CLIP-SP:Vision-language Model with Adaptive Prompting for Scene Parsing

Jiaao Li[1], **Yixiang Huang**[1], **Ming Wu**[1](✉), **Bin Zhang**[1], **Xu Ji**[1], **and Chuang Zhang**[1]

Abstract We present a novel framework, named CLIP-SP, and design an adaptive prompt method to leverage pre-trained knowledge from CLIP for scene parsing. Our approach addresses the limitations of DenseCLIP, which has shown the superior performance of CLIP pre-trained models over ImageNet pre-trained models in image segmentation, but struggles with the rough pixel-text score maps for complex scene parsing. We argue that owing to containing all textual information on a dataset, the pixel-text score maps, *i.e.*, dense prompts are inevitably mixed with noise. To overcome this challenge, we propose a two-step method. Firstly, we extract visual and language features and perform multi-label classification to identify the most likely categories in the input images. Secondly, based on the top-k categories and confidence scores, our method generates scene tokens which can be treated as adaptive prompts for implicit modeling of scenes, and incorporates them into the visual features to feed the decoder for segmentation. Our method imposes a constraint on prompts and suppresses the probability of irrelevant categories appearing in the scene parsing results. Our method has achieved competitive performance, limited by the available visual-language pre-trained models. Compared with the DenseCLIP, our CLIP-SP achieves a performance improvement on ADE20K, yielding +1.14% mIoU with a ResNet-50 backbone.

Keywords visual-language pre-trained model, scene parsing, adaptive prompt

1 Introduction

Scene parsing is a challenge task in semantic segmentation, which aims to segment and parse an image into different regions associated with semantic categories, *e.g.*, road, person, sky and so on. Since Long *et al*. [3] proposed fully convolutional networks (FCNs) as the pioneer, many efforts have proposed various advanced improvements such as contextual representation aggregation [4–6, 23], multi-scale representation learning [4, 8], and vision transformer architecture designs [7, 9, 10]. The large-scale pre-training models further promote the development of semantic segmentation because of their more robust representation and better modeling of intrinsic relationships.

Fundamental vision-language pre-training models such as CLIP (Contrastive Language-Image Pretraining) [2] capture both fine visual and linguistic features and have shown excellent generalization ability to various downstream vision tasks. However, directly applying CLIP to scene parsing remains a challenge. With the help of prompt support sets composed of the target image, mask and text, CLIPSeg [11] achieves good performance on zero-shot and one-shot segmentation. However the method is hard to extend to complex scene parsing, because it relies on constructing high quality prompt support sets. DenseCLIP [1] roughly utilizes all categories of the dataset to generate prompts, which combines the visual features of image inputs and the linguistic features of all categories. To fit the segmentation task, the final mode of prompts is the mask , *i.e.*, pixel-text score maps. We argue that utilizing all categories to generate prompts inevitably introduces noise, because redundant prompts are useless and misleading due to the equal treatment. This method partially disregards the information provided by the image inputs, thus the guidance from language approximates to be a retrieval of categories on a certain dataset.

Through statistical observation shown in Figure 1, it is found that 99.7% of images contain less than 25 categories for a single image on the validation set of the challenging ADE20K [12] dataset, and the maximum number of categories in one image is 27, which is much less than the total number of 150. To address the aforementioned issue and take advantage of the above observations, we adopt a two-step approach

1 School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: J. Li, lja84@bupt.edu.cn; Y. Huang, huangyixiang@bupt.edu.cn; M. Wu, wuming@bupt.edu.cn; B. Zhang, bluezb@bupt.edu.cn; X. Ji, jixv@bupt.edu.cn; C. Zhang, zhangchuang@bupt.edu.cn.

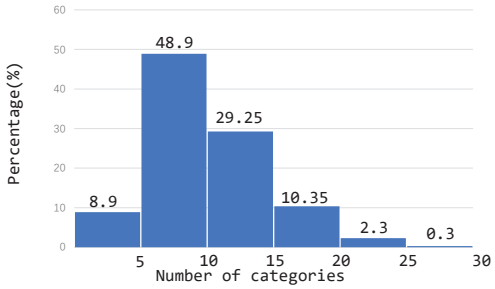

Fig. 1 Statistical data on the number of categories for each image on the ADE20K [12] validation set.

that starts with an additional branch to narrow the selection range of categories for adaptive generation of high quality prompts based on image inputs. Our approach mimics the way humans recognize objects in a scene at a glance, where the concepts present in the scene are instantly identified and then attributed to objects. To adaptively generate image-specific prompts, in the first stage we introduce a lightweight decoder for multi-label classification that takes advantage of multi-modal knowledge of CLIP. Our approach involves a dual graph design, which allows the decoder to effectively incorporate both image and text features. Moreover, we propose a novel selection strategy to boost model performance and accelerate training by utilizing a subset of the ground-truth labels for promoting at the beginning of the training process.

