of the target object, thus mitigating the aforementioned hallucination problem. The score distributions shown in Figure 5 demonstrate the effectiveness of $\widetilde{v}_n$, which shows that compared to v_n, a greater number of $\widetilde{v}_n$ for the incorrectly predicted local regions fall below 0. Conversely, for the correctly predicted complete objects, the distribution exhibits an opposite trend. Note that our inference method does not require adding any extra parameters; it only necessitates a slight 3% increase in computational cost for an additional run of the LLM's last layer. Therefore, our approach can enhance performance at almost zero cost, as demonstrated by the experimental results presented in Table VI.

IV. Experiments

A. Datasets and Metrics

We evaluate our method on three commonly used datasets: PASCAL-5^i [64], COCO-20^i [55], and FSS-1000 [37]. The PASCAL-5^i dataset is comprised of images sourced from the PASCAL VOC 2012 dataset with annotations extended by the SDS dataset. COCO-20^i is proposed in [55] and built based on MSCOCO. Following previous work [103], [92], [48], [13], we employ a cross-validation strategy for our experiments. Specifically, we divide the total classes of each dataset into four equal subsets, using three subsets for training and the remaining one subset for testing in each experiment. In this way, for

TABLE I
PERFORMANCE COMPARISON WITH OTHER METHODS ON PASCAL-5^i

Backbone	Method	1-shot						5-shot					
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU	Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU
ResNet50	NTRENet [45]	65.4	72.3	59.4	59.8	64.2	77.0	66.2	72.8	61.7	62.2	65.7	78.4
	BAM[31]	69.0	73.6	67.6	61.1	67.8	79.7	70.6	75.1	70.8	67.2	70.9	82.2
	AAFormer[84]	69.1	73.3	59.1	59.2	65.2	73.8	72.5	74.7	62.0	61.3	67.6	76.2
	SSP[14]	60.5	67.8	66.4	51.0	61.4	-	67.5	72.3	75.2	62.1	69.3	-
	IPMT[46]	72.8	73.7	59.2	61.6	66.8	77.1	73.1	74.7	61.6	63.4	68.2	81.4
	ABCNet[83]	68.8	73.4	62.3	59.5	66.0	76.0	71.7	74.2	65.4	67.0	69.6	80.0
	HDMNet [59]	71.0	75.4	68.9	62.1	69.4	-	71.3	76.2	71.3	68.5	71.8	-
	MIANet[94]	68.5	75.8	67.5	63.2	68.7	79.5	70.2	77.4	70.0	68.8	71.6	82.2
	MSI[52]	71.0	72.5	63.8	65.9	68.3	79.1	73.0	74.2	66.6	70.5	71.1	81.2
	SCCAN[92]	68.3	72.5	66.8	59.8	66.8	77.7	72.3	74.1	69.1	65.6	70.3	81.8
	AMFormer[82]	71.1	75.9	69.7	63.7	70.1	-	73.2	77.8	73.2	68.7	73.2	-
	HPA[13]	67.5	72.4	65.2	56.7	65.5	76.4	71.2	73.9	68.8	63.8	69.4	81.1
	BAM-final[32]	69.2	74.7	67.8	61.7	68.3	80.3	71.8	75.7	72.0	67.5	71.8	83.1
	PFENet++[48]	63.3	71.0	65.9	59.6	64.9	76.8	66.1	75.0	74.1	64.3	69.9	81.1
	PGMA-Net[8]	73.4	80.8	70.5	71.7	74.1	83.5	74.0	81.5	71.9	73.3	75.2	84.2
	LLaFS [103]	74.2	78.8	72.3	68.5	73.5	84.8	75.9	80.1	75.8	70.7	75.6	85.3
	LLaFS++	**77.8**	**82.1**	**75.8**	**72.9**	**77.2**	**86.7**	**79.7**	**83.6**	**77.9**	**73.8**	**78.8**	**87.7**
ResNet101	NTRENet[45]	65.5	71.8	59.1	58.3	63.7	75.3	67.9	73.2	60.1	66.8	67.0	78.2
	DCAMA[66]	65.4	71.4	63.2	58.3	64.6	77.6	70.7	73.7	66.8	61.9	68.3	80.8
	VAT[18]	70.0	72.5	64.8	64.2	67.9	79.6	75.0	75.2	68.4	69.5	72.0	83.2
	ABCNet[83]	65.3	72.9	65.0	59.3	65.6	78.5	71.4	75.0	68.2	63.1	69.4	80.8
	MSI[52]	73.1	73.9	64.7	68.8	70.1	82.3	73.6	76.1	68.0	71.3	72.2	82.3
	SCCAN[92]	70.9	73.9	66.8	61.7	68.3	78.5	73.1	76.4	70.3	66.1	71.5	82.1
	AMFormer[82]	71.3	76.7	70.7	63.9	70.7	-	74.4	78.5	74.3	67.2	73.6	-
	HPA[13]	67.2	73.1	64.3	59.8	66.1	76.6	68.3	75.2	66.4	67.8	69.4	80.4
	BAM-final[32]	69.9	75.4	67.1	62.1	68.6	80.2	72.6	77.1	70.7	69.8	72.5	84.1
	PFENet++[48]	63.1	72.4	63.4	62.2	65.3	75.5	67.2	76.1	75.5	67.2	71.5	82.7
	PGMA-Net[8]	76.8	82.3	75.7	**75.7**	77.6	86.2	77.7	82.7	76.9	**77.0**	78.6	86.9
	LLaFS	75.0	79.3	72.9	69.4	74.1	85.1	77.0	81.1	76.5	72.1	76.7	85.8
	LLaFS++	**78.8**	**82.4**	**76.2**	73.2	**77.7**	**87.2**	**80.5**	**84.4**	**78.7**	74.8	**79.6**	**88.4**

TABLE II
PERFORMANCE COMPARISON WITH OTHER METHODS ON COCO-20^i.

