LLaFS: When Large Language Models Meet Few-Shot Segmentation

Lanyun Zhu[1] Tianrun Chen[2] Deyi Ji[3] Jieping Ye[3] Jun Liu[1*]
Singapore University of Technology and Design [1] Zhejiang University [2] Alibaba Group [3]

Abstract

This paper proposes LLaFS, the first attempt to leverage large language models (LLMs) in few-shot segmentation. In contrast to the conventional few-shot segmentation methods that only rely on the limited and biased information from the annotated support images, LLaFS leverages the vast prior knowledge gained by LLM as an effective supplement and directly uses the LLM to segment images in a few-shot manner. To enable the text-based LLM to handle image-related tasks, we carefully design an input instruction that allows the LLM to produce segmentation results represented as polygons, and propose a region-attribute table to simulate the human visual mechanism and provide multi-modal guidance. We also synthesize pseudo samples and use curriculum learning for pretraining to augment data and achieve better optimization. LLaFS achieves state-of-the-art results on multiple datasets, showing the potential of using LLMs for few-shot computer vision tasks.

1. Introduction

Image segmentation is a fundamental task in computer vision with extensive applications. The development of deep learning algorithms [7, 9, 12, 72, 73] trained on large-scale datasets has brought significant advancements to this field [3–5, 61, 69, 71]. However, annotating pixel-level segmentation ground truth on a large scale is extremely resource-intensive. Therefore, a more source-efficient learning strategy, few-shot segmentation, has received much attention from academia and holds immense practical value.

In few-shot segmentation, the model should develop category-specific segmentation capabilities based on only a small amount of annotated data, called support images. To achieve this, existing few-shot segmentation methods [24, 27, 38, 42, 43, 49, 66, 67] typically adopt a support-feature-guided framework. In this framework, relevant features of the target category are extracted from annotated support images and used as guiding information to segment query images. To achieve higher performance, researchers have proposed many methods to explore better ways for support feature extraction [24, 38, 43] and query segmentation assistance [17, 56, 62]. Although these efforts have demonstrated some success, their segmentation performance is still far from satisfactory. This is because the very limited number of support images contain only a small, incomplete, and biased set of information, so the framework that relies solely on these support-based features for query segmentation inherently suffers from information constraints and cannot achieve a sufficiently high level of accuracy. Therefore, we believe that the further advancement of few-shot segmentation urgently requires an entirely new framework, which should be capable of utilizing richer and more comprehensive information, thereby breaking through the existing framework's bottlenecks to reach better results.

We discover that recent advances in large language models (LLMs) [2, 50, 70] can offer potential opportunities to achieve this goal. Specifically, LLMs pre-trained on large-scale corpora have accumulated a vast amount of prior knowledge, which can effectively supplement the insufficient information in support images, thereby resulting in the more effective guidance. Moreover, LLMs have shown to be effective few-shot learners in the field of NLP [2]. This naturally inspires us to further extend their capabilities to few-shot tasks in other modalities. Based on these insights, we hereby innovatively employ LLMs to tackle few-shot segmentation and introduce an entirely new framework named LLaFS. Unlike some previous segmentation methods that also use language models (LMs) but only for auxiliary purposes, such as utilizing LMs to extract intermediate features [13, 28, 64] or to generate attribute prompts [37], our LLaFS directly employs LLMs to produce segmentation results. This makes LMs no longer work as only auxiliary tools, but fully unlock their complete potential in handling the complex computer vision tasks in an end-to-end manner. In this way, we provide an important exploration towards a unified framework that allows LLMs to tackle few-shot tasks in other modalities beyond NLP.

We find that integrating LLM to few-shot segmentation is non-trivial as we face three critical technical challenges: 1) *How to enable the text-based LLM to comprehend and address an image processing task?* 2) *How to leverage both the visual information from support images and the*

*Corresponding Author

text information from the LLM to guide the query segmentation? and 3) *How to effectively train the model with only limited data?* To address the first challenge, we draw inspiration from instruction tuning [44] and introduce a task instruction, which is used to explicitly define the few-shot segmentation task within the input of the LLM. To tackle the second challenge, we treat support images as in-context demonstration samples and design a region-attribute corresponding table to extract fine-refined multi-modal guidance information. For the third challenge, we further propose a pseudo-sample synthesis method to augment the pretraining samples and introduce a curriculum learning mechanism to achieve better optimization. By incorporating these designs, our LLaFS can handle few-shot segmentation effectively. We conduct experiments on multiple datasets and achieve state-of-the-art (SOTA) results that significantly outperform existing methods.

In summary, the contributions of this work are as follows: 1) We introduce LLaFS, the first framework to address few-shot segmentation using large language models. 2) 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 multimodal guidance, and a pseudo-sample-based curriculum pretraining mechanism. 3) Our approach achieves state-of-the-art performance on multiple datasets.