In this paper, we take a step further to explore the applicability of the pre-trained CLIP models for scene parsing. Compared with the state-of-the-art method DenseCLIP, our proposed method exhibits +1.14% mIoU on ADE20K with a ResNet-50 [13] backbone of the CLIP. The main contributions of this work are summarized as follows:

(1) A two-stage framework is proposed, which comprises one path of adaptively generating prompts through multi-label image classification, and the other for prompt-guided semantic segmentation.
(2) A lightweight dual graph decoder is proposed for multi-label image classification, which fully utilizes language knowledge from CLIP serving as the adaptor to prompt according to the image inputs.
(3) A simple but effective selection strategy is proposed to improve the performance of fine-tuning, which transforms uni-modal inputs into multi-modal by using partial ground-truth labels for prompting. Our trick achieves an improvement in performance and is computationally efficient, introducing no additional computational overhead during inference and almost negligible overhead during training.

2 Related Work

2.1 Semantic segmentation

Semantic segmentation has long been a major topic in the vision community and is still a challenging task for parsing diverse contexts in different scenes. In this field there exist extensive studies which can be generally divided into pixel-based methods and region-based methods. The pioneer work of FCNs[3] which treats semantic segmentation as pixel classification adopts fully convolution networks to make dense prediction. A number of later works strive to improve the pixel classifier performance via expanding the receptive field [5], constructing more reliable contextual information [23], and fully utilizing multi-scale features [8]. The region-based methods split semantic segmentation into mask prediction and mask region classification [24, 25]. Our method can be regarded as mainly operating in the neck between the encoder and decoder to enhance features.

2.2 Transferable Representation Learning

Pretraining is the primary impetus to promote the development of computer vision over the years. The universal approaches to solve various downstream vision tasks are based on the ImageNet pretraining that helps to speed up convergence. To get the larger scale data and simultaneously free manual annotation, inspired by the success in NLP, some works focus on masked signal modeling and self-supervised learning [26, 27], which are friendly to dense prediction tasks in the initial design. Another attempts to utilize the supervision directly from natural language to learn visual representation, *e.g.*, CLIP [2], ALIGN [28]. Contrastive learning and large-scale image-text pairs make CLIP successful, which has showed impressive performance of zero-shot transfer on several classification tasks. However, image features are considered to be short of fine local information due to the loose supervision from language, which makes downstream dense prediction confusing. The latest work DenseCLIP [1] demonstrates that the reasonable application of text features from CLIP helps widespread visual models achieve better performance. Despite its great success, we try to explain the main causes of why the CLIP visual encoder is not easy to be fine-tuned to the semantic segmentation task, and propose a solution from another perspective.

2.3 Multi-label classification

The multi-label classification task aims to identify multiple predefined labels in a given image. Existing studies exploit the label correlations to model the semantic relationships between different categories [33–35] and handle the imbalance

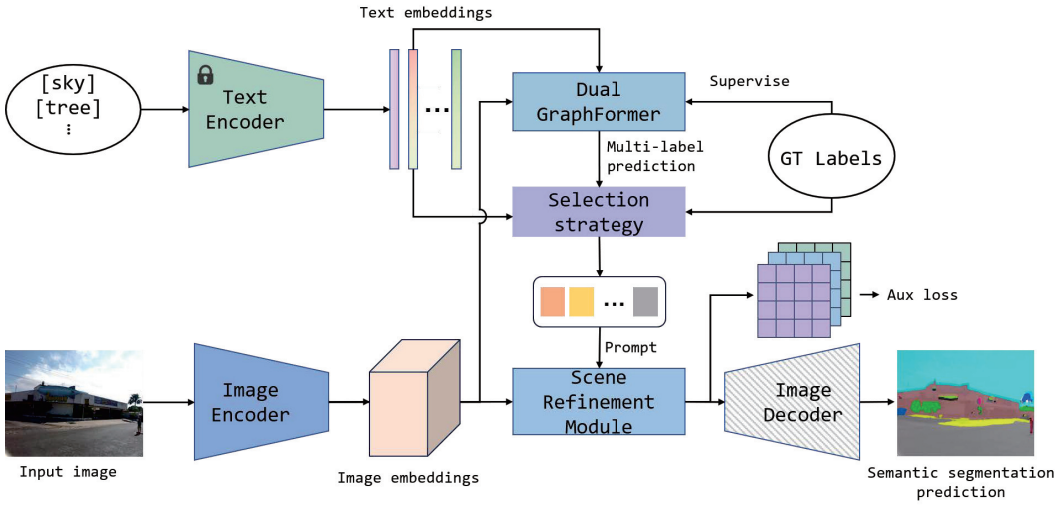

Fig. 2 The overall framework of CLIP-SP. CLIP-SP first extracts the image embeddings and text embeddings of all categories, and then utilize them to obtain multi-label predictions. Trough the selection strategy, we use both the text embeddings and confidence scores of corresponding multi-label predictions to generate scene tokens, *i.e.*, adaptive prompts. In the scene refine module, we combine the image embeddings and scene tokens to obtain refined features for implicit modeling scenes through adaptive prompts.

issue through well-designed loss functions [31, 32]. A recent state-of-the-art method ADDS [17] extends CLIP to zero-shot multi-label classification has inspired us. We propose a dual graph decoder to exploit the language knowledge and compare it to a simple MLP decoder in our framework to verify effectiveness. Even with the simple MLP decoder, the final performance on semantic segmentation is remarkably improved compared with the method without the multi-label classification. Through experiments, we found that the multi-label classification task forces the model to pay enough attention to minor concepts and small objects. It does help the model gain better local information and achieve better performance in semantic segmentation, as shown in RankSeg [36].

