LANGUAGE-GUIDED FEW-SHOT SEMANTIC SEGMENTATION

Jing Wang[⋆] *Yuang Liu* [†] *Qiang Zhou* [⋆] *Fan Wang*[⋆]

[⋆] DAMO Academy, Alibaba Group
[†] East China Normal University

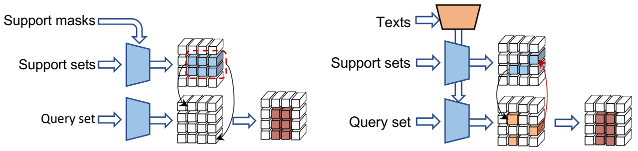
ABSTRACT

Few-shot learning is a promising way for reducing the label cost in new categories adaptation with the guidance of a small, well labeled support set. But for few-shot semantic segmentation, the pixel-level annotations of support images are still expensive. In this paper, we propose an innovative solution to tackle the challenge of few-shot semantic segmentation using only language information, *i.e.*image-level text labels. Our approach involves a vision-language-driven mask distillation scheme, which contains a vision-language pretraining (VLP) model and a mask refiner, to generate high quality pseudo-semantic masks from text prompts. We additionally introduce a distributed prototype supervision method and complementary correlation matching module to guide the model in digging precise semantic relations among support and query images. The experiments on two benchmark datasets demonstrate that our method establishes a new baseline for language-guided few-shot semantic segmentation and achieves competitive results to recent vision-guided methods.

Index Terms— Few-shot learning, semantic segmentation, vision-language

1. INTRODUCTION

Semantic segmentation is an important task of pattern recognition [1], which aims to allocate a category label to each pixel. With the development of deep learning, the accuracy of semantic segmentation has risen dramatically, but with the growing need of large-scale dense labels. Meanwhile, the well-trained model cannot be directly applied to new categories until re-training. Few-shot learning is a recent trending topic who aims to solve the label shortage and quick adaptation problem in deep learning. Instead of training a task-specialized model from scratch, few-shot learning tries to train a task-independent model in a “meta-learning” paradigm to dig the common knowledge shared across different tasks [2]. The model can be easily adapted to new tasks with a few support samples after “meta-training”. Many researchers have explored the efficiency of few-shot on classification [3, 4, 5], object detection [6, 7, 8], and semantic segmentation [9, 10, 11]. Even the few-shot learning can decrease the cost of adaptation to new tasks, the “meta-learning” process requests a sufficient amount of well-labeled base data. Comparing to the image-level text label and the bounding box label, the pixel-wise dense segmentation map adopted in semantic segmentation is harder to acquire. In this paper, we consider a more valuable and challenging situation in few-shot semantic segmentation (FSS), *i.e.*, language-guided few-shot semantic segmentation (LFSS), where only the image-level labels are available.



(a) Regular FSS (b) Language-guided FSS

Fig. 1: Comparison between the regular few-shot segmentation (a) and the proposed language-guided approach (b). In regular few-shot, ground-truth of support masks are adopted to select representative features in support feature maps, and target features in query feature maps will be picked through a singe-direction matching. In the proposed method, the cheap but abstract text labels are adopted to mark target features in support and query images generally, then the double-direction matching can help to pick more accurate target features.

The LFSS is rarely studied because of the information scarcity. Instead of dense masks, [10, 12, 13] has explored to train the few-shot segmentation model by scribble, bounding box annotations, or sparse pixel annotations. These annotations are more sparse than the pixel-level annotations, but still require a strong artificial prior. [14] firstly introduces class label supervision to FSS, they train the model following regular few-shot learning (fully-annotated support masks are necessary), but during testing, they only take the class labels as prior to lead the nearest neighbor classification and generate a general support proposal for object segmentation in query images. [15] propose a novel multi-modal interaction module for few-shot segmentation, they design a co-attention mechanism to align the visual input and natural word embedding. To explore more information from the text labels, [16] conduct the efficient classification activation maps (CAM) [17] to extract pseudo masks from category text labels as supervision. Due to the inaccurate pseudo masks and the gap between visual and text embedding, the performance of these language-guided works are far away from the vision-guided methods.

Recently, [18] has expanded the VLP model to few-shot learning, where they treat CLIP [19] as an efficient classifier and conduct CAM to generate more accurate pseudo masks from text prompts. These pseudo masks directly take place of the ground-truth support masks to train the few-shot model. However, as pseudo masks can't be as subtle as the manual labels, training few-shot model in fully-supervised manner with them is suboptimal. In this paper, we propose a Language-guided Few-shot semantic Segmentation model (**LFSS**). It consists of a VLP-driven mask distillation **(VLMD)** mechanism for generating high quality pseudo masks and a custom feature learning module for digging exact guidance from coarse pseudo masks. Firstly, We employ MaskCLIP [20], a semantic segmentation model expanded from CLIP [19], to transfer text labels into pseudo masks. We then adopt a mask refiner to remove false mask predictions. In vision-guided few-shot semantic segmentation, prototype learning is a widely adopted method where masked average pooling (MAP) extract one or few class prototypes from the regions of interest (ROI) in support feature maps. Matching support prototypes with query features can acquire the semantic similar target features [5, 21, 22]. However, in LFSS, the coarse pseudo masks will lead to inaccurate prototypes. To address this, we have designed a distributed prototype supervision **(DPS)** and a complementary correlation matching **(CCM)** module to reduce the effect of the pseudo mask and reveal the correct semantic relations among the support and query images.