Backbone	Method	1-shot						5-shot					
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU	Fold-0	Fold-1	Fold-2	Fold-3	Mean	FB-IoU
ResNet50	NTRENet[45]	36.8	42.6	39.9	37.9	39.3	68.5	38.2	44.1	40.4	38.4	40.3	69.2
	BAM[31]	43.4	50.6	47.5	43.4	46.2	67.4	49.3	54.2	51.6	49.6	51.2	71.9
	SSP[14]	35.5	39.6	37.9	36.7	37.4	-	40.6	47.0	45.1	43.9	44.1	-
	AAFormer[84]	39.8	44.6	40.6	41.4	41.6	67.7	42.9	50.1	45.5	49.2	46.9	68.2
	MM-Former[99]	40.5	47.7	45.2	43.3	44.2	-	44.0	52.4	47.4	50.0	48.4	-
	IPMT[46]	41.4	45.1	45.6	40.0	43.0	-	43.5	49.7	48.7	47.9	47.5	-
	ABCNet[83]	42.3	46.2	46.0	42.0	44.1	69.9	45.5	51.7	52.6	46.4	49.1	72.7
	HDMNet [59]	43.8	55.3	51.6	49.4	50.0	72.2	50.6	61.6	55.7	56.0	56.0	77.7
	MIANet[94]	42.5	53.0	47.8	47.4	47.7	71.5	45.8	58.2	51.3	51.9	51.7	73.1
	MSI[52]	42.4	49.2	49.4	46.1	46.8	-	47.1	54.9	54.1	51.9	52.0	-
	SCCAN[92]	40.4	49.7	49.6	45.6	46.3	69.9	47.2	57.2	59.2	52.1	53.9	74.2
	AMFormer[82]	44.9	55.8	52.7	50.6	51.0	72.9	52.0	61.9	57.4	57.9	57.3	78.8
	HPA[13]	41.0	46.9	44.3	43.2	43.8	68.3	46.2	56.2	49.2	50.4	50.5	71.4
	BAM-final[32]	43.9	51.4	47.9	44.5	46.9	72.3	49.8	55.4	52.3	50.2	51.9	74.7
	PFENet++[48]	40.9	46.0	42.3	40.1	42.3	65.7	47.5	53.3	47.3	46.4	48.6	70.3
	LLaFS [103]	47.5	58.8	56.2	53.0	53.9	75.2	53.2	63.8	63.1	60.0	60.0	79.5
	LLaFS++	**50.8**	**62.7**	**60.2**	**56.4**	**57.5**	**78.8**	**53.9**	**64.9**	**63.8**	**61.1**	**60.9**	**79.9**
ResNet101	NTRENet[45]	38.3	40.4	39.5	38.1	39.1	67.5	42.3	44.4	44.2	41.7	43.2	69.6
	SSP[14]	39.1	45.1	42.7	41.2	42.0	-	47.4	54.5	50.4	49.6	50.2	-
	IPMT[46]	40.5	45.7	44.8	39.3	42.6	-	45.1	50.3	49.3	46.8	47.9	-
	ABCNet[83]	36.5	35.7	34.7	31.4	34.6	59.2	40.1	40.1	39.0	35.9	38.8	62.8
	MSI[52]	44.8	54.2	52.3	48.0	49.8	-	49.3	58.0	56.1	52.7	54.0	-
	SCCAN[92]	42.6	51.4	50.0	48.8	48.2	69.7	49.4	61.7	61.9	55.0	57.0	74.8
	AMFormer[82]	40.5	45.7	44.8	39.3	42.6	-	45.1	50.3	49.3	46.8	47.9	-
	HPA[13]	43.2	50.5	45.5	46.2	46.3	68.8	49.4	58.4	52.5	50.9	52.8	74.4
	BAM-final[32]	45.2	55.1	48.7	45.0	48.5	69.9	48.3	58.4	52.7	51.4	52.7	74.1
	PFENet++[48]	42.0	44.1	41.0	39.4	41.6	65.4	47.3	55.1	50.1	50.1	50.7	70.9
	LLaFS [103]	48.1	59.3	56.5	53.6	54.4	75.6	53.2	64.1	63.3	60.2	60.2	79.6
	LLaFS++	**51.1**	**63.0**	**61.4**	**56.9**	**58.1**	**79.2**	**54.2**	**65.2**	**63.9**	**61.3**	**61.1**	**80.0**

TABLE III
PERFORMANCE COMPARISON ON FSS-1000

Backbone	Method	mIoU	
		1-shot	5-shot
ResNet50	MSI[52]	90.0	90.6
	PFENet++[48]	88.6	89.1
	LLaFS[103]	92.3	92.8
	LLaFS++	**93.2**	**93.5**
ResNet101	MSI[52]	90.6	91.0
	PFENet++[48]	88.6	89.2
	LLaFS[103]	92.7	93.0
	LLaFS++	**93.4**	**93.8**

TABLE IV
COMPARISON WITH LLM-BASED SEGMENTATION METHODS

Method	PASCAL-5^i		COCO-20^i		FSS-1000	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
LISA[30]	70.2	73.7	51.9	58.0	90.5	91.1
PixelLM[62]	69.5	73.2	51.0	57.2	90.0	90.5
LLaFS[103]	74.1	76.7	54.4	60.2	92.7	93.0
LLaFS++	**77.7**	**79.6**	**58.1**	**61.1**	**93.4**	**93.8**

PASCAL-5^i, we have 15 classes for training and 5 classes for testing, and for COCO-20^i, we have 60 classes for training and 20 classes for testing in each experiment. This approach results in four sets of experimental results along with their mean result for each dataset. The FSS-1000 dataset contains images of 1000 classes, of which 486 classes are new classes not present in previous benchmarks. The overall classes in FSS-1000 are divided into 520, 240, and 240 classes for training, validation, and testing, respectively. Following previous methods [48], we

report the results on the test set. We use two widely-adopted metrics for evaluation, including mean intersection-over-union (mIoU) and foreground-background IoU (FB-IoU).

B. Implementation Details

We set the threshold α in Eq.1 to 0.22, and the number N of polygon embedding $\mathbf{P}_n$ to 15. The ground truth of polygon vertices is obtained in polar coordinates [86]. Specifically, starting from the object center, 16 rays are uniformly emitted at equal angular intervals $\triangle\theta = 22.5^\circ$. The points of intersection between these rays and the object contour are taken as the ground truth of the polygon vertices. AdamW is used as the optimizer with the cosine annealing schedule and an initial learning rate of 0.0002. The model is trained on A100 GPUs. More implementation details about model structures and training settings are presented in Appendix.B.