2. Related Work

Few-Shot Segmentation. To address the issue of conventional semantic segmentation methods [3, 6, 10, 18–20, 55, 61, 65, 74] that require a large number of training samples, the task of few-shot segmentation (FSS), which allows to segment a query image using only a small number of annotated support images, has been proposed and gained significant attention [1, 21, 23, 31, 35, 43, 47, 52, 63]. Current FSS methods typically adopt a prototype-guided approach [8, 17, 23, 24, 42, 63]. They use masked average pooling (MAP) to extract global [8, 34, 42] or local [24] average prototypes from the backbone features of support images, and then employ these prototypes for guiding the segmentation of query images through feature fusion [23, 24, 34], distance computation [14, 38], or attention mechanisms [45]. However, these methods can only leverage a limited amount of information extracted from a very small number of support images, thus potentially leading to suboptimal results and reduced robustness. To overcome this limitation, [64] uses the more comprehensive word embedding as the general class information to assist in segmentation. Despite some improvements, [64] is still constrained by the limited capabilities of small language models and has not delved deeper into how to better integrate textual information and support image data to achieve more effective guidance. In this paper, we are the first to employ large language models (LLMs) to achieve FSS by using our carefully designed instructions that contain a more effective multimodal guidance. Furthermore, we utilize the LLM to directly produce segmentation results, rather than merely using intermediate features as done in [64]. This offers a brand-new paradigm to FSS.

Large Language Models. The advent of large language models (LLMs) such as GPT [2] and Llama [50] 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 [32, 58], adapters [15, 22] and LoRA [16]. Recently, researchers have started exploring visual large language models [25, 29, 48, 53] to establish a unified framework for multimodal data processing, aiming to override the restriction of LLMs being solely applicable to language data. However, none of them are designed for few-shot tasks in computer vision. In this paper, we introduce the first visual LLM framework for handling few-shot segmentation. To achieve this, we draw inspiration from instruction tuning [44] and in-context learning [11, 39], and carefully design a suitable form of instruction and demonstration examples tailored for few-shot segmentation. By doing so, our method can enable the LLM to comprehend image data and perform few-shot segmentation effectively.

3. Method

3.1. 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 K support images $\{I_s^k\}_{k=1}^K$ and their ground truth maps $\{G_s^k\}_{k=1}^K$.[1] As shown in Fig.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, followed by a refinement network to optimize the results. For the feature extractor, we adopt the approach in Blip2 [25] by using an image encoder followed by a Q-former and a fully-connected layer to generate a set of visual tokens. We use ResNet50 as the image encoder and keep it frozen during training. For the instruction, we carefully design it as the combination of two parts: segmentation task instruction (Sec.3.2.1) and fine-grained in-context instruction (Sec.3.2.2) to provide comprehensive and detailed guidance. For the LLM, we employ CodeLlama [46] with 7 billion parameters that have been fine-tuned through instruction tuning. Note that compared to

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

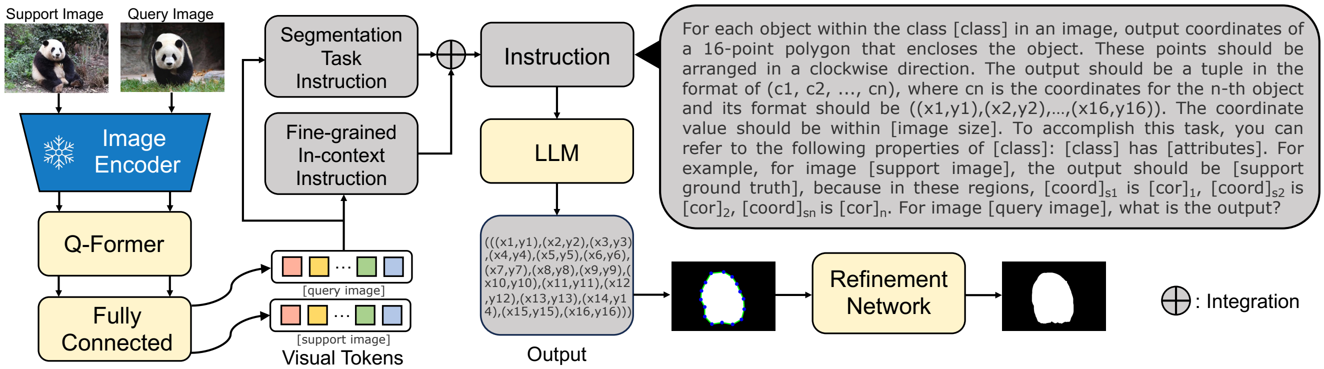


Figure 1. **Overview of LLaFS.** The image encoder and Q-former extract image features and generate a set of visual tokens. Subsequently, a segmentation task instruction and fine-grained in-context introduction are introduced to provide detailed and comprehensive information. These two instructions are integrated and fed into the LLM 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.

vanilla Llama, we empirically find that CodeLlama fine-tuned 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 fine-tuning. All these components work together within the LLaFS framework to achieve high-quality few-shot segmentation.

3.2. Instruction

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 Fig.1. In the following Sec.3.2.1 and Sec.3.2.2, we introduce these two instructions in detail.

3.2.1 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 these knowledge and capability to handle complex tasks [44]. To achieve better results, the instructions need to be sufficiently clear and detailed, whereas those using only simple terminologies such as 'segment an image' are evidently 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 [53] by representing the pixel-wise segmentation output as a 16-point polygon that encloses the target object [30]. Note that it is hard for LLMs to directly generate pixel-wise image 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, training solely on text contents makes LLMs difficult to comprehend visual information precisely, especially in our few-shot image segmentation task, where the number of available training images is very scarce. For this, inspired by the success of in-context learning in NLP [11, 39], we further propose to encode the support image and its ground truth as a visual demonstration example, using it as an intuitive reference in the instruction to teach LLM how to segment a particular 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-point polygon that encloses the object. These points should be arranged in a clockwise direction. The output should be a tuple in the format of (c1, c2, ..., cn), where cn is the coordinates for the n-th object and its format should be ((x1,y1),(x2,y2),...,(x16,y16)). The coordinate value should be within [image size]. For example, for image [support image], the output should be [support ground truth]. Here, [support image] is the visual tokens from the support image.