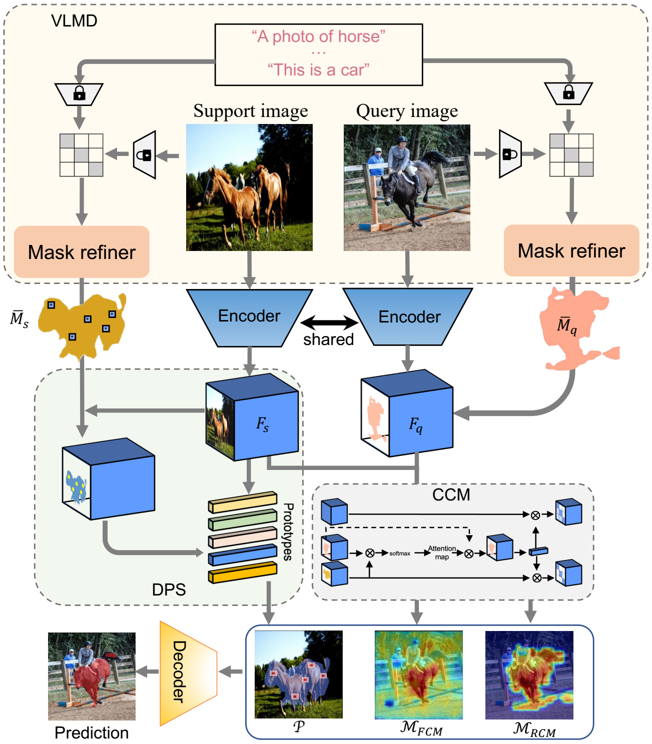


Fig. 2: Overview of our LFSS framework, which consists of the proposed vision language pre-training model-driven mask distillation (VLMD), distributed prototype supervision module (DPS), and complementary correlation matching module (CCM).

2. RELATED WORKS

2.1. Few-shot Semantic Segmentation

Many methods have been proposed to aggregate the guidance from support images to segment new objects of the same class in query images in few-shot style. For example, extracting representative prototypes from support feature maps by masked average pooling [5, 21, 22], calculating pixel-wise correlation between support and query features [23, 24], and so on. However, pixel-wise annotation of support images are required for the regular few-shot segmentation models. To further reduce the label cost of training, language-guided methods are proposed in few-shot segmentation. [10, 12, 13] tried to train the model with sparse labels like bounding box or scribbles. [14] firstly proposed to train a regular few-shot segmentation model on base but testing it with only class labels. [15] explored the effectiveness of combining the visual embedding with text embedding in few-shot segmentation. To reduce the gap between vision and text, [16] extracted pseudo masks from text labels by CAM, and [18] introducing the powerful vision-language pretraining model CLIP to transfer the text labels into pseudo masks and achieved comparable performance to the fully-supervised few-shot segmentation model.

2.2. Vision-Language Model

As a pioneering work towards vision-language pre-training, CLIP [19] has promoted a wide range of multi-modal applications [25, 26, 27] and shows great potential in zero-shot or few-shot vision tasks [28, 29, 30]. Especially, a group of researchers has extended it into dense prediction tasks, *e.g.*, semantic segmentation [31, 29], image generation [26] and object detection [28]. DenseCLIP [31] is the pioneer that employs CLIP in semantic segmentation and tickles the issue of pixel-text matching via context-aware prompting. Ding *et al.* [29] decouples the zero-shot segmentation task as a class-agnostic grouping task and a zero-shot classification task to perform segment-text matching. However, the above methods all depend on complicated prompt engineering, and are limited to the lack of fine-annotated images.

3. METHOD

3.1. Problem setup

For a regular 1-way K-shot few-shot segmentation task T, a support set $S = (I_s, M_s)$ and a query set $Q = (I_q, M_q)$ are required, where I and M represent image and ground truth mask respectively, $|S| = K$, S and Q are sampled from the same category. The goal is to train model who can predict M_q for I_q with a given S, subject to K being small for few-shots. In this paper, we consider a more challenging setting in few-shot semantic segmentation, where only the text class labels (L) of the support images are available, *i.e.*$S = (I_s, L_s)$ and $Q = (I_q, L_q)$. We adopt the widely used episodic training paradigm to train our model, where datasets are split into D_{train} and D_{test} with category set C_{train} and C_{test} respectively, and $C_{train} \bigcap C_{test} = \phi$. We repeatedly sample task T from D_{train} during training, and the trained model are directly evaluated on D_{test} to predict M_q for I_q *i.e.*:

$$\hat{M_q} = f(\{(I_s^k, L_s^k)\}_{k=1}^{K}, I_q, \theta \| c \in C_{test}) \tag{1}$$

where $f(, \|\theta)$ is the trained model.

3.2. Overview

In this work, we aim to train an accurate few-shot semantic segmentation model with only text labels. The overall architecture of the proposed method are shown as Figure 2, which is a double-branch architecture, consisted of the vision language pre-training driven mask distillation module (VLMD) and a custom feature learning stream. The support and query images along with language descriptions are first fed to the VLMD to extract reliable pseudo masks. At the same time, the backbone will extract multi-level features from support and query images respectively. Then with the guidance of the pseudo masks, a distributed prototype supervision (DPS) module are applied on the support features to extract local representative prototypes and a complementary correlation matching (CCM) module learns to generate a fine-grained correlation map by matching the query and support features. We will take the calculation process of one-shot as example to introduce these effective modules amply in the follow sections.