C. Main Results

1) Comparison with Few-shot Segmentation methods: In this section, we compare our method with existing state-of-the-art few-shot segmentation techniques on three datasets: PASCAL-5^i, COCO-20^i, and FSS-1000, with the results presented in Table I, Table II, and Table III, respectively. To evaluate the generalization capabilities of our approach, we report comparative results utilizing two different backbone scales: ResNet50 and ResNet101. All the compared methods are advanced approaches published at top conferences (CVPR, ICCV, etc.) or in top journals (T-PAMI, etc.) within the recent two years. The results of the compared methods are directly taken from their original publications. Our method displays superior performance across most datasets and experimental settings, consistently outperforming the existing approaches and showing a significant enhancement over the previous SOTA results. For instance, using the ResNet50 backbone on the PASCAL-5^i dataset, LLaFS [103] reaches an mIoU of 73.5% and an FB-IoU of 84.8%. In this paper, we introduce LLaFS++, which incorporates additional improvements and thus achieving an even higher mIoU of 77.2% and FB-IoU of 86.7% in the 1-shot scenario, which surpasses LLaFS by 3.7% and 1.9%, and outperforms the previous best results by 3.1% and 3.2%, respectively. Considering the more demanding COCO-20^i dataset, which presents a bigger challenge due to a greater number of classes and more diverse images, our method shows even higher advantages, outperforming the previous state-of-the-art techniques by 8.3% in mIoU and 9.3% in FB-IoU for the 1-shot scenario using the ResNet101 backbone. Furthermore, LLaFS++ also shows significant advantages on the FSS-1000 dataset, surpassing the second-best method by 3.2% in the one-shot setting with the ResNet50 backbone. Compared to other benchmarks, FSS-1000 includes a broader range of categories. The strong performance on this dataset highlights the powerful category generalization ability of LLaFS++ enabled by the extensive knowledge of LLMs. It is worth noting that MIANet [94] and PGMA-Net [8], two methods we compare against, also employ language models (word2vec) or vision-language models (CLIP) to facilitate few-shot segmentation. Our method, different from them, leverages the more powerful large language model (LLM) with several innovative and task-specific designs that enhance the LLM's capability in addressing the few-shot segmentation problem. Particularly, our fine-grained in-context instruction delves deeper into how to better integrate textual and visual information from language models and annotated images for getting a better multimodal guidance. With these novel designs, our method significantly outperforms MIANet and PGMA-Net by 8.5% and 3.1% on PASCAL-5^i, respectively. Furthermore, using the ViT backbone, we compare our method with SegGPT [80], a general in-context segmentation framework, and our method still achieves superior performance (see Appendix.C1 for details). These results demonstrate the excellent performance of our LLaFS++ and highlight the huge potentiality of using LLMs to tackle few-shot segmentation.

TABLE V
EFFECTIVENESS OF DIFFERENT COMPONENTS IN THE LLAFS FRAMEWORK.

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

2) Comparison with LLM-based Segmentation Methods: We further compare our method with other LLM-based segmentation techniques to highlight the superior advantages of our LLaFS++ framework. We choose two recently published advanced algorithms from CVPR 2024 for this comparison: LiSA [30] and PixelLM [62] and the comparative results are shown in Table IV. Given that these methods are not originally designed for few-shot segmentation tasks, we perform some minor adjustments to their model structures to better suit the task. Specifically, on top of their existing textual input, we include support image features and support ground truth features as additional inputs for the language models. The support image features are obtained directly from the CLIP encoder, while the support ground truth features are acquired through SAM's [29] prompt encoder. After incorporating these changes, we retrain the altered models on few-shot segmentation datasets, allowing us to fairly evaluate their performance against our LLaFS++ framework. As shown in Table IV, we note that although LiSA and PixelLM also employ LLMs with a 7B parameter size, their performance on all three datasets is significantly worse than that of LLaFS++. This is because our LLaFS++ contains several task-tailored designs such as the novel instructions and pseudo-sample-based pre-training mechanisms, which enable the LLM to handle few-shot segmentation more effectively. The results demonstrate the excellent performance of LLaFS++ in comparison to other LLM-based segmentation methods. It also suggests that the superior performance of LLaFS++ is NOT attributable only to the use of an LLM, but is also a result of our carefully designed, innovative, and task-tailored methods that enhance the LLM's ability to process few-shot segmentation.

TABLE VI
EFFECTIVENESS OF EXTENSION COMPARED TO LLAFS

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

'RAA': region-attribute alignment method; GFLOPs represent the computations for the RAA process.

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

'IN': inference method; 'FH': the frequency of oversegmentation hallucinations occurring in all test images.

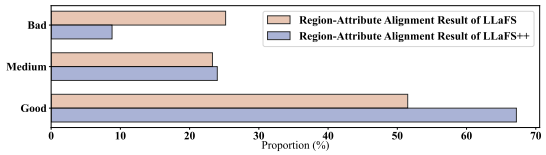


Fig. 6. Six volunteers' average scores regarding the quality of the region-attribute alignment results for 500 randomly sampled images.

TABLE VII
EFFECTIVENESS OF LARGE LANGUAGE MODEL.