3 Method

We begin with a brief introduction of CLIP [2] and our failure case in a naive solution as the preliminary. Then we propose an improved solution, followed by presenting the proposed CLIP-SP in detail.

3.1 Overview of CLIP

CLIP is a visual-language pre-training method that consists of two encoders, including an image encoder $\mathcal{V}(\cdot)$ (ResNet [13] or ViT [9]) and a text encoder $\mathcal{T}(\cdot)$ (Transformer [16]). CLIP aligns the embedding spaces of visual and language during pre-training on 400 million image-text pairs through contrastive learning, where original image-text pairs are regarded as positive samples, while mismatched image-text pairs are negative ones.

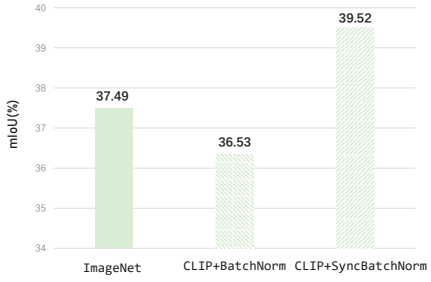

Fig. 3 Results of different pre-training settings on the ADE20K dataset. We report the single-scale mIoU of ResNet50 backbones with different configurations and the same decoder, semantic FPN [8].

Presently, several works [1, 11, 14, 15] have shown that CLIP inherently embedded local image semantics in its features as it learns to associate image content with natural language descriptions during pre-training. However, transferring the pre-trained knowledge of CLIP for the dense downstream task is nontrivial. At the beginning of the simple experiment, we only fine-tuned the image encoder, and the performance of semantic segmentation on the ADE20k dataset is even worse than the same model pretrained on ImageNet as shown in the Figure 3. An interesting discovery is that compared with the same experiment in DenseCLIP, the key point is that in the original model we used the default BatchNorm rather than SyncBatchNorm on 4 RTX 3090 GPUs with a batch size of 16. It shows that there is an obvious internal covariate shift during pretraining on massive image-text pairs, because CLIP adopts a huge minibatch and the BatchNorm layer is sensitive

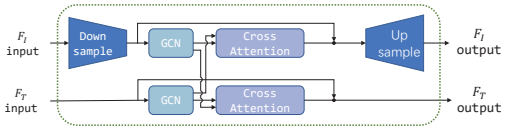

Fig. 4 The overview of our block design of multi-label decoder, named Dual GraphFormer. F_I is the image features and F_T is the text features.

to the batch size during the training.

3.2 Framework

The overall framework of our method is illustrated in Figure 2, which consists of one path for denoising through multi-label image classification and utilizing predictions to generate adaptive prompts, and the other path for semantic segmentation. The actual input of our framework is only an image, because we keep the text encoder frozen and the text embeddings of all categories are unchanged during the training and inference for fixing language knowledge from the pretraining. This procedure can be regarded as constructing a codebook S. Because valuable correlations between text embeddings and image embeddings are constructed during the pretraining on a large-scale dataset, and fine-tuning both $\mathcal{V}(\cdot)$ and $\mathcal{T}(\cdot)$ is more difficult than fine-tuning $\mathcal{V}(\cdot)$ with the guidance of fixed text embeddings. We utilize the template "a photo of a [Label]." proposed by CLIP to construct text embeddings and handle how to adaptively select a subset of them containing less noise. Then we use both selected text embeddings and image embeddings to generate prompts, and combine them with the last stage of feature map output by the image encoder to explicitly incorporate language knowledge and local information. Ultimately we feed the features aggregated with prompts to the semantic segmentation decoder. We explain the formulations and details of both multi-label image classification and the method of prompting as follows.

Dual GraphFormer for Multi-label Decoder. The motivation to complete the multi-label classification task in our framework is to eliminate interference from irrelevant categories in the scene and then reduce the misleading dense prompts. We are inspired by the Dual-Modal Decoder on ADDS [17], which uses both frozen $\mathcal{V}(\cdot)$ and $\mathcal{T}(\cdot)$ and only fine-tunes the decoder to maintain the original alignment between image and text features for zero-shot. To simplify the design and make it more interpretable, we retain the two-way cross-attention module and based on this, we propose our module named Dual GraphFormer. As illustrated in Figure 4, on the one way we use text features as the query and image features as the key and value, and on the other operate in the opposite way. Further, to exploit correlations of both label and the image features of neighborhood, we add graph convolution layers [30] and try to integrate them via cross-attention.