3.3. VLP-driven Mask Distillation (VLMD)

As text labels are abstract and information-limited, acquiring more information from them poses the first challenge. To tackle this, we introduce a VLMD module to project the text labels to pseudo masks, which consisting of a mask generator and mask refiner. For segmenting targets annotated by text labels in images, we adopts the VLP model, MaskCLIP [20], to generate high quality pseudo masks. Specifically, we adopt the modified ResNet as image encoder, then we remove the query and key embedding layers from the last global attention pooling layer, and directly feed the feature map from the final residual block into the value-embedding layer and the following linear layer, which are reformulated into two respective 1 × 1 convolution layers to keep the spatial dimension of feature maps (this process can be visualized in Fig.2(b) of [20]). The text encoder are unchanged. The cosine similarity between the text embedding and the image feature maps can tell the category of each pixel.

However, despite the MaskCLIP model can help to generate high quality pseudo masks, they can not as elaborate as the manual masks used in regular few-shot segmentation. To reduce false predictions in these pseudo masks, we introduce a mask refiner to improve their accuracy.Leveraging the notion that pixels belonging to the same object are more similar than those to different objects of same class, we adopt a self-supported approach to refine the initial pseudo masks. As illustrated in Figure 3, features extracted from the backbone can be separated into foreground and background features based on the initial pseudo masks, then we conduct MAP to aggregate the respective foreground prototypes and background prototypes from support and query feature maps:

$$P^f = \frac{\sum_{x=1,y=1}^{w,h} F_{x,y} \odot M_{x,y}}{\sum_{x=1,y=1}^{w,h} M_{x,y}} \, , \tag{2}$$

where $F \in R^{c \times w \times h}$ represents features that extracted by backbone network, $\odot$ is Hadamard product. As the query and support images should contain objects of the same class, we add the foreground prototypes to weight the specified objects, formulated as follows:

$$P_m^f = \alpha P_s^f + (1 - \alpha) P_q^f \, , \tag{3}$$

α is the balance factor, s and q represent support and query set respectively. Different from the foreground, the background prototypes are independently responsible for corresponding feature maps as the background between support and query sets are quit different, and a self-attention operation is adopted to acquire the background prototypes:

$$P^b = \texttt{softmax}(F_b \cdot F^\top) \cdot F_b \, , \tag{4}$$

where $F_b = F \odot (1 - M)$represents the background features, F represents the full feature map. Then we calculate the cosine similarity between the features and prototypes to obtain new masks:

$$S^f = \frac{F \cdot P^\top}{\|F\| \, \|P\|} \, , \tag{5}$$

where $P \in \{P_m^f, P_b\}$. Then the features are assigned to foreground or background according to the similarity score. After mask refinement, most false predictions of the initial masks can be removed, acquired the refined support mask $\bar{M}_s$ and query mask $\bar{M}_q$.

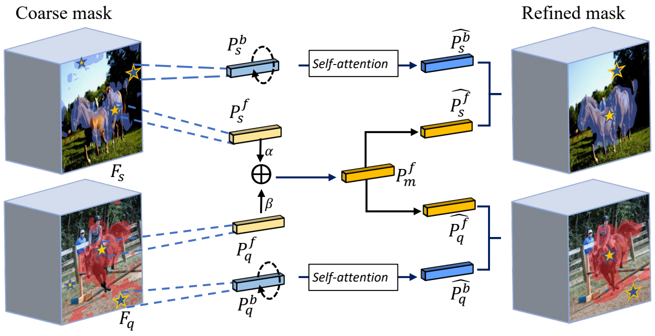


Fig. 3: The detail of self-supported mask refinement module.

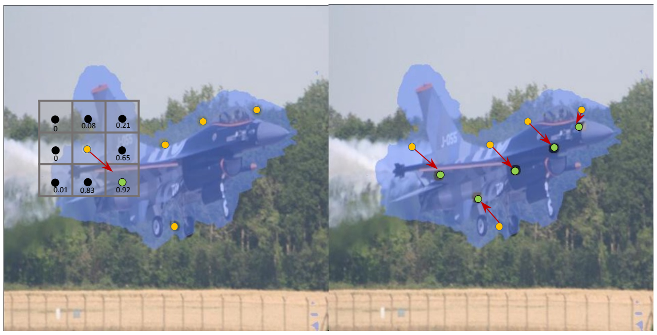


Fig. 4: The detail of selecting the super-pixel seeds.

3.4. Distributed Prototype Supervision (DPS)

Prototype learning is a popular feature alignment method in few-shot segmentation [32, 11, 22]. Typically, all foreground support features are compressed into a global prototype by MAP (refer to Eq.(1)), which is semantically rich but lack of spatial information. To solve this issue, we designed a custom Distributed Prototype Supervision (DPS) module, which extracts multiple local prototypes from the coarse pseudo masks instead of a global prototype. As shown in Figure 4, we first distribute N_{sp} initial seed points in the pseudo mask, where a Euclidean distance transform is adopted to place the seed points far from the boundary of mask and other seed points:

$$D(x, y) = \min_{l \in L} \sqrt{((x - x_l)^2 + (y - y_l)^2)}\,, \tag{6}$$

where x and y represent the spatial coordinate values of a seed, the max $D(x, y)$ represents the furthest distance. $L = B \cup P$ represents the background feature points(B) and the labeled points (P), the selected points will be added to P after each iteration. After placing the initial seed points, we extract corresponding features in feature map as super-pixel seeds:$O^0 \in \{R^{C \times N_{sp}}\}$ (C is the channels of feature map). To prevent the incorrect placement of seed points in the background region of pseudo coarse mask, we utilize a part-aware module to rectify the location of the initial seed points. As shown in Fig 4, after placing a seed point, we sample an $n * n$ grid, *i.e.*G, around it and calculate the similarity between features locate in G and the P_q^f:

$$S_{i,j} = \frac{g_{i,j} \cdot (P_q^f)^\top}{\|g_{i,j}\|_2 \left\|P_q^f\right\|_2}\,, \tag{7}$$

where $S_{i,j}$ represents the similarity score, $g_{i,j} \in \mathbb{R}^{1 \times C}$ means support features locate at (i, j) in G. The point whose corresponding feature with the highest similarity score will replace the original seed, formulated as follows:

$$\hat{i}, \hat{j} = \texttt{argmax}_{i,j}(S_{i,j})\,. \tag{8}$$

After adjusting the seed points, we assume that all seeds are located at target objects and extract the new super-pixel seeds $O^0 \in \{R^{C \times N_{sp}}\}$. To extract semantic prototypes, we cluster the feature map into N_{sp} super-pixel with guidance of the super-pixel seeds. We firstly add coordinates of each pixel to the feature maps to increase spatial priors. Then we cluster feature maps in an iterative manner. During each iteration, we first calculate the correlation map C^t between each foreground feature point p and all super-pixel seeds:

$$C_{p,i}^t = e^{-Q(F_p, O_i^{t-1})}\,, \tag{9}$$

where F_p represents foreground pixels, $i \in N_{sp}$. Q is a distance function defined as:

$$Q(F, O) = \sqrt{(d_f(F_1, F_2))^2 + \left(\frac{d_s(O_1, O_2)}{r}\right)^2} \tag{10}$$

where d_f and d_s are Euclidean distance for features and coordinate values, r is a temperature value [33]. Then we update the super-pixel centroids following:

$$O_i^t = \frac{1}{\sum_{N_{fp}} C_{p,i}^t} \sum_{p=1}^{N_{fp}} C_{p,i}^t F_p\,, \tag{11}$$

where N_{fp} is the foreground pixels number. After clustering, the resulting super-pixel centroids are treated as the part-aware prototypes, dubbed as P_{sc}. Instead of expanding the prototypes to specified shape and concatenating them with feature maps, we calculate association map between the P_{sc} and support feature map instead:

$$\mathcal{P} = \sum_{i}^{N_{sp}} \frac{P_{sc} \cdot F_s}{\|P_{sc}\| \, \|F_s\|}\,. \tag{12}$$

3.5. Complementary Correlation Matching (CCM)

Even prototypes work effectively in matching objects with semantic similarity, but the sparse nature stops them from fine-grained relation exploitation. To make better use of the pseudo mask, we proposed a complementary correlation matching module (CCM), which consisted of a ROI-guided correlation matching (RCM) and a full image correlation matching (FCM).

We first extract an attention map from the query image and support image with the guidance of their pseudo masks, formulated as follows:

$$\mathcal{A} = \texttt{softmax}\left(\frac{(F_q \odot \bar{M}_q) \cdot (F_s \odot \bar{M}_s)^{\top}}{\|F_q \odot \bar{M}_q\| \, \|F_s \odot \bar{M}_s\|}\right) . \tag{13}$$

As the most common part of the query and support images should be the objects of the specified class, we highlight the target area by multiplying the support feature maps with the attention map $\mathcal{A}$ and extracting a more focused prototype P_a by MAP:

$$P_a = \frac{\sum_{x=1,y=1}^{w,h} \mathcal{A}^{x,y}(F_s^{x,y} \bar{M}_s^{x,y})}{\sum_{x=1,y=1}^{w,h} \bar{M}_s^{x,y}} . \tag{14}$$

Then we obtain the ROI-guided correlation map by matching P_a with the masked query feature map:

$$\mathcal{M}_{RCM} = \frac{P_a \cdot (F_q \odot \bar{M}_q)^{\top}}{\|P_a\| \, \|F_q \odot \bar{M}_q\|} . \tag{15}$$

The $\mathcal{M}_{RCM}$ helps locate exact objects in query image from the coarse masked ROI, however, it's isolated from those omitted by the pseudo masks. To solve this problem, we further extract the FCM by matching all query features with the P_a:

$$\mathcal{M}_{FCM} = \frac{P_a \cdot F_q^{\top}}{\|P_a\| \, \|F_q\|} . \tag{16}$$

The RCM and FCM works complementarily to detect all targets in query images, so we concatenate them together to get the fine-grained correlation map:

$$\mathcal{M} = \mathcal{M}_{RCM} \oplus \mathcal{M}_{FCM} , \tag{17}$$

where $\oplus$ is the channel-wise concatenation operation. Finally, we concatenate the query features F_q with prototype-associated map $\mathcal{P}$ and the fine-grained correlation map $\mathcal{M}$ to obtain more guidance, thus the final feature map $\mathcal{F}$ that fed to the decoder is:

$$\mathcal{F} = F_q \oplus \mathcal{P} \oplus \mathcal{M} . \tag{18}$$

The final prediction is acquired by:

$$\hat{M}_q = \textbf{Dec}(\mathcal{F}) , \tag{19}$$

Dec is a light-weight decoder.