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

TABLE VIII
EFFECTIVENESS OF SUPPORT IMAGES

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

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

D. Ablation Study

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

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

2) Effectiveness of Extension Compared to LLaFS: Our LLaFS++ proposed in this paper extends LLaFS [103] in two significant aspects: (1) a better method for aligning image regions with class attributes to build the region-attribute corresponding table, and (2) a contrastive prediction method for inference to mitigate hallucinations. As presented in Table VI, discarding these enhancements and reverting to the original methods used in LLaFS results in a significant decrease in performance, which validates the high effectiveness of our extended approaches. Also note that the region-attribute alignment method used in LLaFS++ (including the process of region encoding network REN) reduces computational load by over 85% compared to LLaFS, since it no longer requires extracting a CLIP feature for every cropped image. Furthermore, we conduct a more detailed evaluation. Specifically, we randomly select 500 images from the PASCAL-5^i test set, and use the methods in LLaFS and LLaFS++ to extract the alignment results between image regions (cropped image in LLaFS and similarity map M in LLaFS++) and class attributes. It is challenging to directly assess these results' quality since we do not have ground truth for such alignment. Thus, we instead conduct a user study by inviting six volunteers, who are completely unrelated to this research, to rate each test image's alignment result as 'bad', 'medium', or 'good'. The average results of the six raters are presented in Figure 6, which indicates the significant advantages of LLaFS++ proposed in this paper. Additionally, we also calculate the frequency of oversegmentation hallucination occurring in all the test images of the PASCAL-5^i dataset in both LLaFS and LLaFS++. Oversegmentation hallucination refers to the issue where the model incorrectly segments multiple local regions of the target object instead of capturing it as a whole (See Sec.III-F2 for details). As shown in Table VI, using the inference method in LLaFS++ reduces this frequency (FH) from 11.8% to 3.9%, demonstrating the effectiveness of our approach.

3) Effectiveness of Large Language Models: With extensive prior knowledge and powerful few-shot capabilities, the large language model (LLM) contributes significantly to the high effectiveness of our LLaFS++ framework. To validate the importance of the LLM within our model, we conduct an experiment by excluding the LLM from LLaFS++ and evaluate the performance of a modified model that is composed of the remaining parts of LLaFS++. More specifically, in constructing this model 'LLaFS++ w/o LLM', we perform some small alterations on model structures to ensure that few-shot segmentation could be executed solely with the remaining components (detailedly illustrated in Appendix). As shown in Table VII, removing LLM significantly decreases mIoU by 15.5% compared to the complete LLaFS++, demonstrating the crucial role of the LLM in ensuring the high performance of our framework. In our method, we employ CodeLlama instead of the vanilla Llama as the large language model. This is because CodeLlama finetuned with code generation datasets is more killed in generating structured information like the segmentation result in our task. This is demonstrated by the result presented in Table VII, which shows that the performance of using CodeLlama is 4.4% better than Llama.

4) Effectiveness of Support Images: Based on the task setting of few-shot segmentation, we leverage a small number of annotated images, called support images, to provide visual reference information for guiding the segmentation process. In fact, LLM-based segmentation models can also perform segmentation in an open-vocabulary manner, that is, to segment a category by solely utilizing its class name but without the need to apply any annotated support image. To evaluate the impact

TABLE IX
ABLATION STUDY OF FINE-GRAINED IN-CONTEXT INSTRUCTION

Method	mIoU	FB-IoU
LLaFS++	77.8	87.1
LLaFS++ w/o class attributes	74.2	84.8
LLaFS++ w/o region-attribute corresponding table	72.3	82.9
LLaFS++ w/o thresholding procedure in Eq.1	73.9	84.7
LLaFS++ w/o instruction refinement	74.5	85.0
LLaFS++ w/o iterative refinement	75.8	85.7

of support images on enhancing the model's effectiveness, we conduct experiments to compare with such an open-vocabulary and support-image-free method, with the results presented in Table VIII. When we keep the training schema unchanged yet removing the support image and its ground truth mask from the input during inference, there is an observed decrease in the model's mIoU by 14.3%. If these elements are also excluded from the training phase, this gap can be reduced to 12.2% since the training and inference inputs become aligned, but the result is still significantly worse than the original LLaFS++. These results demonstrate that our LLaFS++ benefits not solely from LLM's prior knowledge in an open-vocabulary manner but indeed gains further improvement from the provided few-shot samples. Moreover, we investigate a scenario within an in-vocabulary setting, where the categories in testing are the same as the categories in training. Concretely, we employ all 20 classes in PASCAL-5^i for training and also apply all these classes for testing. In this scenario, LLaFS++ still significantly outperforms the methods that do not leverage support images, indicating that incorporating a small number of annotated samples during evaluation can effectively enhance model performance, even for categories that have already been well trained with extensive training data. Such results demonstrate the crucial role of support images within our few-shot segmentation framework.

5) Ablation of Fine-grained In-context Instruction: The fine-grained in-context instruction constitutes a crucial component of our LLaFS++ framework, which combines visual information from support images with textual cues from the LLM's pretrained knowledge to form a comprehensive reference that can guide the segmentation of query images effectively. We conduct a thorough evaluation of various components and designs within this instruction and present the results in Table IX. As illustrated in Sec.III-C, the fine-grained in-context instruction is primarily made up of two parts: attributes of the target class and a region-attribute corresponding table derived from the support image. Table IX shows that excluding these components can respectively decrease the mIoU by 3.6% and 5.5%, demonstrating the importance of these reference information in guiding the segmentation of query images. We also evaluate the detailed designs within this instruction, including (1) the thresholding procedure (Eq.1) to exclude support-non-matching attributes, (2) the instruction refinement framework to resolve class ambiguities (Sec.III-C5), and (3) the iterative execution of this refinement. Table IX shows that removing any of these designs will cause a significant reduction in performance, thus demonstrating their important contributions to enhancing model effectiveness.

TABLE X
ABLATION STUDY OF PSEUDO-SAMPLE-BASED CURRICULUM PRETRAINING

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

'SF', 'QF', 'F', 'B' respectively refer to support foreground, query foreground, foreground, background.