Dual GraphFromer needs two modal inputs. For the image path, input $V \in \mathbb{R}^{C\times(H_4W_4+1)}$ is the concatenated image features $[\bar{\mathbf{z}}, \mathbf{z}]$, where H_4, W_4, C are the height, width and the number of channels of the image features from the 4-th stage of the image encoder, and $\mathbf{z}$ is the local token and $\bar{\mathbf{z}}$ is the global token in the global average pooling of CLIP. For the text path, input $S \in \mathbb{R}^{K\times C}$ is the text embeddings of all categories, where K, C are the number of total categories and channels of the text embedding which is the same with the image embedding for CLIP. At the beginning of block, the compressed matrix $B \in \mathbb{R}^{N\times(H_4W_4+1)}$ is generated by applying a convolutional operation on V, with the goal of reducing the number of nodes from (H_4W_4+1) to N, because there is still redundancy after sampling at 32x downsampling. The transpose of the compressed matrix B is used to recover to the origin number for decompression. $w_v, w_s \in \mathbb{R}^1$ are the learnable factors used to weight the short connection. The normal graph convolution and our implementation are formulated as:

$$H_{out} = \sigma(AH_{in}W), \tag{1}$$

$$H_{out} = \mathrm{GELU}(\mathrm{LayerNorm}(\mathrm{Conv}(H_{in}))), \tag{2}$$

where H_{out} and H_{in} are the output and input features of nodes in the single layer, σ is the nonlinear activation function, A is related to adjacency matrix and W is weight parameter matrix. We replace the origin weight parameter matrix with the LayerNorm layer with elementwise affine to smooth the relation between different samples while preserving the relation between different features like in the transformer. We treat the A as a learnable parameter matrix through one-dimensional convolution with the kernel size of 1. Specially on the text path, we use the co-occurrence matrix of labels on the training set to initialize A with the prior information of statistics. We use 3 blocks to construct the decoder because there is no significant improvement in performance while stacking more than 3 blocks. The logits $S_l \in \mathbb{R}^{K\times 1}$ are obtained by passing the final output $S^{'}$ of the text path, through a fully-connected layer with sigmoid activation function acting as a classifier.

Adaptive Prompting. After obtaining the S_l, we can generate adaptive prompts for modeling scenes in input images. Firstly, we sort the logits S_l and take the top k indices and value of S_l to narrow down the selection range and filter noise. The number of k is related to scene complexity, which can be simply measured by the number of categories in the

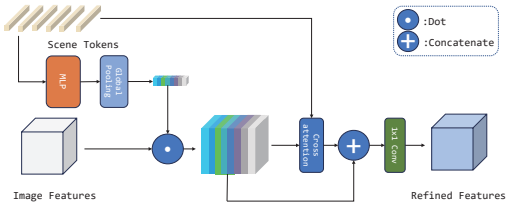

Fig. 5 The overview of our scene refinement module.

single image. We obtain the corresponding text embeddings S_k from the codebook S based on k indices, which can be treated as a rough scene representation. Considering that different scenes have the same categories and the difference lies in the different main body, we concatenate the corresponding probabilities with S_k to enhance the representation ability. After this operation, we ultimately obtain scene tokens $T \in \mathbb{R}^{k\times(C+1)}$, *i.e.*, adaptive prompts.

3.3 Scene Refinement Module

We propose the Scene Refinement Module to fuse the information of adaptive prompts and image features, and then directly feed image features aggregated by scene information to semantic segmentation decoder to predict masks. The detail of our design is illustrated in Figure 5. To make full use of scene tokens T, we further mine the information within it. Because scene tokens contains fine-grained information of scenes and text embeddings are closely related to image features during the visual-language pretraining, we can shrink it to model the channel activation of scenes based on the image features $x_4 \in \mathbb{R}^{C_4\times H_4\times W_4}$ from the 4-th stage of the image encoder $\mathcal{V}(\cdot)$, which is similar to the approach in SENet [37]:

$$T_g = GlobalPooling(MLP(T)), \tag{3}$$

$$x_4^{'} = T_g \cdot x_4, \tag{4}$$

where $T_g \in \mathbb{R}^{C_4}$. The MLP layer is used for modeling channel-associations, and we use the parameterized global pooling layer *i.e.*, the linear project to shrink features. To obtain dense features from adaptive prompts T for aggregating image features, we adopt the non-local approach [38] to calculate cross-attention, and the query is image features $x_4^{'}$, both the key and value are T. Finally, to completely fuse the information carried in prompts into image features, we concatenate the output of the cross-attention and image features $x_4^{'}$ and leverage 1x1 convolution to fuse features.

3.4 Loss Functions

The final loss is the sum of three intermediate losses: one for the multi-label decoder, $\mathcal{L}_{ml}$, one for the segmentation decoder, $\mathcal{L}_{seg}$, and an auxiliary loss $\mathcal{L}_{aux}$ for supervising the scene awareness in the scene refinement module. We use λ_1, and λ_2 to balance the final loss:

$$\mathcal{L} = \lambda_1\mathcal{L}_{ml} + \lambda_2\mathcal{L}_{aux} + \mathcal{L}_{seg}. \tag{5}$$