3.6. Objective Function

The binary cross entropy (BCE) loss is adopted to train the model. To speed up convergence, we employ a circle training strategy. Specifically, the support image is firstly deemed as query image and fed to the model to acquire an $\hat{M}_s$. Then $\hat{M}_s$ is set as the new support mask to support the prediction of query mask $\hat{M}_q$. The overall loss function is formulated as:

$$\mathcal{L} = \beta \mathcal{L}_{BCE}(\hat{M}_s, M_s^{gt}) + (1 - \beta) \mathcal{L}_{BCE}(\hat{M}_q, M_q^{gt}) , \tag{20}$$

where M_s^{gt}/M_q^{gt} represent the ground-truth of support/query sample, β is the balance factor.

Table 1: Comparisons with fully-supervised FSS and LFSS methods on Pascal-5^i. "P", "I" and "B" represent the three types of semantic annotation ("Ann."): Pixel, Image and Box. "BB." means the backbone. The "-" is placeholder for unreported results by original paper.

BB.	Method	Ann.	5^0	5^1	5^2	5^3	Mean
VGG-16	PFENet	P	56.9	68.2	54.4	52.4	58.0
	HSNet	P	59.6	65.7	59.6	54.0	59.7
	PANet	B	-	-	-	-	45.1
	CAM-WFSS	I	36.5	51.7	45.9	35.6	42.4
	IMR-HSNet	I	**58.2**	63.9	52.9	51.2	56.5
	Ours	I	56.3	**65.2**	**53.6**	**55.7**	**57.7**
ResNet-50	PFENet	P	61.7	69.5	55.4	56.3	60.8
	HSNet	P	64.3	70.7	60.3	60.5	64.0
	ASGNet	P	58.8	67.9	56.8	53.7	59.3
	CANet	B	-	-	-	-	52.0
	VS-WFSS	I	42.5	64.8	48.1	46.5	50.5
	IMR-HSNet	I	**62.6**	**69.1**	56.1	56.7	61.1
	Ours	I	59.9	**69.1**	**56.7**	**58.9**	**61.2**

4. EXPERIMENTS

4.1. Experimental Settings

We evaluate our approach on two public datasets that widely enrolled in regular few-shot semantic segmentation, *i.e.*, Pascal-5^i [32] and COCO-20^i [34] (i is the number of folds). Following the setting of few-shot segmentation, we split each dataset into four folds, set three of them as training set and sample 1000 episodes from the remaining fold as test set. The mean intersection over union (mIoU) of all classes is utilized to measure the performance. To fairly compare with state-of-the-art (SOTA) methods, we set the popular convolution neural network VGG-16 and ResNet-50 pretrained on ImageNet as backbone, a light-weight decoder contains an ASPP (atrous spatial pyramid pooling) block and three plain convolution blocks works for the final segmentation. The pretrained MaskCLIP is adopted for initial pseudo masks generation, the visual and text encoders of MaskCLIP are modified ResNet-50. The backbone and MaskCLIP are frozen during model training to prevent overfitting.

For mask generation, we expand the text label with 85 prompt templates followed MaskCLIP [20] and fed them to the text encoder, then average the processed text embeddings of the same class. We resize the input to 400 × 400 in both training and testing stage following [18], and no extra data augmentation trick is adopted. The learning rate is set to 0.001. We train the model for 200 epochs on 8 NVIDIA V100 GPUs with Adam optimizer. The hyperparameters α and β are empirically set at 0.5, and $n = 3$ for saving calculation.