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

7) Settings of Hyper-parameter α: As illustrated in Sec.III-C3, it is observed that not every attribute extracted for the target class can find a corresponding region in the support foreground. To prevent the introduction of misleading information due to this issue, we use a thresholding method to exclude the regional features calculated from attributes that do not correspond with the support. As illustrated in

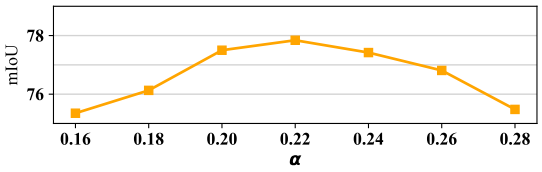
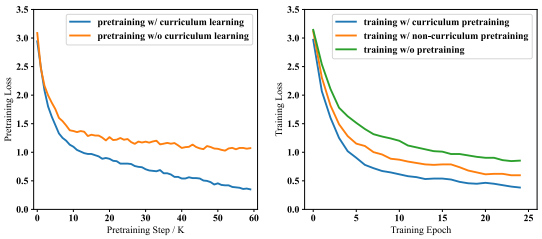


Fig. 7. Performance of using different values for the threshold α in Eq.1



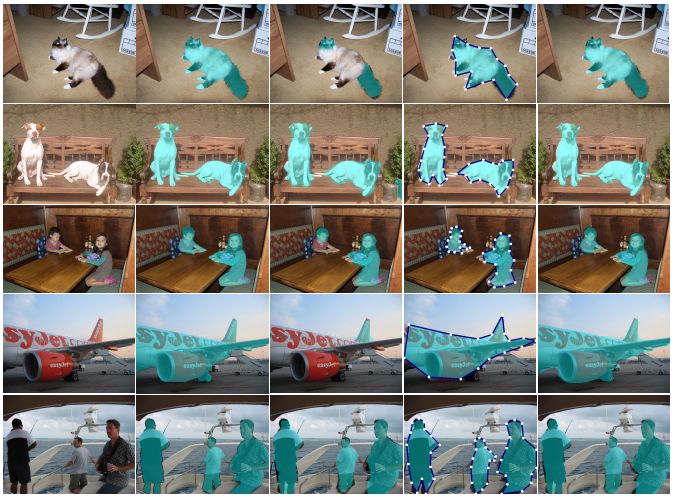
(a) Pretraining Loss Curve (b) Training Loss Curve

Fig. 8. Pretraining (a) and training (b) loss curves in different settings. Curriculum pretraining results in the best convergence in both pretraining and training stages. (Best viewed in color)

Eq.1, this process is made possible by a predefined threshold α. We experiment with different values for α to find the optimal choice and present the results in Figure 7. It is observed that both excessively small and large values for α can decrease the mIoU. This might be due to the fact that an excessively small value of α could lead to a false positive problem, where non-matching attributes may be erroneously classified as matching; while an excessively large value of α could lead to a false negative issue, where matching attributes are incorrectly deemed non-matching. Both conditions can adversely affect the quality of the generated region-attribute corresponding table. Based on the results shown in Figure 7, we choose $\alpha = 0.22$ as the threshold setting in our framework.

E. Loss Curves

In Sec.III-E, we introduce a curriculum-learning-based method to accelerate the optimization convergence of our model. To evaluate the effectiveness of this approach, we compare the loss curves for models pretrained with and without this curriculum learning strategy. The results presented in Figure 8(a) indicate that without the use of curriculum learning, the pretraining task becomes excessively challenging, which causes the model optimization to quickly reach a bottleneck with difficulties in the further convergence. After utilizing curriculum learning, this issue is significantly alleviated and the model can continuously converge. In Figure 8(b), we further present a comparison of the loss reduction conditions during the training phase after using different pretraining methods: pretraining with curriculum learning, pretraining without curriculum learning, and no pretraining at all. The model that has not undergone any pretraining is observed to have the lowest convergence rate, while the model pretrained with the curriculum learning strategy shows the swiftest convergence in the training phase, which demonstrates the effectiveness of our proposed curriculum-based pretraining method.



(a) Image (b) Ground Truth (c) Result from LLaFS (d) Result from LLaFS++'s LLM (e) Result from LLaFS++'s refinement network

Fig. 9. Visualization of segmentation results for LLaFS and LLaFS++.

F. Visualizations of Segmentation Results

To provide an intuitive demonstration of our method's high performance and to illustrate the progress we have made in this extended work, we present visualizations of segmentation results generated by LLaFS++ and compare them with those from LLaFS [103]. These visualization results are presented in Figure 9, with each row from left to right showcasing the query image, the query ground truth, the segmentation result from LLaFS, the segmentation result from LLaFS++'s LLM, and the segmentation result from LLaFS++'s refinement network, respectively. A frequent error observed in the original LLaFS is its tendency toward oversegmentation hallucination, which refers to the model's mistake to segment only partial regions rather than the entirety of the target object in the query image (illustrated in detail in Sec.III-F2). LLaFS++, the extended version in this paper with several improvements and new designs, shows stronger segmentation capabilities with the more accurate segmentation outputs compared to LLaFS. Furthermore, the issue of oversegmentation hallucination is also significantly mitigated benefiting from our newly proposed inference method. It is noteworthy that the polygons produced by the LLaFS++'s LLM already exhibit strong segmentation results, while the results output from the refinement network are further refined and more precise, particularly at the object edges. Another important observation is that in cases where an image consists of multiple objects, our method still demonstrates robust performance by accurately predicting multiple polygons to enclose different objects. These results demonstrate the excellent performance of our LLaFS++, showcasing its high effectiveness in handling the task of few-shot segmentation.

G. Extended Experiments

LLMs are known for their strong generalization abilities to effectively and robustly handle different conditions, such as different tasks, diverse input formats, and different domains,

TABLE XI
EXPERIMENTAL RESULTS ON GENERALIZED FEW-SHOT SEGMENTATION

Method	1-shot			5-shot		
	$\mathrm{mIoU_b}$	$\mathrm{mIoU_n}$	$\mathrm{mIoU_m}$	$\mathrm{mIoU_b}$	$\mathrm{mIoU_n}$	$\mathrm{mIoU_m}$
CAPL [69]	65.5	18.9	42.2	66.1	22.4	44.3
DIaM [16]	70.9	35.1	53.0	70.9	55.3	63.1
VP [19]	76.4	39.8	58.1	76.4	56.1	66.3
PixelLM [62]	78.6	53.5	66.1	79.1	62.6	70.9
LLaFS++	**79.8**	**62.7**	**71.3**	**80.2**	**71.4**	**75.8**

TABLE XII
RESULTS ON CROSS-DOMAIN FEW-SHOT SEGMENTATION

Method	1-shot	5-shot
Meta-Memory [77]	65.6	70.1
BAM-final [32]	69.0	71.7
IFA [56]	71.0	80.9
PGMA-Net [8]	72.4	72.8
LLaFS++	**79.6**	**84.3**