3.2.2 Fine-grained In-context Instruction

Motivation. The above task instruction makes segmenting a class possible by leveraging LLM's knowledge of the class. In the instruction, the class to be segmented is indicated by the [class] token, which is typically a single noun. However, considering that LLMs have never been trained on images, it is challenging for them to directly align this abstract noun with an image region that may possess a complex internal structure. To address this issue, we drew inspiration from human brains and found that when classifying an unseen new class, the human cognitive system follows a mechanism of '*from general to detailed, from abstract to concrete*' [41, 59]. Specifically, given an unseen class represented by a *general* noun, the human brain first decomposes

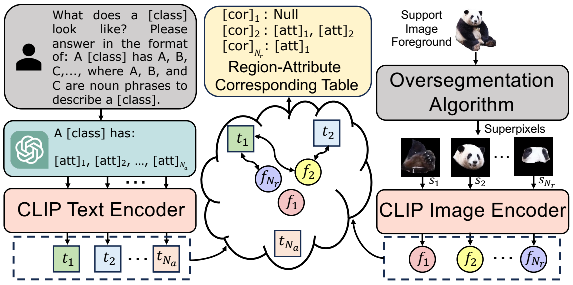


Figure 2. Illustration of how to construct the region-attribute corresponding table used in the fine-grained in-context instruction.

it into *detailed* attributes based on the acquired knowledge. For example, in the case of an unseen class 'owl', a person can first gather information from references to learn about the owl's attributes such as 'large round eyes' and 'hooked beak'. Subsequently, it can search the image for *concrete* regions that match these *abstract* attributes to determine the presence of the class.

Inspired by this, we propose a fine-grained in-context instruction to simulate such a human cognitive mechanism based on the support images. For this, we first use ChatGPT to extract detailed attributes of the target class, then we search for regions in support images that correspond to these attributes and generate a corresponding table accordingly. The obtained attributes and table constitute an in-context instruction that is fed into the LLM, which serves as a demonstration example to guide the LLM on how to recognize an image class in a more human-like and fine-grained manner. This alleviates the challenge that LLM cannot perform segmentation well when only inputted with an abstract class noun. Lastly, we introduce an expert-guide framework that refines the instruction to increase its class representation ability. The following sections explain how to generate and refine this instruction in detail.

Attributes Generation We first simulate the step of '*from general to detailed*' to extract class attributes. Specifically, as shown in Fig.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 utilize ChatGPT 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 $\{[\text{att}]_i\}_{i=1}^{N_a}$.

Region-attribute Corresponding Table. After that, we simulate the second step of '*from abstract to concrete*'. To implement this, as shown in Fig.2, we propose a region-attribute corresponding table to find the alignment between support image regions and class attributes. For this, we first

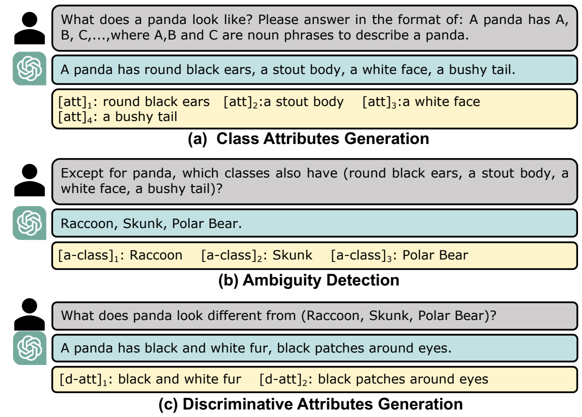


Figure 3. Examples of using ChatGPT for (a) class attributes generation, (b) ambiguity detection and (c) discriminative attributes generation.

divide the support foreground into multiple local regions. Specifically, for each object in the support image within the target class, we employ the method in selective search [51] to generate a set of superpixels $\{s_i\}_{i=1}^{N_r}$ with different scales in an unsupervised manner. Each s_i aggregates pixels that are close in position and similar in features, so it can represent a local region with a specific semantic meaning. Based on each s_i, a masked image is generated and passed through CLIP's[2] image encoder to produce a feature f_i. We calculate the cosine similarity between f_i and the embedding t_j for each attribute $[\text{att}]_j$, and utilize a thresholding process to establish region-attribute correspondence. Formally,

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

where cos refers to cosine similarity, α is a pre-defined threshold. The obtained $[\text{cor}]_i$ contains attributes that align with s_i. In this way, we get $\{[\text{cor}]_i\}_{i=1}^{N_r}$ from $\{s_i\}_{i=1}^{N_r}$, which serves as an attribute-region corresponding table that can provide the fine-grained multi-modal reference.