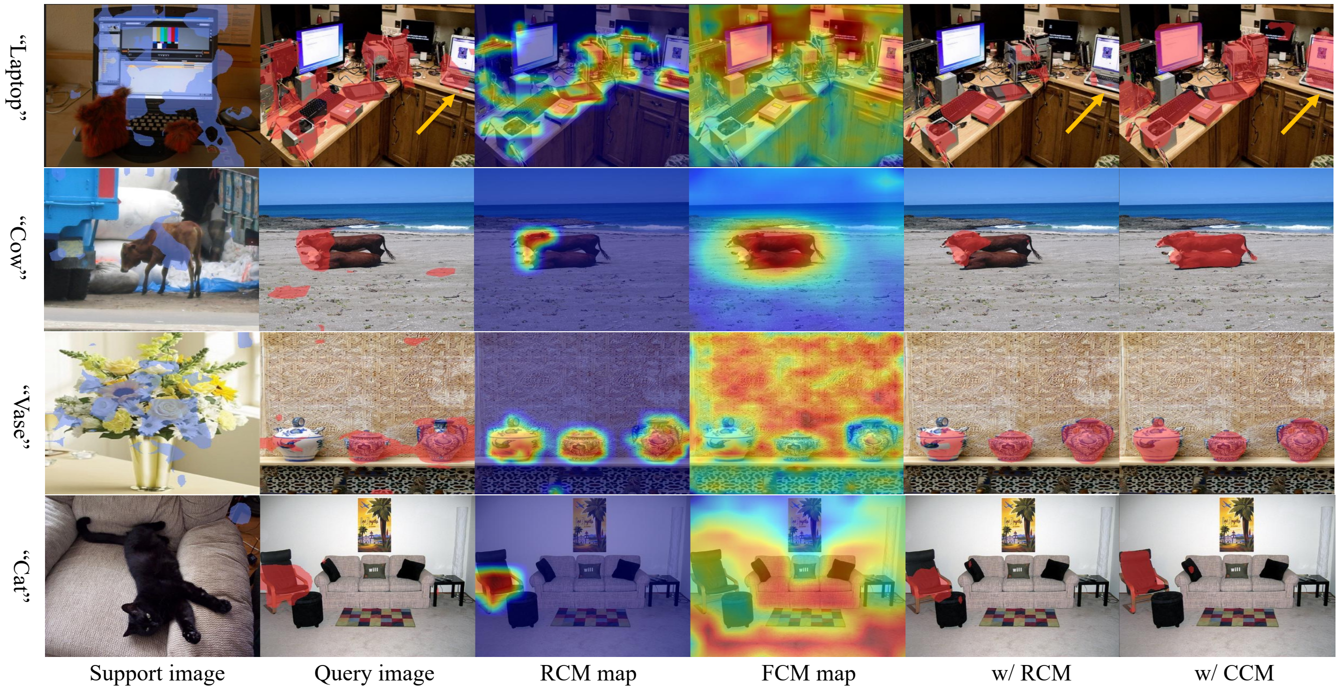


Fig. 5: Visualization of RCM (the 3-rd column), FCM (the 4-th column) map and the corresponding segmentation results with RCM and CCM respectively (the last two columns).

Table 2: Comparisons with fully-supervised FSS and LFSS methods on COCO-20^i.

BB.	Method	Ann.	20^0	20^1	20^2	20^3	Mean
VGG-16	PFENet	P	35.4	38.1	36.8	34.7	36.3
	BAM-base	P	39.0	47.0	46.4	41.6	43.5
	PANet	B	12.7	8.7	5.9	4.8	8.0
	CAM-WFSS	I	24.2	12.9	17.0	14.0	17.0
	IMR-HSNet	I	34.9	38.8	37.0	40.1	37.7
	Ours	I	**37.6**	**49.6**	**42.5**	**43.4**	**43.3**
ResNet-50	PFENet	P	36.5	38.6	34.5	33.8	35.8
	HSNet	P	36.3	43.1	38.7	38.7	39.2
	BAM-base	P	41.9	45.4	43.9	41.2	43.1
	ASGNet	P	-	-	-	-	34.6
	VS-WFSS	I	-	-	-	-	15.0
	IMR-HSNet	I	39.5	43.8	42.4	44.1	42.4
	Ours	I	**42.9**	**51.8**	**44.4**	**46.8**	**46.4**

4.2. Comparison with State-Of-The-Art

We compared our methods with the SOTA LFSS, *i.e.*CAM-WFSS [16], IMR-HSNet [18], VS-WFSS [15], and some recent fully-supervised few-shot segmentation (FFS) works, *i.e.*PFENet [21], PANet [10], HSNet [23], ASGNet [33], BAM-base [22], in 1-shot setting. The results on Pascal-5^i are displayed in Table 1. With only text supervision, all our model with different backbones surpass other LFSS methods and most FSS methods. For FSS, our method surpasses the prototype-based PFENet but is not as excellent as the HSNet, who introduced pixel-level correlation to achieve fine-grained feature alignment. Specially, the proposed method outperforms the language-guided version HSNet, *i.e.*, IMR-HSNet. Our model exceeds the IMR-HSNet with 1.2% and 0.1% mIoU for VGG-16 and resnet-50 backbones respectively. The IMR-HSNet directly adopts HSNet to train the LFSS model but neglects that the gap between the elaborate manual labels and the coarse pseudo masks. Take this in mind, we design this custom network to mitigate the effect of false predictions in pseudo masks and achieve better results.

Table 2 summaries the evaluation results on COCO-20^i, which is a more challenging dataset contains 80 categories, many FSS models' performance dropped on this dataset because of its complexity. However, our method shows excellent generality on COCO, exceed SOTA LFSS method, *i.e.*IMR-HSNet, by a large margin (5.6% with VGG-16 and 4.0 % with ResNet-50). False predictions of pseudo masks are more general in COCO dataset due to its variety. The proposed model designs the VLMD to generate high quality masks and reduce apparent errors, followed by custom DPS and CCM who learn to dig exact information from the pseudo masks to provide more guidance for targets segmentation. As a result, we not only outperform the LFSS, but also surpass recent FSS, *i.e.*, BAM and HSNet. Our method with ResNet-50 backbone im-

Table 3: The mIoU performance between pseudo masks (initial and refined) and the ground-truth.

Dataset	Initial mask	Refined mask
VOC	26.94	32.52
COCO	26.99	33.84

proves 7.2% mIoU over the HSNet on COCO, but lost behind it on Pascal dataset. We infer that the pixel-level correlation proposed by HSNet is not good at extracting key information from complex scenario. As data in COCO is category-diverse and appearance-diverse, the pixel-level correlation is harder to dig and easy to be disturbed by other objects. In our method, the pseudo masks will locate the targets generally and guide the prototype extraction and correlation matching, which eases the few-shot training.

4.3. Ablation Study

Ablation studies are conducted to excavate the effectiveness of each component. We first evaluate the VLMD, Table 3 depicts the mIoU between the pseudo mask and its corresponding ground-truth, we find the pseudo masks become more concise after refinement, the mIoU improves 5.58% and 6.85% on Pascal-5^i and COCO-20^i respectively.