TABLE XIII
EXPERIMENTAL RESULTS ON WEAK LABELS

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

'PM': pixel-level mask; 'BB': bounding box.

by leveraging their extensive and diverse pre-trained knowledge. This observation motivates us to investigate whether LLaFS++, as an LLM-based segmentation model, also exhibits such strong generalization capabilities in visual tasks. To this end, we further conduct a series of extended experiments on generalized few-shot segmentation (Sec.IV-G1), cross-domain few-shot segmentation (Sec.IV-G2), weak-label few-shot segmentation (Sec.IV-G3), and few-shot object detection (Appendix.C7) to evaluate LLaFS++'s class generalization ability, domain generalization ability, input format generalization ability, and task generalization ability, respectively. To ensure a fair comparison with previous methods, we use the ResNet101 backbone for few-shot object detection and ResNet50 for other tasks.

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

2) Cross-Domain Few-Shot Segmentation: The recently proposed task of cross-domain few-shot segmentation focuses on addressing the challenging domain shift problem within few-shot segmentation. This task not only needs to segment new, previously unseen classes as the normal few-shot segmentation, but also requires the model to be able to process testing images in the different domains from the training images. Following previous works, we conduct experiments in the COCO-to-PASCAL setting, where the model is trained using the COCO-20^i dataset and tested on the PASCAL-5^i dataset. Results presented in Table XII indicate that LLaFS++ achieves the best performance. Note that we do not make any changes to the model structure or the training method of LLaFS++ in this experiment, yet it still outperforms all comparative methods, including both other few-shot segmentation methods like PGMA-Net [8] and those designed specifically for cross-domain few-shot segmentation [77], [56]. This could be attributed to the rich and general pretrained knowledge of LLM, which enables our framework to handle different domains effectively. These results indicate that LLaFS++ can achieve excellent performance even in the presence of domain shifts, thus demonstrating its high robustness and effectiveness.

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

V. CONCLUSION

This paper introduces LLaFS++, an extension of LLaFS, as a novel and effective framework that leverages large language models (LLMs) to address few-shot segmentation. To adapt

LLMs for this visual task, we introduce a segmentation task instruction to provide detailed task definitions, a fine-grained in-context instruction to simulate human cognitive mechanisms and offer fine-grained multimodal reference information, and a pseudo-sample-based curriculum pretraining mechanism to augment the training samples required for instruction tuning. Furthermore, we improve upon LLaFS by introducing a better method for constructing in-context instructions with lower computational cost and higher precision, as well as a novel inference strategy to mitigate potential over-segmentation hallucinations caused by regional guidance information. With these new techniques and improvements, LLaFS++ achieves outstanding performance, as demonstrated by extensive experiments across various datasets and scenarios, where LLaFS++ not only achieves state-of-the-art results compared to previous advanced methods, but also shows substantial improvements over the original LLaFS (+2.1% on PASCAL-5^i and +3.6% on COCO-20^i). We consider LLaFS++ a significant step forward in leveraging LLMs to tackle few-shot challenges in computer vision. Despite its significant success, our work still has a limitation: LLaFS++ is designed only for image segmentation and cannot directly handle video segmentation, which is also crucial for many real-world applications. In the future, we plan to extend LLaFS++ to support a broader range of input formats, including both images and videos. Additionally, we will explore the potential of leveraging advanced LLM techniques, such as chain-of-thought reasoning and reinforcement learning, to further enhance segmentation performance.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[3] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13979–13988, 2021.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Proc. Adv. Neural Inf. Process. Syst.*, 33:1877–1901, 2020.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[8] Shuai Chen, Fanman Meng, Runtong Zhang, Heqian Qiu, Hongliang Li, Qingbo Wu, and Linfeng Xu. Visual and textual prior guided mask assemble for few-shot segmentation and beyond. *IEEE Trans. Multimedia*, 26:7197–7209, 2024.

[9] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. In *Int. Conf. Comput. Vis.*, pages 909–919, 2023.

[10] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12475–12485, 2020.

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1290–1299, 2022.

[12] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Proc. Adv. Neural Inf. Process. Syst.*, 34:17864–17875, 2021.

[13] Gong Cheng, Chunbo Lang, and Junwei Han. Holistic prototype activation for few-shot segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4650–4666, 2022.

[14] Qi Fan, Wenjie Pei, Yu-Wing Tai, and Chi-Keung Tang. Self-support few-shot semantic segmentation. In *Eur. Conf. Comput. Vis.*, pages 701–719. Springer, 2022.

[15] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Proc. Adv. Neural Inf. Process. Syst.*, 35:30583–30598, 2022.

[16] Sina Hajimiri, Malik Boudiaf, Ismail Ben Ayed, and Jose Dolz. A strong baseline for generalized few-shot semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11269–11278, 2023.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Int. Conf. Comput. Vis.*, pages 2961–2969, 2017.

[18] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *Eur. Conf. Comput. Vis.*, pages 108–126. Springer, 2022.

[19] Mir Rayat Imtiaz Hossain, Mennatullah Siam, Leonid Sigal, and James J Little. Visual prompting for generalized few-shot segmentation: A multi-scale approach. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[20] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Int. Conf. Mach. Learn.*, pages 2790–2799. PMLR, 2019.

[21] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Int. Conf. Learn. Represent.*, 2021.

[22] Tao Hu, Pengwan Yang, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees GM Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. In *Proc. AAAI Conf. Artif. Intell.*, volume 33, pages 8441–8448, 2019.

[23] Kai Huang, Feigege Wang, Ye Xi, and Yutao Gao. Prototypical kernel learning and open-set foreground perception for generalized few-shot semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 19256–19265, 2023.

[24] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2989–2998, 2023.

[25] Siyu Jiao, Yunchao Wei, Yaowei Wang, Yao Zhao, and Humphrey Shi. Learning mask-aware clip representations for zero-shot segmentation. *Proc. Adv. Neural Inf. Process. Syst.*, 36:35631–35653, 2023.