Instruction Construction. We integrate the obtained class attributes $\{[\text{att}]_i\}_{i=1}^{N_a}$ and corresponding table $\{[\text{cor}]_i\}_{i=1}^{N_r}$, and write the fine-grained in-context instruction as: The [class] has [attributes]. For example, in [support image], $[\text{coord}]_{s_1}$ is $[\text{cor}]_1$, $[\text{coord}]_{s_2}$ is $[\text{cor}]_2$..., $[\text{coord}]_{s_{N_r}}$ is $[\text{cor}]_{N_r}$. Here, $[\text{coord}]_{s_i}$ is the coordinates of s_i in the format of a 16-point polygon. By using this instruction, we provide the LLM with a reference about the attributes of the target class and their corresponding regions in the support image. In this way, a demonstration example can be created that simulates how the human cognitive mechanism recognizes the support image as a class. Using this example as a reference, the LLM can be taught how to understand

[2]We use CLIP fintuned through the method in [26] to ensure high-quality region-text alignment.

and segment an image class in a fine-grained manner.

Expert-guide Framework for Instruction Refinement. The above-mentioned instruction constructed by the obtained attributes $\{[\text{att}]_i\}_{i=1}^{N_a}$ and table $\{[\text{cor}]_i\}_{i=1}^{N_r}$ can be directly input into LLM for guidance. However, due to variations in camera angles and instances of occlusion, not every attribute can be directly matched to a region within the support image. Thus, the obtained table $\{[\text{cor}]_i\}_{i=1}^{N_r}$ may contain only a subset of attributes within $\{[\text{att}]_i\}_{i=1}^{N_a}$. Unfortunately, we find the combinations of these partial attributes may be insufficient for the unambiguous recognition of the target class. For example, the combination 'wheels, windows, doors' can refer to 'train', 'car', and 'bus' interchangeably. Due to the ambiguous table, the instruction may be misleading. To alleviate this issue, we propose an expert-guide framework to refine the instruction. In this framework, we first employ ChatGPT to identify ambiguous classes of the existing table, then we extract additional attributes that can distinguish the target class from these ambiguous classes for refinement. As long as these additional attributes can be aligned with local regions in the support image, the refined table based on them will become unambiguous. In this way, the class representation ability and comprehensiveness of the instruction can be improved.

Specifically, this framework generates a refined instruction through the following three steps: ***1) Ambiguity Detection.*** As shown in Fig.3(b), We employ ChatGPT to identify potential ambiguous classes in the obtained table $\{[\text{cor}]_i\}_{i=1}^{N_r}$. Specifically, we denote the attributes contained in $\{[\text{cor}]_i\}_{i=1}^{N_r}$ as [partial-attributes] and ask ChatGPT 'Except for [class], which classes also have [partial-attributes]?'[3] In this way, we obtain a set of ambiguous classes denoted as [a-classes]=$\{[\text{a-class}]_i\}_{i=1}^{N_{ac}}$ from ChatGPT's feedback. ***2) Discriminative Attributes Generation.*** As shown in Fig.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 ChatGPT to generate attributes that are more discriminative from these classes. The obtained attributes $\{[\text{d-att}]_i\}_{i=1}^{N_d}$ are added to [attributes] for updating. ***3) Table and Instruction Refinement.*** Finally, we use these updated attributes to reperform Eq.1 to obtain a refined table. The updated attributes and table are reassembled to form a refined fine-grained in-context instruction.

We found that a single execution of the three steps already resolves ambiguities in over 85% of the instructions. For the remaining 15% of instructions, we observe that because the newly-acquired discriminative attributes still couldn't find matching regions in the support image, the resulting table after refinement remains to be ambiguous. Therefore, we iteratively apply the last two steps of this framework until the ambiguity is eradicated. To achieve this, from the second iteration onwards, we replace the text prompt in the discriminative attributes generation step with 'Apart from [all-discriminative-attributes], tell me more differences in appearance between [class] and [a-classes]', where [all-discriminative-attributes] refers to the discriminative attributes $[\text{d-att}]_i$ obtained from all previous iterations. By doing so, we enable the iterative framework to continuously discover more discriminative attributes and verify whether they have matched regions in the support image. Eventually, when the framework successfully discovers discriminative attributes $[\text{d-att}]_i$ that can be aligned with the support image or reaches the maximum number of iterations, we terminate the iteration. For efficiency, we set the maximum number of iterations to 3, in which 96% of the ambiguities have been entirely eradicated.

3.3. Segmentation Prediction

We integrate segmentation task instruction and fine-grained in-context instruction to formulate the complete instruction as shown in Fig.1. With this instruction as input, the LLM can predict the coordinates of a 16-point polygon that surrounds the target object. Finally, to rectify the imprecision caused by the polygon representation of object edges, a refinement network comprising a pixel decoder and a mask transformer is introduced to generate a refined segmentation mask by using the polygon as the initial mask. Please see Supp for the detailed structures of this network.

3.4. Curriculum Pretraining with Pseudo Samples

Motivation. After carefully designing the model structure and instruction format, the next challenge is how to train LLaFS effectively to achieve high segmentation performance. Previous work [29] has shown that the success of instruction tuning often relies on extensive training data. However, due to the difficulty of obtaining pixel-annotated labels, the segmentation datasets used for training typically contain only an insufficient number of images. To address this issue, we propose to generate pseudo support-query pairs and use them to pretrain the LLM. The LLM's ability to handle few-shot segmentation can thus be enhanced by seeing more visual samples.

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 randomly generate a contour within an image region. 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 to

[3] Due to space limitations, we omit the description of format control prompts for inputting into ChatGPT. See Supp for details.

