Prompting classes: Exploring the Power of Prompt Class Learning in Weakly Supervised Semantic Segmentation

Balamurali Murugesan*  
ETS Montreal

Rukhshanda Hussain*  
Jadavpur University

Rajarshi Bhattacharya*  
Jadavpur University

Ismail Ben Ayed  
ETS Montreal

Jose Dolz  
ETS Montreal

Abstract

Recently, CLIP-based approaches have exhibited remarkable performance on generalization and few-shot learning tasks, fueled by the power of contrastive language-vision pre-training. In particular, prompt tuning has emerged as an effective strategy to adapt the pre-trained language-vision models to downstream tasks by employing task-related textual tokens. Motivated by this progress, in this work we question whether other fundamental problems, such as weakly supervised semantic segmentation (WSSS), can benefit from prompt tuning. Our findings reveal two interesting observations that shed light on the impact of prompt tuning on WSSS. First, modifying only the class token of the text prompt results in a greater impact on the Class Activation Map (CAM), compared to arguably more complex strategies that optimize the context. And second, the class token associated with the image ground truth does not necessarily correspond to the category that yields the best CAM. Motivated by these observations, we introduce a novel approach based on a PrOmt cLass LEarning (POLE) strategy. Through extensive experiments we demonstrate that our simple, yet efficient approach achieves SOTA performance in a well-known WSSS benchmark. These results highlight not only the benefits of language-vision models in WSSS but also the potential of prompt learning for this problem. The code is available at code link.

1. Introduction

Image semantic segmentation is a fundamental problem in computer vision, as it serves as a precursor of many tasks, such as medical image analysis or autonomous driving. Fu-
eled by the advances in deep learning, semantic segmentation has experienced a tremendous progress. Nevertheless, obtaining precise pixel-wise annotations is a labor-intensive and time-consuming task.

To alleviate the annotation burden, weakly supervised semantic segmentation (WSSS) has emerged as an appealing alternative, where labels typically come in the form of image tags [15, 21, 28, 31], bounding boxes [26, 55], scribbles [41, 57] or global constraints [25, 46], among others. In particular, image-level WSSS has received significant attention, as it offers a cost-effective alternative to pixel-level annotations (e.g., 20 seconds reported in [5]). Under this setting, WSSS commonly leverages class activation maps (CAMs) [75] obtained from image classification networks to localize objects. Specifically, these maps are later used as pixel-wise pseudo-labels to train a segmentation model, mimicking full supervision. However, CAMs tend to highlight discriminative regions, while ignoring other useful cues, which results in suboptimal pseudo-labels that do not cover the whole extent of the target objects. Narrowing down the existing gap between classification and segmentation tasks is therefore crucial for the progress of WSSS models. To solve this issue, existing approaches intend to complete generated CAMs by forcing the network to focus on more non-discriminative regions, which can be achieved by region mining strategies [21, 30], or integrating iterative processes [3]. Despite employing complex CAM refinement strategies, sometimes involving multiple training steps, existing approaches still exhibit suboptimal performance in terms of both completeness of the target objects and segmentation accuracy.

This motivates the exploration of complementary learning strategies that can further improve the segmentation performance of these models. Vision-language pre-training (VLP) models, such as the recently introduced Contrastive Language-Image Pre-training (CLIP) [47] strategy, have the potential to bring WSSS approaches to the next level, as it can associate much wider visual concepts in an image with their corresponding text labels in an open-world scenario. This contrasts with standard WSSS settings, where the fixed set of predetermined object categories limits the quality of generated CAMs due to unnecessary background activations from class-related background pixels. For example, CLIMS [66] exposed these issues showing that background pixels related to the class ‘railroad’ contributed to the prediction of the CAM associated to the category ‘train’, leading to over-segmented CAMs.

With the rise of these powerful vision-language pre-training models, recent evidence has highlighted the importance of their text input, typically referred to as prompt, in adapting these models to downstream tasks and datasets. For instance, Zhou et al. [77] empirically demonstrated that the use of ‘a [CLS]’ or ‘a photo of a [CLS]’ as a prompt led to substantial differences in the classification performance of the model. Following these findings, recent literature has focused on tuning the context of these prompts, typically as continuous learnable vectors [24, 76, 77]. Despite its potential importance, the impact of modifying the [CLS] token has been largely overlooked in the context of prompt learning. Additionally, while prompt learning has shown promising results in fine-tuning and classification tasks such as zero-shot image recognition, its effectiveness on other visual recognition problems is not well-understood.

Based on these observations, we explore in this work how vision-language pre-training can be further leveraged to improve the performance of WSSS models. In particular, we want to address the following questions: (1) Is prompt learning useful in weakly supervised segmentation?, (2) Which parts of the prompt have a greater impact on the generated CAMs? (3) Can we devise a simple yet effective alternative to improve the segmentation performance under the weakly-supervised learning paradigm?

Our contributions can be summarized as follows:

- We provide empirical evidence that modifying the input prompt in VLP models has a direct impact on the generated CAMs in a weakly supervised segmentation scenario, which in turn affects the performance of the segmentation network.

- More interestingly, our findings reveal that replacing the [CLS] token in the input prompt has a greater impact on the performance than modifying the prompt context, which contrasts with recent observations in classification problems (See Fig. 1). Furthermore, the [CLS] token associated with the actual image ground truth does not necessarily correspond to the category that yields the best CAM, and the performance varies considerably across closely related categories. These insights shed light on the importance of careful prompt design in optimizing the performance of segmentation models trained under the weakly-supervised paradigm.