The multi-label decoder loss function $\mathcal{L}_{ml}$ is the asymmetric loss [32], which can effectively handle long tailed distributions and is also used in the recent multi-label classification work [17]. The segmentation decoder loss function $\mathcal{L}_{seg}$ is the cross-entropy loss, and it is the major loss. The auxiliary loss $\mathcal{L}_{aux}$ can be defined as a cross-entropy loss:

$$\mathcal{L}_{aux} = CrossEntropy(m, \hat{y}), \tag{6}$$

$$m = SoftMax(\frac{QK^T}{\sqrt{d_k}}), \tag{7}$$

$$\hat{y}_{hwi} = \begin{cases} 1, & if \quad l_i = y_{hw}, \\ 0, & otherwise, \end{cases} \tag{8}$$

where m is the attention map in the non-local operation of the scene refinement module, $\hat{y}$ is the one-hot label of m, y_{hw} is the corresponding label in the (h, w) cell and l_i is the label corresponding to the i-th position of top-k results. We treat the attention map as a coarse scene parsing result and the loss can help accurately building connections between prompts and image features.

3.5 Selection Strategy

Although the model can converge normally under our framework, we still worry that in the early stages of training, the poor performance of multi-label decoder makes the prompts almost ineffective, which hinders the model from converging to the optimal solution. The similar types of work, cross-modal adaptation [22] shows that it is a better fine-tuning paradigm for multi-modal models such as CLIP to fine-tune by using cross-modal information as training samples than uni-modal information for downstream uni-modal tasks. It inspires us to introduce the ground-truth labels to alleviate the negative impact of inaccurate top-k results at the beginning of training. We proposed a simple but effective strategy called batch drop. We chose a ratio r to mask the batch of top-k labels from S_l and replace them with labels sampled from the ground-truth. During training, r decreases exponentially, because the performance of multi-label classification is gradually improved and we also need to reduce the reliance on real samples.

4 Experiments

We perform experiments on ADE20K [12], a challenging large-scale semantic segmentation dataset covering an ex-

Table 1 Semantic segmentation results on ADE20K. We compare the performance of CLIP-SP with existing methods when using the same backbone. We report the mIoU of single-scale, the FLOPs and the number of parameters. The FLOPs are measured with 1024 × 1024 input using the fvcore library. The results show that our CLIP-SP outperforms other methods. "$*$" represents our implementation under the same settings.

Backbone	Method	Pre-train	mIoU(%)	GFLOPs	Params (M)
ResNet-50	FCN[3]	ImageNet	36.10	793	50
	EncNet[18]	ImageNet	40.10	566	36
	PSPNet[4]	ImageNet	42.48	716	49
	CCNet[19]	ImageNet	42.08	804	50
	DeeplabV3+[5]	ImageNet	43.95	712	44
	UperNet[20]	ImageNet	42.05	953	67
	DNL[21]	ImageNet	41.87	939	50
	Semantic FPN[8]	ImageNet	37.49	227	31
	Semantic FPN*	CLIP	39.52	249	31
	DenseCLIP + Semantic FPN *[1]	CLIP	43.45	269	50
	CLIP-SP + Semantic FPN$^*(ours)$	CLIP	**44.59**	260	65
ResNet-101	Semantic FPN[8]	ImageNet	40.37	305	50
	Semantic FPN*	CLIP	42.72	327	50
	DenseCLIP + Semantic FPN *[1]	CLIP	45.09	346	68
	CLIP-SP + Semantic FPN$^*(ours)$	CLIP	**46.24**	334	71
ViT-B	Semantic FPN[8]	ImageNet	48.32	1037	101
	Semantic FPN*	CLIP	49.06	1037	101
	DenseCLIP + Semantic FPN *[1]	CLIP	50.58	1043	105
	CLIP-SP + Semantic FPN$^*(ours)$	CLIP	**51.46**	1039	106

tensive range of 150 categories. All models were trained on the 20k training sets, and evaluated on the 2k validation sets. We report on mIoU in the validation sets in accordance with the common practice [19, 20] as well as the FLOPs and the number of parameters for fair comparisons. Our baseline only contains the pre-trained image encoder of the CLIP as the segmentation backbone, and the Semantic FPN [8] as the decoder. The following subsections describe the details of the experiments and results.

4.1 Implementation Details

For a fair comparison, we trained all the models on 4 RTX3090 GPUs with a batch size of 16 for 160k iterations. We use the ResNet-50 pre-trained image encoder of the CLIP for both CLIP-SP and baseline. Our model applies the loss function ASL for multi-label classification. To evaluate the effectiveness of our framework, we use a simple MLP decoder on the path of multi-label classification for comparison, which consists of three linear layers and a fixed classifier, *i.e.*, S. We set the lr multiplier of the image encoder to 0.1 and the initial learning rate to 0.0001 following the schedule of DenseCLIP. We use weighting terms $\lambda_1 = 10$ and $\lambda_2 = 0.4$, to keep the learning balanced.

4.2 Comparison with the state-of-the-art

We compare the proposed method with state-of-the-art algorithms on ADE20K in Table 1. We include the FLOPs, the number of parameters, and the mIoU in single-scale testings. The experiments results show that for the same backbone, our CLIP-SP with a simple Semantic FPN can outperform the state-of-the-art methods and is +1.14%, +1.15%, and +0.88% higher than DenseCLIP on ResNet-50, ResNet-101 and ViT-B backbones with the same input size and reduce the computational overhead. Besides, only a small number of computational cost and parameters have been increased.