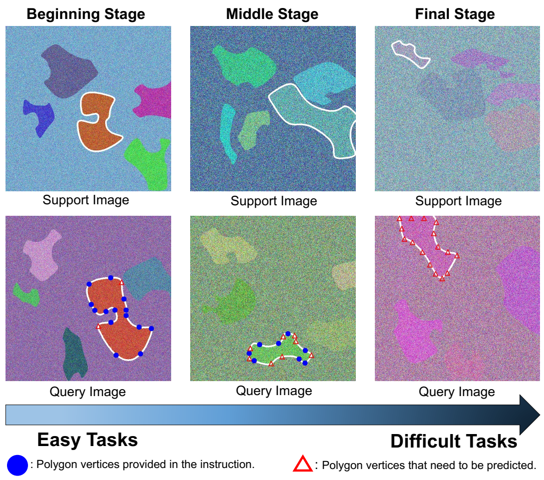


Figure 4. Examples of pseudo samples generated at different pretraining stages. Foreground regions are marked by white contours. As pretraining progresses, pseudo images have reduced intra-image foreground-background differences and greater support-query foreground differences. Meanwhile, the number 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)

simulate the complex backgrounds in real images, then we fill each one with a random Gaussian noise that has a mean value distinct from the foreground's noise. The obtained image is used as the support image. ***3) Pseudo query generation.*** We use a similar approach to generate a query image. Note that in this process, the contour and the mean value of foreground noise are no longer completely randomly determined but adjusted based on those used for generating the support image. This is done to ensure that the foreground regions of both support and query images have similar contour shapes and internal features, so they can reflect the same category. Please refer to Supp for the adjustment details.

Curriculum Pretraining. The synthetic support-query pairs can be directly used for pretraining. However, it is observed that this approach will result in a slow convergence rate. A reason for this problem is that the LLM has not been previously trained on image data, so optimizing it to directly grasp a complex image processing task is challenging. To address this issue, we propose a progressive pretraining approach inspired by the success of curriculum learning [54], 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, 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 support and query, while reducing the internal difference between foreground and background within each image. This makes it more challenging for LLM to perform few-shot guidance and partition foreground-background areas as pretraining progresses. ***2) Polygon generation.*** We observe that generating a polygon represented by a combination of vertex coordinates is another challenge for the LLM. Therefore, we adopt a progressive strategy here as well. Instead of training the model to directly predict the coordinates of all 16 points of the polygon, we randomly provide the coordinates of M points in the instruction and let the LLM to predict the coordinates of the remaining $16 - M$ points. During pretraining, we gradually decrease the value of M from 15 to 0. This 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 points for segmentation. Experimental results show that this curriculum learning approach allows the model to converge better and achieve higher results. Please see Supp 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.

4. Experiments

4.1. Implementation Details and Training Settings

We set the number of queries in the Q-former to 100 and the threshold α in Eq.1 to 0.2. The ground truth of polygon vertices is obtained in polar coordinates [60]. Specifically, starting from the object center, 16 rays are uniformly emitted at equal angular intervals $\triangle\theta = 22.5°$. The points of intersection between these rays and the object contour are taken as the ground truth of the polygon vertices. More implementation details about pseudo sample generation and curriculum pretraining are presented in Supp.

The overall model is trained in three stages. In the first stage, we freeze the LLM, pretrain the Q-former and fully-connected layers for 100K steps using the datasets[4] and methods in Blip2 [25]. 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 for 60k steps. In the third stage, we train the fully-connected layers, LLM and refinement network on the realistic few-shot segmentation dataset for 25 epochs. AdamW [36] is used as the optimizer with the cosine annealing schedule and an initial learning rate of 0.0002. The model is trained on 16 A100 GPUs.

4.2. Comparison with State-of-the-arts

Table.1 presents the comparisons with other few-shot segmentation methods on two datasets: PASCAL-5^i and

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

Dataset	Method	Conference	1-shot					5-shot				
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
PASCAL-5^i	NTRENet [33]	CVPR'22	65.4	72.3	59.4	59.8	63.2	66.2	72.8	61.7	62.2	65.7
	BAM[23]	CVPR'22	69.0	73.6	67.5	61.1	67.8	70.6	75.1	70.8	67.2	70.9
	AAFormer[57]	ECCV'22	69.1	73.3	59.1	59.2	65.2	72.5	74.7	62.0	61.3	67.6
	SSP[8]	ECCV'22	60.5	67.8	66.4	51.0	61.4	67.5	72.3	75.2	62.1	69.3
	IPMT[34]	NeurIPS'22	72.8	73.7	59.2	61.6	66.8	73.1	74.7	61.6	63.4	68.2
	ABCNet[56]	CVPR'23	68.8	73.4	62.3	59.5	66.0	71.7	74.2	65.4	67.0	69.6
	HDMNet [45]	CVPR'23	71.0	75.4	68.9	62.1	69.4	71.3	76.2	71.3	68.5	71.8
	MIANet[64]	CVPR'23	68.5	75.8	67.5	63.2	68.7	70.2	77.4	70.0	68.8	71.7
	MSI[40]	ICCV'23	71.0	72.5	63.8	65.9	68.5	73.0	74.2	70.5	66.6	71.1
	SCCAN[62]	ICCV'23	68.3	72.5	66.8	59.8	66.8	72.3	74.1	69.1	65.6	70.3
	LLaFS	CVPR'24	74.2	78.8	72.3	68.5	73.5	75.9	80.1	75.8	70.7	75.6
COCO-20^i	NTRENet[33]	CVPR'22	36.8	42.6	39.9	37.9	39.3	38.2	44.1	40.4	38.4	40.3
	BAM[23]	CVPR'22	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
	SSP[8]	ECCV'22	35.5	39.6	37.9	36.7	47.4	40.6	47.0	45.1	43.9	44.1
	AAFormer[57]	ECCV'22	39.8	44.6	40.6	41.4	41.6	42.9	50.1	45.5	49.2	46.9
	MM-Former[68]	NeurIPS'22	40.5	47.7	45.2	43.3	44.2	44.0	52.4	47.4	50.0	48.4
	IPMT[34]	NeurIPS'22	41.4	45.1	45.6	40.0	43.0	43.5	49.7	48.7	47.9	47.5
	ABCNet[56]	CVPR'23	42.3	46.2	46.0	42.0	44.1	45.5	51.7	52.6	46.4	49.1
	HDMNet [45]	CVPR'23	43.8	55.3	51.6	49.4	50.0	50.6	61.6	55.7	56.0	56.0
	MIANet[64]	CVPR'23	42.5	53.0	47.8	47.4	47.7	45.8	58.2	51.3	51.9	51.7
	MSI[40]	ICCV'23	42.4	49.2	49.4	46.1	46.8	47.1	54.9	54.1	51.9	52.0
	SCCAN[62]	ICCV'23	40.4	49.7	49.6	45.6	46.3	47.2	57.2	59.2	52.1	53.9
	LLaFS	CVPR'24	47.5	58.8	56.2	53.0	53.9	53.2	63.8	63.1	60.0	60.0