For feature learning module, we adopt ResNet-50 as feature extractor, and set a simple baseline by concatenating the global prototype and RCM as feature map. All models contain the same decoder, and the concatenated feature map is fed to the decoder directly to segment objects. The mean IoU on all categories of Pascal and COCO are summarized in Table 4. Effected by coarse mask, the global prototype contains part of background information, so the results of baseline model are just passable. To improve the effectiveness of the prototype, we firstly replace the global prototype with DPS module to induce the model to focus on specific objects part instead of background, the model performs better on COCO dataset but worse on Pascal according to the results. We find the pseudo masks of Pascal data contain more false predictions as Pascal contains fewer categories while the MaskCLIP tries to annotate every pixel to a category. To curb the effect of false positives in pseudo masks, we distill more accurate masks by self-supported mask refiner. With finer masks, the DPS can extract valuable prototypes from target objects and the RCM can extract more focused association map. Quantitatively, the model's performance improves a lot after received finer masks (5% on Pascal and 2.3% on COCO). Finally, we introduce the CCM to capture more target information and prevent the omission of target in query images. The results are further improved on two dataset (2.4% on Pascal and 6.7% on COCO).

Table 4: Effectiveness of each component in our LFSS framework.

Dataset	DPS	Mask refine	CCM	mIoU (%)
VOC				53.9 (baseline)
	✓			53.6 (↓ 0.3)
	✓	✓		58.6 (↑ 4.7)
	✓	✓	✓	**61.2** (↑7.3)
COCO				36.1 (baseline)
	✓			37.4 (↑1.3)
	✓	✓		39.7 (↑3.6)
	✓	✓	✓	**46.4** (↑10.3)

Moreover, we implement extra studies on VOC-5^1 to find out suitable hyperparameters (α, n) for our model, the results are displayed in Tab 5. It's found that the model achieves best performance when $\alpha = 0.5$ and $n = 3$.

Table 5: Impacts of n and α on first fold of VOC-5^i. We set α=0.5 for testing n and set n=3 in reverse.

n	1	3	5
mIoU	68.0	**69.1**	68.7

α	0.1	0.3	0.5
mIoU	68.2	68.6	**69.1**

4.4. Visualization

To observe the results more intuitively, we visualized the association map generated by CCM, the refined masks, and some segmentation results respectively. As shown in Figure 5, the first two rows display samples that MaskCLIP failed to detect target objects in query images (annotated by yellow arrows), we found that the RCM also omitted these targets effected by the pseudo masks. So the model fail to segment them when only RCM is included (w/ RCM). Fortunately, we found the FCM will help to relocate the omitted objects after a full image matching. The last two rows display samples that with sick quality support masks, we find the RCM works effectively as targets in query images are detected by pseudo masks. Misrecognition and omission of targets are common during mask generation as we directly applied the general MaskCLIP to segment Pascal and COCO without fine-tuning. To this end, we add the $\mathcal{M}_{RCM}$ and $\mathcal{M}_{FCM}$ to acquire the final CCM that contains all possible target objects to improve model's performance.

The qualitative results of segmentation are plotted in Figure 6. The initial pseudo masks from MaskCLIP are coarse who contain many false positives (the second column, support images with light blue mask and query image with light red mask). The proposed mask refiner works effectively in reducing the wrongly recognized background (the third column). Even the refined masks are still rough and might omit some target areas, our method can induce the model to focus on exact target and achieve accurate segmentation (the final column).

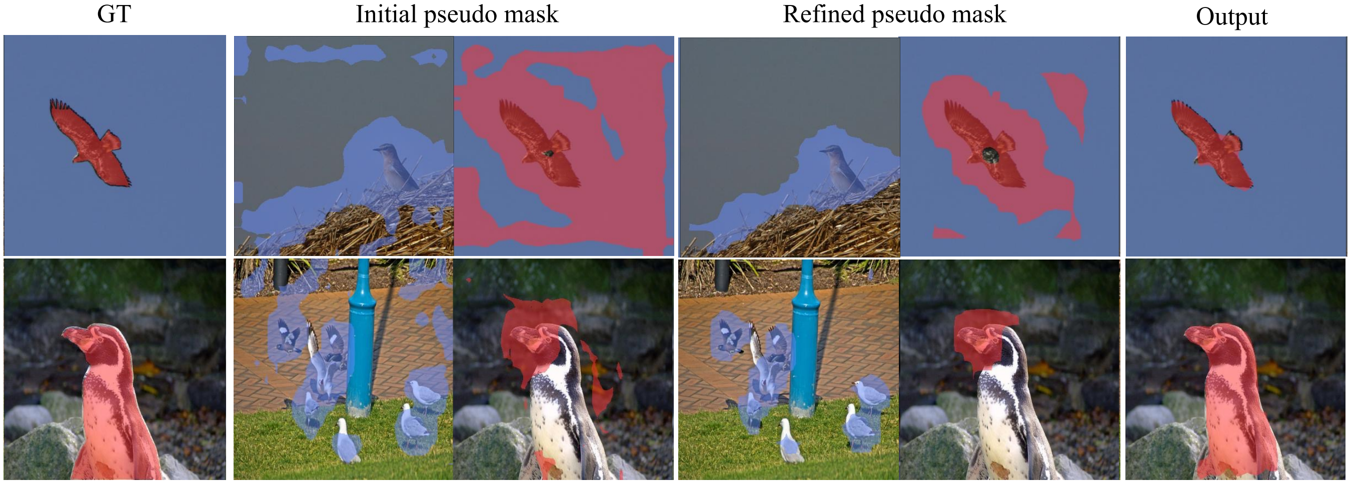


Fig. 6: Qualitative results of initial mask, refined mask and output mask.