4.3 Ablation Studies

To further validate the effects of different components of our CLIP-SP, we perform detailed ablation studies with the ResNet-50 backbone and the results are shown in Table 2. Our baseline model is Semantic FPN with the backbone ResNet-50 on the CLIP pre-train. Firstly, to evaluate the effectiveness of our framework, we chose a weak multi-label decoder, *i.e.*, MLP decoder without the batch drop. The base experiment achieves 3.62% singlescale mIoU improvement, which shows there is a significant improvement while using the language knowledge of CLIP. Secondly, replacing the

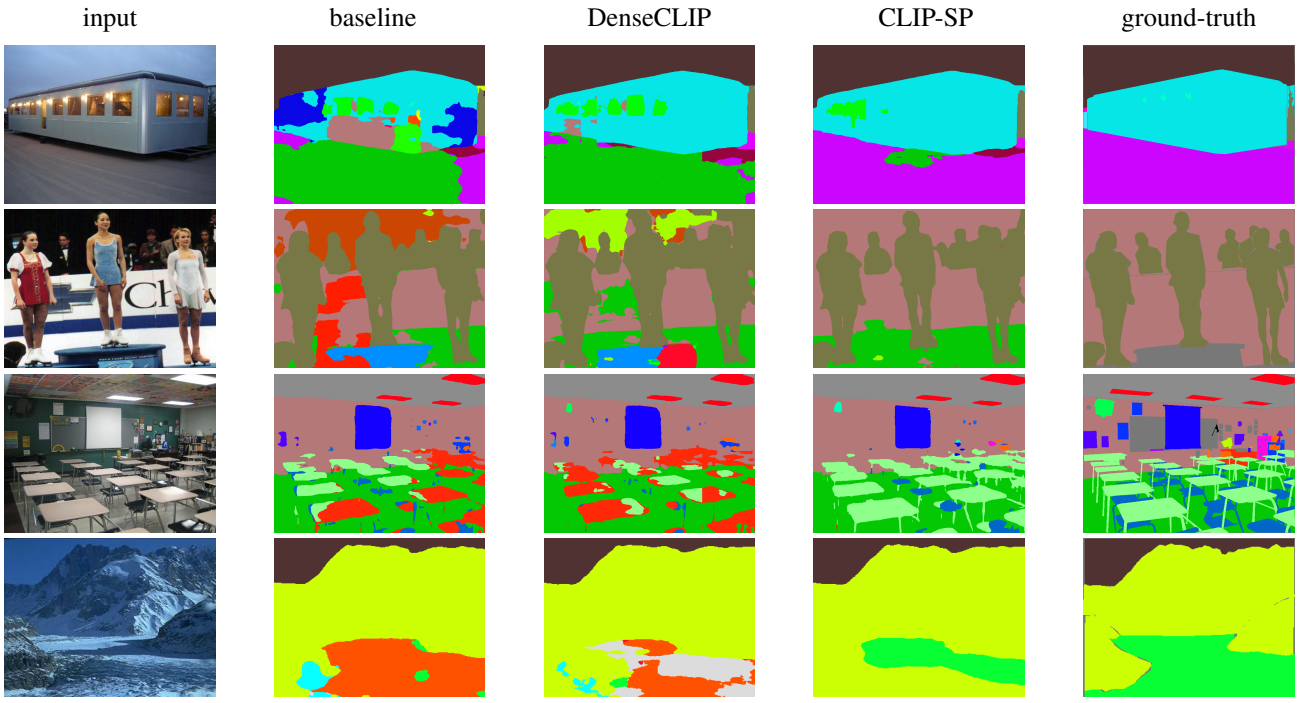

Fig. 6 Qualitative results on ADE20K. We visualize the segmentation results on ADE20K validation set of our CLIP-SP based on ResNet-50, the baseline model and DenseCLIP.

Table 2 Ablation study on the ADE20K. The MLP decoder and Dual GraphFormer decoder are used for multilabel classification.

Method	MLP decoder	Dual GraphFormer	Batch Drop	mIoU(%)
Baseline				39.52
CLIP-SP	✓			43.14
		✓		44.06
		✓	✓	**44.59**

MLP decoder with Dual GraphForm decoder, we obtains the 0.92% improvement. After adding the batch drop, we gain the 0.53% improvements. It shows that the performance may hit a bottleneck and these two modules may not contribute too much to the overall performance.