Table 1. Performance comparison with other methods on PASCAL-5^i and COCO-20^i.

Method	mIoU
LLaFS	74.2
LLaFS w/o support images	56.9

Table 2. Effectiveness of support images.

Method	mIoU
LLaFS	74.2
LLaFS w/o class attributes	70.7
LLaFS w/o region-attribute corresponding table	69.7
LLaFS w/o instruction refinement	70.6
LLaFS w/o iterative refinement	71.8

Table 3. Ablation results of fine-grained in-context instruction.

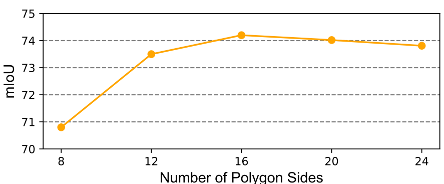


Figure 5. Ablation for the number of polygons' sides.

Method	mIoU
LLaFS	74.2
LLaFs w/o segmentation task instruction w/ abstract summary	67.7
LLaFS w/o fine-grained in-context instruction	67.0
LLaFS w/o refinement network	69.1
LLaFS w/o pseudo-sample-based curriculum pretraining	63.5

Table 4. Effectiveness of different components in our LLaFS.

COCO-20^i. Following previous papers, we use different folds as the test set, with the remaining folds utilized for training. This approach yields four sets of experimental results along with their mean result. Across all datasets and experimental settings, our method consistently outperforms others and achieves a remarkably significant improvement compared to previous state-of-the-art results. For instance, on PASCAL-5^i, LLaFS improves mIoUs by 4.1% and 3.8% in the 1-shot and 5-shot settings respectively compared to the second-ranking method. Notably, our approach still exhibits great advantages on the more complex and challenging COCO-20^i dataset, with increases of 3.9% and 4.0% in the 1-shot and 5-shot settings respectively. This could be attributed to the rich prior knowledge embedded in the LLM and our carefully designed instructions, which enable the models to handle complex images effectively and robustly. These results demonstrate the outstanding performance of our method and highlight the huge potentiality of using LLMs for tackling few-shot segmentation tasks.

4.3. Ablation Study

Using the 1-shot setting of the PASCAL-5^i dataset, we perform several ablation studies to evaluate different components and designs in our method. More ablation studies are presented in **Supp**.

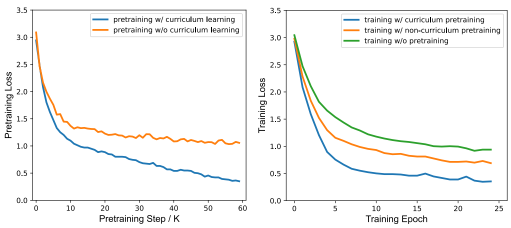
Effectiveness of Different Components We validate the effectiveness of key components in our method, including (1) the segmentation task instruction, (2) fine-grained in-context instruction, (3) the refinement network, and (4) the pseudo-sample-based curriculum pretraining. Experimental results are presented in Table.4. Replacing the detailed segmentation task introduction with an abstract summary 'perform image segmentation' decreases mIoU by 6.5%. Not using the other components individually decreases the mIoU by 7.2%, 5.1%, and 10.7%, respectively. These results demonstrate the importance and effectiveness of each component in our approach.

Effectiveness of Support Images. In the instruction, the annotated support images provide the LLM with crucial visual guidance. As shown in Table.2, if we do not use the support image as a demonstration example, mIoU decreases by 17.3%. This highlights the importance of the support image as a few-shot sample in our approach, demonstrating that our LLsFS benefits not solely from LLM's prior knowledge in an open-vocabulary manner but indeed gains further improvement from the provided few-shot samples.