[26] Siyu Jiao, Gengwei Zhang, Shant Navasardyan, Ling Chen, Yao Zhao, Yunchao Wei, and Humphrey Shi. Mask matching transformer for few-shot segmentation. In *Proc. Adv. Neural Inf. Process. Syst.*

[27] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Occlusion-aware instance segmentation via bilayer network architectures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[28] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9404–9413, 2019.

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Int. Conf. Comput. Vis.*, pages 4015–4026, 2023.

[30] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[31] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8057–8067, 2022.

[32] Chunbo Lang, Gong Cheng, Binfei Tu, Chao Li, and Junwei Han. Base and meta: A new perspective on few-shot segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[33] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13906–13915, 2020.

[34] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8334–8343, 2021.

[35] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Int. Conf. Mach. Learn.*, 2023.

[36] Liulei Li, Wenguan Wang, Tianfei Zhou, Ruijie Quan, and Yi Yang. Semantic hierarchy-aware segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[37] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2869–2878, 2020.

[38] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*, 2022.

[39] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1280–1289, 2022.

[40] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7061–7070, 2023.

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proc. Adv. Neural Inf. Process. Syst.*, 2023.

[42] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. Polyformer: Referring image segmentation as sequential polygon generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18653–18663, 2023.

[43] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4165–4173, 2020.

[44] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proc. Annu. Meeting. Assoc. Comput. Linguist.*, pages 61–68, 2022.

[45] Yuanwei Liu, Nian Liu, Qinglong Cao, Xiwen Yao, Junwei Han, and Ling Shao. Learning non-target knowledge for few-shot semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11573–11582, 2022.

[46] Yuanwei Liu, Nian Liu, Xiwen Yao, and Junwei Han. Intermediate prototype mining transformer for few-shot semantic segmentation. *Proc. Adv. Neural Inf. Process. Syst.*, 35:38020–38031, 2022.

[47] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *Eur. Conf. Comput. Vis.*, pages 142–158. Springer, 2020.

[48] Xiaoliu Luo, Zhuotao Tian, Taiping Zhang, Bei Yu, Yuan Yan Tang, and Jiaya Jia. Pfenet++: Boosting few-shot semantic segmentation with the noise-filtered context-aware prior mask. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[49] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Int. Conf. Comput. Vis.*, pages 6941–6952, 2021.

[50] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, 2022.

[51] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *Int. J. Comput. Vis.*, 129(5):1551–1579, 2021.

[52] Seonghyeon Moon, Samuel S Sohn, Honglu Zhou, Sejong Yoon, Vladimir Pavlovic, Muhammad Haris Khan, and Mubbasir Kapadia. Msi: Maximize support-set information for few-shot segmentation. In *Int. Conf. Comput. Vis.*, pages 19266–19276, 2023.

[53] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 19413–19423, 2023.

[54] Gregory L Murphy and Douglas L Medin. The role of theories in conceptual coherence. *Psychological review*, 92(3):289, 1985.

[55] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Int. Conf. Comput. Vis.*, pages 622–631, 2019.

[56] Jiahao Nie, Yun Xing, Gongjie Zhang, Pei Yan, Aoran Xiao, Yap-Peng Tan, Alex C Kot, and Shijian Lu. Cross-domain few-shot segmentation via iterative support-query correspondence mining, 2024.

[57] Atsuro Okazawa. Interclass prototype relation for few-shot segmentation. In *Eur. Conf. Comput. Vis.*, pages 362–378. Springer, 2022.

[58] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[59] Bohao Peng, Zhuotao Tian, Xiaoyang Wu, Chengyao Wang, Shu Liu, Jingyong Su, and Jiaya Jia. Hierarchical dense correlation distillation for few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 23641–23651, 2023.

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pages 8748–8763. PmLR, 2021.

[61] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*, 2023.

[62] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[63] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[64] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.

[65] Hengcan Shi, Son Duy Dao, and Jianfei Cai. Llmformer: Large language model for open-vocabulary semantic segmentation. *Int. J. Comput. Vis.*, pages 1–18, 2024.

[66] Xinyu Shi, Dong Wei, Yu Zhang, Donghuan Lu, Munan Ning, Jiashun Chen, Kai Ma, and Yefeng Zheng. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *Eur. Conf. Comput. Vis.*, pages 151–168. Springer, 2022.

[67] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023.

[68] Yanpeng Sun, Jiahui Chen, Shan Zhang, Xinyu Zhang, Qiang Chen, Gang Zhang, Errui Ding, Jingdong Wang, and Zechao Li. Vrp-sam: Sam with visual reference prompt. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 23565–23574, 2024.

[69] Zhuotao Tian, Xin Lai, Li Jiang, Shu Liu, Michelle Shu, Hengshuang Zhao, and Jiaya Jia. Generalized few-shot semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11563–11572, 2022.

[70] Zhi Tian, Bowen Zhang, Hao Chen, and Chunhua Shen. Instance and panoptic segmentation using conditional convolutions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):669–680, 2022.

[71] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(2):1050–1065, 2020.

[72] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[73] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *Eur. Conf. Comput. Vis.*, pages 730–746. Springer, 2020.

[74] Jin Wang, Bingfeng Zhang, Jian Pang, Honglong Chen, and Weifeng Liu. Rethinking prior information generation with clip for few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[75] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Int. Conf. Comput. Vis.*, pages 9197–9206, 2019.

[76] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023.

[77] Wenjian Wang, Lijuan Duan, Yuxi Wang, Qing En, Junsong Fan, and Zhaoxiang Zhang. Remember the difference: Cross-domain few-shot semantic segmentation via meta-memory transfer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7065–7074, 2022.

[78] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):4555–4576, 2021.

[79] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Solo: A simple framework for instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):8587–8601, 2021.

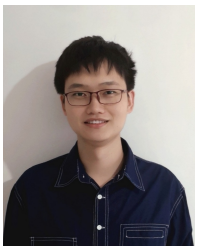
[80] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Towards segmenting everything in context. In *Int. Conf. Comput. Vis.*, pages 1130–1140, 2023.