5. CONCLUSION

In this work, we have tackled the challenge of languge-guided semantic segmentation by introducing a pretrained VLP model to generate pseudo masks from text labels as full-supervision. To reduce the false positives of pseudo masks and mine pure foreground representation, we propose a mask refine algorithm and a distributed prototype supervision strategy. The complementary correlation matching module learns a comprehensive fine-grained attention map to avoid objects omission. The extensive experiments on two public datasets evaluate the outstanding performance of our method, and the ablation study demonstrates the effectiveness of each component. In the future work, we plan to explore more complex LFSS tasks like general few-shot semantic segmentation by distilling more information from vision-language models.

6. REFERENCES

[1] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, 2022.

[2] Xiaoxu Li, Zhuo Sun, Jing-Hao Xue, and Zhanyu Ma, "A concise review of recent few-shot meta-learning methods," *Neurocomputing*, vol. 456, pp. 463–468, 2021.

[3] Ying Liu, Hengchang Zhang, Weidong Zhang, Guojun Lu, Qi Tian, and Nam Ling, "Few-shot image classification: Current status and research trends," *Electronics*, p. 1752, 2022.

[4] Davis Wertheimer, Luming Tang, and Bharath Hariharan, "Few-shot classification with feature map reconstruction networks," in *CVPR*, 2021, pp. 8012–8021.

[5] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[6] Simone Antonelli, Danilo Avola, Luigi Cinque, Donato Crisostomi, Gian Luca Foresti, Fabio Galasso, Marco Raoul Marini, Alessio Mecca, and Daniele Pannone, "Few-shot object detection: A survey," *ACM Computing Surveys (CSUR)*, pp. 1–37, 2022.

[7] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell, "Few-shot object detection via feature reweighting," in *ICCV*, 2019, pp. 8420–8429.

[8] Gongjie Zhang, Kaiwen Cui, Rongliang Wu, Shijian Lu, and Yonghong Tian, "Pnpdet: Efficient few-shot detection without forgetting via plug-and-play sub-networks," in *WACV*, 2021, pp. 3823–3832.

[9] Shuai Luo, Yujie Li, Pengxiang Gao, Yichuan Wang, and Seiichi Serikawa, "Meta-seg: A survey of meta-learning for image segmentation," *Pattern Recognition*, p. 108586, 2022.

[10] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *ICCV*, 2019, pp. 9197–9206.

[11] Nanqing Dong and Eric P Xing, "Few-shot semantic segmentation with prototype learning.," in *BMVC*, 2018.

[12] 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 *CVPR*, 2019, pp. 5217–5226.

[13] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A. Efros, and Sergey Levine, "Conditional networks for few-shot semantic segmentation," in *ICLR*, 2018.

[14] Hasnain Raza, Mahdyar Ravanbakhsh, Tassilo Klein, and Moin Nabi, "Weakly supervised one shot segmentation," in *ICCVW*, 2019.

[15] Mennatullah Siam, Naren Doraiswamy, Boris N Oreshkin, Hengshuai Yao, and Martin Jagersand, "Weakly supervised few-shot object segmentation using co-attention with visual and semantic embeddings," *arXiv preprint arXiv:2001.09540*, 2020.

[16] Yuan-Hao Lee, Fu-En Yang, and Yu-Chiang Frank Wang, "A pixel-level meta-learner for weakly supervised few-shot semantic segmentation," in *WACV*, 2022, pp. 2170–2180.

[17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.

[18] Haohan Wang, Liang Liu, Wuhao Zhang, Jiangning Zhang, Zhenye Gan, Yabiao Wang, Chengjie Wang, and Haoqian Wang, "Iterative few-shot semantic segmentation from image label text," in *IJCAI*, 2022.

[19] 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 *ICML*, 2021, pp. 8748–8763.

[20] Chong Zhou, Chen Change Loy, and Bo Dai, "Extract free dense labels from clip," in *ECCV*, 2022, pp. 696–712.

[21] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE TPAMI*, 2020.

[22] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *CVPR*, 2022, pp. 8057–8067.

[23] Juhong Min, Dahyun Kang, and Minsu Cho, "Hypercorrelation squeeze for few-shot segmentation," in *ICCV*, 2021, pp. 6941–6952.

[24] Ehtesham Iqbal, Sirojbek Safarov, and Seongdeok Bang, "Msanet: Multi-similarity and attention guidance for boosting few-shot segmentation," *arXiv preprint arXiv:2206.09667*, 2022.

[25] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling, "Expanding language-image pretrained models for general video recognition," in *ECCV*, 2022, pp. 1–18.

[26] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or, "Stylegan-nada: Clip-guided domain adaptation of image generators," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–13, 2022.

[27] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li, "Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning," *Neurocomputing*, vol. 508, pp. 293–304, 2022.

[28] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui, "Open-vocabulary detection via vision and language knowledge distillation," *arXiv preprint arXiv:2104.13921*, 2021.

[29] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai, "Decoupling zero-shot semantic segmentation," in *CVPR*, 2022, pp. 11583–11592.

[30] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu, "A review of generalized zero-shot learning methods," *IEEE TPAMI*, 2022.

[31] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu, "Denseclip: Language-guided dense prediction with context-aware prompting," in *CVPR*, 2022, pp. 18082–18091.

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

[33] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *CVPR*, 2021, pp. 8334–8343.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.