Number of Polygon's Sides. In the segmentation task instruction, we represent the segmentation output mask as a region enclosed by a 16-point polygon. We find that the number M of sides in the polygon is also a factor affecting the model's performance. Therefore, we test the relationship between mIoU and M and present the results in Fig.5. We observe that when M is small, the model's performance is suboptimal. This is because polygons with a smaller number of sides cannot accurately describe object edges. As M increases, the mIoU gradually improves. However, when M exceeds 16, we observe a slight decrease in performance. This could be because a larger M increases the task's complexity for LLM to tackle. Based on the results,

Method	mIoU
LLaFS	74.2
LLaFS w/o pseudo samples	63.5
LLaFS w/o curriculum strategy	67.3
LLaFS w/ random pseudo query generation	63.9

Table 5. Ablation of pseudo-sample-based curriculum pretraining.



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

Figure 6. 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)

we chose $M = 16$ as our setting.

Ablation of Fine-grained In-context Instruction. Table.3 presents the evaluation of different components and designs in our fine-grained in-context instruction, including (1) class attributes, (2) attribute-region corresponding table, (3) expert-guide framework for instruction refinement, and (4) iterative refinement. We observe that when these components are removed, mIoU decreases by 3.5%, 4.5%, 3.6% and 2.4%, respectively. These results demonstrate the rationality of our designs in this instruction and their effectiveness in improving performance.

Ablation of Pseudo-sample-based Curriculum Pretraining. We further evaluate the key techniques in our pseudo-sample-based curriculum pretraining mechanism and the results are presented in Table.5. (1) When we do not employ pseudo-samples for pretraining, mIoU decreases by 10.7%. (2) Removing the curriculum training strategy that gradually increases the training task difficulty reduces mIoU by 6.9%. (3) 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, mIoU decreases by 10.3%. These results demonstrate the effectiveness of our method's designs.

4.4. Loss Curves

In Fig.6, we present the loss curves during the pretraining and training stages. We observe from Fig.6(a) 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 convergence. After using our curriculum learning mechanism, the model achieves significantly better convergence. Furthermore, in Fig.6(b), we compare the loss reduction during the training stage when using curriculum pretraining, non-curriculum pretraining, and not using pretraining. The model without pretraining converges the slowest, while the model with curriculum pretraining converges the fastest during the training stage. These results demonstrate the effectiveness of our curriculum-learning-based pretraining approach in enhancing the model's convergence speed.

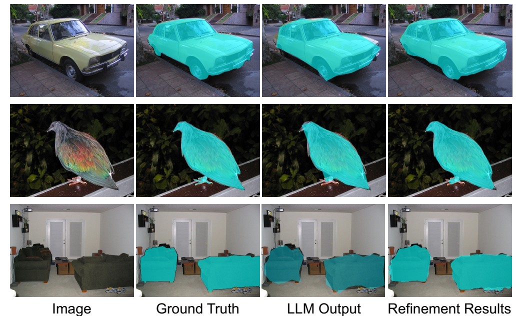


Figure 7. Visualization of segmentation results.

4.5. Visualization

In Fig.7, we present visual samples of the segmentation results from LLaFS, including the image, ground truth, the LLM output, and the results after refinement. It can be observed that the polygons output by LLM already achieve good segmentation performance, and the results after refinement are more accurate, particularly in the object's edge regions. It is worth noting that, as shown in the third row of Fig.7, our method still achieves high-performance segmentation when there are more than one target object in the image. These results demonstrate the excellent performance of LLaFS.

5. Conclusion

This paper proposes LLaFS, a novel framework that for the first time, leverages large language models (LLMs) to address few-shot segmentation in an end-to-end manner. To enable LLMs to perform this visual task, we introduce a segmentation task instruction to provide detailed task definitions, and propose a fine-grained in-context instruction to simulate human cognitive mechanisms and provide refined multimodal reference information. We also propose a pseudo-sample-based curriculum pretraining mechanism to augment the training samples required for instruction tuning. Our extensive experiments demonstrate the effectiveness of LLaFS, which achieves significantly superior state-of-the-art results across multiple datasets and settings. We consider LLaFS as an important exploration towards an LLM framework capable of addressing few-shot tasks in different modalities beyond natural language processing.

Acknowledgement This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-08-006), MOE AcRF Tier 2 projects MOE-T2EP20222-0009 and MOE-T2EP20123-0014.

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13979–13988, 2021. 2

[2] 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. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2

[4] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3367–3375, October 2023.

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[8] Qi Fan, Wenjie Pei, Yu-Wing Tai, and Chi-Keung Tang. Self-support few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 701–719. Springer, 2022. 2, 7

[9] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Ai-generated content (aigc) for various data modalities: A survey, 2023. 1

[10] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022. 2

[11] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022. 2, 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[13] Shuting He, Henghui Ding, and Wei Jiang. Primitive generation and semantic-related alignment for universal zero-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11238–11247, 2023. 1

[14] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2022. 2

[15] 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 *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2

[16] 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 *International Conference on Learning Representations*, 2021. 2

[17] Kai Huang, Feigege Wang, Ye Xi, and Yutao Gao. Prototypical kernel learning and open-set foreground perception for generalized few-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19256–19265, 2023. 1, 2

[18] Deyi Ji, Haoran Wang, Hanzhe Hu, Weihao Gan, Wei Wu, and Junjie Yan. Context-aware graph convolution network for target re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1646–1654, 2021. 2

[19] Deyi Ji, Haoran Wang, Mingyuan Tao, Jianqiang Huang, Xian-Sheng Hua, and Hongtao Lu. Structural and statistical texture knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16876–16885, 2022.