Table 3 Influence of different number of Dual GraphFormer blocks. Δ is compared with CLIP-SP with the baseline.

num	1	3	5
mIoU (%)	43.25	**44.59**	42.25
Δ	+3.73	**+5.07**	+2.73

Table 4 Influence of the size of the selected label set, *i.e.*, k. Δ is compared with the baseline.

k	20	25	30	35	40	150
mIoU (%)	43.96	44.09	**44.59**	43.49	44.22	43.65
Δ	+4.44	+4.57	**+5.07**	+3.97	+4.70	+4.13

Table 5 Influence of different ratios in batch drop. Δ is compared with CLIP-SP without batch drop.

r_{start}	1.0	0.9	0.8	0.9	0.9
r_{end}	0.0	0.0	0.0	0.1	0.2
mIoU (%)	43.69	44.19	43.38	**44.59**	43.75
Δ	-0.37	+0.13	-0.68	**+0.53**	-0.31

Influence of different number of Dual GraphFormer blocks. We study the influence of the number of Dual GraphFormer blocks, as shown in Table 3. According to the results, our method achieves the best performance on ADE20K when we adopt 3 Dual GraphFormer blocks. Increasing the number of blocks, our method even performs worse.

Influence of different top k. We study the influence of the size of the selected label set, *i.e.*, k, as shown in Table 4. According to the results, our method achieves the best performance on ADE20K when k=30. Excessive k value cannot effectively filter, resulting in decreased performance, especially when k=150.

Influence of different ratios in batch drop. We study the influence of different ratios in Batch Drop, as shown in Table 5. The hyper parameter, exponential decay coefficient is 0.9999 in the experiments. r descends from r_{start} to r_{end} and then remain unchanged. According to the results, we find that at the beginning of training, the over reliance on ground-truth label is harmful to performance. And keeping a small ratio r during the middle to late stage of training may play a role similar to the drop out and obtain a better generalization.

4.4 Visualization

To better demonstrate the superiority of CLIP-SP, we provide several qualitative results in Figure 6. We compare the segmentation maps of our method with the baseline model and DenseCLIP, and find that CLIP-SP is more effective in reducing the probability of irrelevant categories appearing in the scene parsing output.

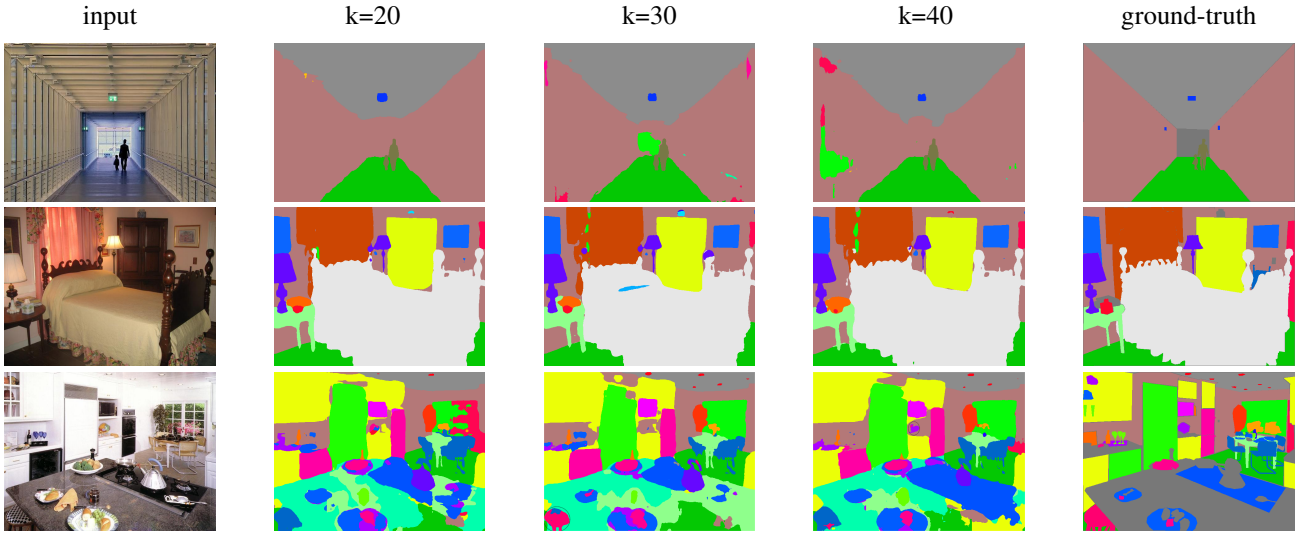

Fig. 7 Qualitative results on ADE20K of different k values. We visualize the segmentation results on ADE20K validation set of our CLIP-SP based on ResNet-50.

Table 6 IoU results for different k values and partial categories of our CLIP-SP based on ResNet-50. The displayed subset of categories are quite representative, as their IoU increases noticeably with the increase of the value of k.

k	IoU (%)									
	counter	**skyscraper**	**blind**	**bar**	**ottoman**	**canopy**	**oven**	**tank**	**tradename**	**lake**
20	24.83	44.24	38.04	18.67	35.62	11.14	33.46	28.03	18.31	39.79
30	27.63	54.36	44.63	22.95	40.77	19.14	40.59	37.64	22.5	53.47
40	33.05	58.22	47.01	49.32	45.91	19.24	56.38	38.79	26.82	57.8

5 Conclusion

In this paper, we have presented a novel framework, CLIP-SP, to reduce the noise in the dense prompt while transferring the language knowledge from the CLIP to scene parsing. The visual features of CLIP contain rich semantics but still need the guidance of local information. We decrease the number of prompts compared with the normal method, and control it within a reasonable range. It shows that more prompts are not better, instead, they will introduce more confusion. Our findings suggest that, by constraining the number of prompts instead of directly constraining classifiers, our method generally results in a lower number of predicted categories compared to other methods.