[81] Yan Wang, Jian Cheng, Yixin Chen, Shuai Shao, Lanyun Zhu, Zhenzhou Wu, Tao Liu, and Haogang Zhu. Fvp: Fourier visual prompting

for source-free unsupervised domain adaptation of medical image segmentation. *arXiv preprint arXiv:2304.13672*, 2023.
[82] Yuan Wang, Naisong Luo, and Tianzhu Zhang. Focus on query: Adversarial mining transformer for few-shot segmentation. *Proc. Adv. Neural Inf. Process. Syst.*, 36:31524–31542, 2023.
[83] Yuan Wang, Rui Sun, and Tianzhu Zhang. Rethinking the correlation in few-shot segmentation: A buoys view. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7183–7192, 2023.
[84] Yuan Wang, Rui Sun, Zhe Zhang, and Tianzhu Zhang. Adaptive agent transformer for few-shot segmentation. In *Eur. Conf. Comput. Vis.*, pages 36–52. Springer, 2022.
[85] Edward J Wisniewski and Bradley C Love. Relations versus properties in conceptual combination. *Journal of memory and language*, 38(2):177–202, 1998.
[86] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12193–12202, 2020.
[87] Enze Xie, Wenhai Wang, Mingyu Ding, Ruimao Zhang, and Ping Luo. Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5385–5400, 2021.
[88] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Proc. Adv. Neural Inf. Process. Syst.*, 34:12077–12090, 2021.
[89] Guo-Sen Xie, Jie Liu, Huan Xiong, and Ling Shao. Scale-aware graph neural network for few-shot semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5475–5484, 2021.
[90] Zhitong Xiong, Haopeng Li, and Xiao Xiang Zhu. Doubly deformable aggregation of covariance matrices for few-shot segmentation. In *Eur. Conf. Comput. Vis.*, pages 133–150. Springer, 2022.
[91] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2945–2954, 2023.
[92] Qianxiong Xu, Wenting Zhao, Guosheng Lin, and Cheng Long. Self-calibrated cross attention network for few-shot segmentation. In *Int. Conf. Comput. Vis.*, 2023.
[93] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. Mining latent classes for few-shot segmentation. In *Int. Conf. Comput. Vis.*, pages 8721–8730, 2021.
[94] Yong Yang, Qiong Chen, Yuan Feng, and Tianlin Huang. Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7131–7140, 2023.
[95] Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively. In *Eur. Conf. Comput. Vis.*, pages 419–437. Springer, 2024.
[96] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. *arXiv preprint arXiv:2312.10032*, 2023.
[97] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 9587–9595, 2019.
[98] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Proc. Adv. Neural Inf. Process. Syst.*, 34:21984–21996, 2021.
[99] Gengwei Zhang, Shant Navasardyan, Ling Chen, Yao Zhao, Yunchao Wei, Honghui Shi, et al. Mask matching transformer for few-shot segmentation. *Proc. Adv. Neural Inf. Process. Syst.*, 35:823–836, 2022.
[100] Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. Psalm: Pixelwise segmentation with large multi-modal model. *arXiv preprint arXiv:2403.14598*, 2024.
[101] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11175–11185, 2023.
[102] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.
[103] Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. Llafs: When large language models meet few-shot segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.
[104] Lanyun Zhu, Tianrun Chen, Jianxiong Yin, Simon See, and Jun Liu. Continual semantic segmentation with automatic memory sample selection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3082–3092, 2023.
[105] Lanyun Zhu, Tianrun Chen, Jianxiong Yin, Simon See, and Jun Liu. Addressing background context bias in few-shot segmentation through iterative modulation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.
[106] Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12537–12546, 2021.

Lanyun Zhu received his Ph.D degree from the Information Systems Technology and Design (ISTD) pillar, Singapore University of Technology and Design in 2025, and B.E. degree from Beihang University, Beijing, China in 2020. His research interests are mainly focused on deep learning and computer vision. He is the reviewer of multiple top journals and conferences including TPAMI, TIP, TMM, IJCV, CVPR, ECCV, ICML, NeurIPS and ICLR.

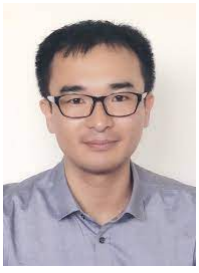
Tianrun Chen received the bachelor's degree in College of Information Science and Electronic Engineering, Zhejiang University and is pursuing the Ph.D degree with the College of Computer Science and Technology, Zhejiang University. He is the founder and the technical director of Moxin (Huzhou) Technology Co., LTD. His research interest includes computer vision and its enabling applications.



Deyi Ji is a Senior Researcher at Alibaba Group, where he works with Prof. Xian-Sheng Hua and Prof. Jieping Ye. Before that, he worked as a Researcher at SenseTime with Prof. Xiaogang Wang and Dr. Wei Wu. He received the M.S. degree of Computer Science from Shanghai Jiao Tong University (SJTU) in 2019, advised by Prof. Hongtao Lu. He received B.S. degree from Huazhong University of Science and Technology (HUST) in 2016, advised by Prof. Xinggang Wang.

Peng Xu received his M.S. degree in Computer Science from Peking University, China, in 2018. He currently serves as a Senior Algorithm Engineer at the Apsara Lab of Alibaba Cloud Intelligence in Beijing, China. His research interests encompass deep learning, computer vision, multimodal large language models, and table-based question answering within large language models.

Jieping Ye (Fellow, IEEE) received the Ph.D. degree in computer science and engineering from the University of Minnesota, in 2005. He is currently the Head of Apsara Lab in Alibaba Group. He is also a Professor of the University of Michigan, Ann Arbor, MI, USA. He has served as a Senior Program Committee/Area Chair/Program Committee Vice Chair of many conferences, including NeurIPS, ICML, KDD, IJCAI, ICDM, and SDM. He has served as an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.



Jun Liu is a Professor and Chair in Digital Health at School of Computing and Communications in Lancaster University. He got the PhD degree from Nanyang Technological University in 2019. He is an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR AND IDENTITY SCIENCE, *ACM Computing Surveys*, and *Pattern Recognition*. He has served as an Area Chair of CVPR, ECCV, ICML, NeurIPS, ICLR and MM.