[20] Deyi Ji, Feng Zhao, Hongtao Lu, Mingyuan Tao, and Jieping Ye. Ultra-high resolution segmentation with ultra-rich context: A novel benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23621–23630, 2023. 2

[21] Siyu Jiao, Gengwei Zhang, Shant Navasardyan, Ling Chen, Yao Zhao, Yunchao Wei, and Humphrey Shi. Mask matching transformer for few-shot segmentation. In *Advances in Neural Information Processing Systems*. 2

[22] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021. 2

[23] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8057–8067, 2022. 2, 7

[24] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021. 1, 2

[25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Inter-*

national Conference on Machine Learning, 2023. 2, 6
[26] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 4
[27] Binghao Liu, Jianbin Jiao, and Qixiang Ye. Harmonic feature activation for few-shot semantic segmentation. *IEEE Transactions on Image Processing*, 30:3142–3153, 2021. 1
[28] Fang Liu, Yuhao Liu, Yuqiu Kong, Ke Xu, Lihe Zhang, Baocai Yin, Gerhard Hancke, and Rynson Lau. Referring image segmentation using text supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22124–22134, October 2023. 1
[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2, 5
[30] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18653–18663, 2023. 3
[31] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4173, 2020. 2
[32] 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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, 2022. 2
[33] Yuanwei Liu, Nian Liu, Qinglong Cao, Xiwen Yao, Junwei Han, and Ling Shao. Learning non-target knowledge for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11573–11582, 2022. 7
[34] Yuanwei Liu, Nian Liu, Xiwen Yao, and Junwei Han. Intermediate prototype mining transformer for few-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 35:38020–38031, 2022. 2, 7
[35] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 142–158. Springer, 2020. 2
[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6
[37] Chaofan Ma, Yuhuan Yang, Chen Ju, Fei Zhang, Ya Zhang, and Yanfeng Wang. Open-vocabulary semantic segmentation via attribute decomposition-aggregation. *arXiv preprint arXiv:2309.00096*, 2023. 1
[38] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6941–6952, 2021. 1, 2
[39] 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. 2, 3
[40] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19266–19276, 2023. 7
[41] Gregory L Murphy and Douglas L Medin. The role of theories in conceptual coherence. *Psychological review*, 92(3):289, 1985. 3
[42] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 622–631, 2019. 1, 2
[43] Atsuro Okazawa. Interclass prototype relation for few-shot segmentation. In *European Conference on Computer Vision*, pages 362–378. Springer, 2022. 1, 2
[44] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023. 2, 3
[45] Bohao Peng, Zhuotao Tian, Xiaoyang Wu, Chengyao Wang, Shu Liu, Jingyong Su, and Jiaya Jia. Hierarchical dense correlation distillation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23641–23651, 2023. 2, 7
[46] 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. 2
[47] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5249–5258, 2019. 2
[48] 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. 2
[49] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 44(2):1050–1065, 2020. 1
[50] 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. 1, 2
[51] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104:154–171, 2013. 4
[52] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019. 2

[53] 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. 2, 3

[54] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2021. 6

[55] 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. 2

[56] Yuan Wang, Rui Sun, and Tianzhu Zhang. Rethinking the correlation in few-shot segmentation: A buoys view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7192, 2023. 1, 7

[57] Yuan Wang, Rui Sun, Zhe Zhang, and Tianzhu Zhang. Adaptive agent transformer for few-shot segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 36–52. Springer, 2022. 7

[58] Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. Multitask prompt tuning enables parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022. 2

[59] Edward J Wisniewski and Bradley C Love. Relations versus properties in conceptual combination. *Journal of memory and language*, 38(2):177–202, 1998. 3

[60] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12193–12202, 2020. 6

[61] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1, 2

[62] Qianxiong Xu, Wenting Zhao, Guosheng Lin, and Cheng Long. Self-calibrated cross attention network for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1, 7

[63] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. Mining latent classes for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8721–8730, 2021. 2

[64] Yong Yang, Qiong Chen, Yuan Feng, and Tianlin Huang. Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7131–7140, 2023. 1, 2, 7

[65] Ying Zang, Chenglong Fu, Runlong Cao, Didi Zhu, Min Zhang, Wenjun Hu, Lanyun Zhu, and Tianrun Chen. Resmatch: Referring expression segmentation in a semi-supervised manner. *arXiv preprint arXiv:2402.05589*, 2024. 2

[66] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9587–9595, 2019. 1

[67] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019. 1

[68] Gengwei Zhang, Shant Navasardyan, Ling Chen, Yao Zhao, Yunchao Wei, Honghui Shi, et al. Mask matching transformer for few-shot segmentation. *Advances in Neural Information Processing Systems*, 35:823–836, 2022. 7

[69] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1

[70] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 1

[71] Lanyun Zhu, Tianrun Chen, Jianxiong Yin, Simon See, and Jun Liu. Continual semantic segmentation with automatic memory sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3082–3092, June 2023. 1

[72] Lanyun Zhu, Tianrun Chen, Jianxiong Yin, Simon See, and Jun Liu. Learning gabor texture features for fine-grained recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1621–1631, 2023. 1

[73] Lanyun Zhu, Deyi Ji, Tianrun Chen, Peng Xu, Jieping Ye, and Jun Liu. Ibd: Alleviating hallucinations in large vision-language models via image-biased decoding. *arXiv preprint arXiv:2402.18476*, 2024. 1

[74] Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12537–12546, 2021. 2