Limitations & challenges. Although our method has achieved improvement, we find that it is not always beneficial because the tendency leads to ignore objects that is hard to identify, but it is advantageous to segment the main objects in the scene. Also, our design for multi-label classification may not be good enough to make full use of the visual-language pre-trained models. Though we believe the better method of multi-label classification can lead to higher improvements, we need to consider the trade-off between computational cost and accuracy. Besides, owing to our method based on the visual-language pre-trained models, it is nontrivial to expand to other excellent visual pre-trained backbones. We hope our initial attempt can inspire more efforts towards adopting a denoising prompting strategy to exploit the pre-trained vision-language knowledge.

Appendix

We provide more analyses of the influence of different top k in Table 4 and the weighting terms of the loss function in detail.

Details of different top-k value results. According to the characteristics of our method, different k values have different impacts on training. Generally a larger k means more redundancy, and it is not necessarily higher than better in Table 4. Different values of k cause the model to pay different attention to categories during training. Here we find that IoU values for 20% categories of ADE20K increase with the increase of k and show the results of the 10 categories with the most significant changes in Table 6. We see that a low k has a significant negative impact on the performance of the model in certain categories, such as the bar, the lake and the oven. Besides, different k values lead to different performances in different scenes and we provide several qualitative results in Figure 7. We find that the lower k usually performs better

due to containing less unrelated categories in simple scenes. When encountering complex scenes the situation becomes more complex. The lower k performs poorly in terms of details in complex scenes for certain categories, as circled in the last row of Figure 7.

Effects of weighting terms in the loss function. Table 7 shows the effects of λ_1 and λ_2. We find that adjusting the weights to make the three loss scales similar is beneficial to the training results.

Table 7 Influence of different weighting terms.

λ_1	1.0	5.0	10.0	1.0	1.0	10.0
λ_2	1.0	1.0	1.0	0.4	0.6	0.4
mIoU (%)	43.60	43.68	43.85	44.08	43.72	**44.59**

Acknowledgements

We thank the reviewers for their valuable comments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, and J. Lu, "Denseclip: Language-guided dense prediction with context-aware prompting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 082–18 091.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

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

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

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

[6] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 173–190.

[7] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[8] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6399–6408.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[11] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7086–7096.

[12] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, pp. 302–321, 2019.

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

[14] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li *et al.*, "Regionclip: Region-based language-image pretraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 793–16 803.

[15] C. Zhou, C. C. Loy, and B. Dai, "Extract free dense labels from clip," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*. Springer, 2022, pp. 696–712.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] S. Xu, Y. Li, J. Hsiao, C. Ho, and Z. Qi, "A dual modality approach for (zero-shot) multi-label classification," *arXiv preprint arXiv:2208.09562*, 2022.

[18] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

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

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

[21] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and H. Hu,

"Disentangled non-local neural networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*. Springer, 2020, pp. 191–207.

[22] Z. Lin, S. Yu, Z. Kuang, D. Pathak, and D. Ramana, "Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models," *arXiv preprint arXiv:2301.06267*, 2023.

[23] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," *arXiv preprint arXiv:1909.11065*, 2019.

[24] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[25] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "K-net: Towards unified image segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 326–10 338, 2021.

[26] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9653–9663.

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[28] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4904–4916.

[29] F. Lin, Z. Liang, S. Wu, J. He, K. Chen, and S. Tian, "Structtoken: Rethinking semantic segmentation with structural prior," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[31] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classification in long-tailed datasets," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 162–178.

[32] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric loss for multi-label classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 82–91.

[33] H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, and G. Mori, "Learning structured inference neural networks with label relations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2960–2968.

[34] Q. Li, M. Qiao, W. Bian, and D. Tao, "Conditional graphical lasso for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2977–2986.

[35] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 522–531.

[36] H. He, Y. Yuan, X. Yue, and H. Hu, "Rankseg: Adaptive pixel classification with image category ranking for segmentation," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*. Springer, 2022, pp. 682–700.

[37] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

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

Author biography

Jiaao Li Jiaao Li received the B.E. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2021. He is a current master's candidate in Artificial Intelligence at Beijing University of Posts and Telecommunications. His research interests include computer vision, multimodal learning, and semantic segmentation. E-mail: lja84@bupt.edu.cn

Yixiang Huang Yixiang Huang received the B.E. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2020. He is currently pursuing the Ph.D. degree in information and communication engineering with the School of Artificial Intelligence of BUPT. His research interests include computer vision, multimodal learning, and semantic segmentation. E-mail: huangyixiang@bupt.edu.cn

Ming Wu Ming Wu received the M.S. and Ph.D. degrees in information and communication engineering from the Beijing University of Posts and Communications, Beijing, China, in 2003 and 2012, respectively, where she is currently an Associate Professor with the School of Artificial Intelligence. Her primary research interests include computer vision, pattern recognition, and satellite imagery intelligent interpretation.