- Based on these observations, we propose a simple yet efficient strategy to leverage language driven models in the challenging task of WSSS. The resulting model, based on a PrOmpt cLass IEarning (POLE) approach, learns the category name that produces the highest correlation between the image and a corresponding text prompt, and uses it to further leverage the segmentation performance.

- Following the literature, we conduct extensive experiments on PASCAL VOC 2012 to well demonstrate the superiority of our method over other state-of-the-art methods for WSSS.

2. Related Work

Weakly supervised semantic segmentation. Due to its low annotation cost, WSSS based on image-level labels has
gained increasing popularity. These methods rely on class activation maps (CAMs) to identify target object regions by discovering informative pixels for the classification task. As discovered regions are typically highly discriminative and fail to cover the whole context of the target objects, recent literature focuses on generating high-quality CAMs by refining initial estimations from simple models. A common strategy is to mine or erase regions at either image [30, 61, 73] or features level [21, 31], and can be seen as a way of preventing a classifier from focusing exclusively in highly discriminative areas. Other works have instead exploited sub-categories dependencies [6], cross-image semantics [16,56], attention mechanisms [63], equivariant constraints [45, 60] and pairwise semantic affinities [3, 59]. Furthermore, additional supervision, such as saliency maps can be also integrated to provide additional hints about the location of the target object [23, 35]. More recently, visual transformers (ViT) [12] have been also leveraged to improve original CAMs [39, 69], demonstrating superior performance than their CNNs counterparts.

**Contrastive Language-Image Pre-training (CLIP) based semantic segmentation.** Very recently, large-scale VLP models, such as CLIP [47], have demonstrated to improve significantly the performance of vision models on classic recognition tasks, such as zero-shot object detection [18], few-shot learning [22] and zero-shot semantic segmentation [37, 78]. Closely related to our work, CLIMS [66] integrates CLIP in the context of weakly-supervised segmentation, which enhances the initial CAMs by highlighting more comprehensive object regions, while suppressing closely-related background areas. Inspired by the improvement observed in the robustness and generality of visual recognition models driven by language assistance, our work delves deeper into understudied factors, particularly in the weakly-supervised scenario. We stress that our work is different from [66]. In particular, CLIMS [66] proposes to leverage standard CLIP in WSSS, whereas we further explore the effect of the given prompt on this task. As we will show in our empirical validation, properly designing the input prompt results in significant improvements over the standard text prompts. More surprisingly, using the class ground truth as categorical name in the input text prompt does not necessarily yields the best segment results.

**Prompt learning** in visual recognition problems is a rapidly growing research direction, whose popularity stems from the promising results observed in NLP tasks [36, 40, 42, 48]. For example, recent works in prompt learning [13, 49, 58, 76, 77] have achieved promising results on several vision-language tasks, notably in classification. In addition to tackle a different task, i.e., classification vs. weakly supervised segmentation, the main differences with our work is that these approaches mostly study prompt learning from a context perspective. In particular, existing literature considers the class token [CLS] as a fixed word embedding, while optimizing the context [13, 49, 58, 76, 77] or attributes [44]. In most cases, the context tokens are represented by learnable continuous vectors, which yield to text embeddings lacking semantic knowledge [77]. In contrast, our approach performs a selection on a finite set of potential synonyms, which facilitates both the search and the interpretation of the selected token.

### 3. Methodology

#### 3.1. Problem setting.

Let us denote \( D = \{(X_i, y_i)\}_{i=1}^N \) as a weakly labeled dataset, where \( X_i \in \mathbb{R}^{D_i} \) is an input image, \( \Omega_i \) denotes its spatial dimensionality, \( y_i \in \{0, 1\}^K \) its associated one-hot encoded image label, and \( K \) indicates the number of categories. Thus, the goal of weakly supervised semantic segmentation is to provide pixel-wise predictions from an input image \( X_i \) given its corresponding image-level label \( y_i \).

#### 3.2. Our framework.

**Preliminaries: Class Activation Maps.** We first revisit the generation of class activation maps (CAM) from the image-level labels, a popular strategy in WSSS. Let us first define a feature extractor \( f_{\theta}(\cdot), \) which can be represented by a deep neural network parameterized by \( \theta \). Thus, for a given image \( X \), the feature extractor provides a representation \( Z \in \mathbb{R}^{C \times H' \times W'} \), where \( C \) is the number of channels and \( H' \) and \( W' \) represent the dimensionality of the feature map. To provide CAMs, a global average pooling (GAP) layer, followed by a \( 1 \times 1 \) convolutional layer \( W \in \mathbb{R}^{C \times K} \) is applied to the learned features \( Z \) from an image \( X \). Then, the resulting logits are mapped into probabilities \( \hat{y} \in [0,1]^K \) by applying a sigmoid function. To train the neural network, we follow the literature [66, 67, 69] and use the multi-label soft-margin loss as the classification function:

\[
\mathcal{L}(\hat{y}, y) = -\frac{1}{K} \sum_{k=1}^{K} (y^k \log \hat{y}^k + (1 - y^k) \cdot \log (1 - \hat{y}^k)).
\]

Once the backbone network is trained, the initial CAMs can be obtained as follows:

\[
P_k(h, w) = W_k^\top Z(h, w),
\]

where \( P_k \) is the activation map for a given category \( k \).

**Learning objectives.** The framework used for this work shares the same overall structure as the recent CLIMS [66], as it represents the first weakly-supervised segmentation approach integrating CLIP text embeddings. Nevertheless, we stress that in our work we study how we can leverage prompt learning to further improve the performance.

---

1. In PascalVOC, multiple classes can be present in the same image, where \( y \) becomes a multi-class one-hot encoded vector.
of weakly supervised segmentation models. In particular, the standard GAP layer employed to generate CAMs is replaced by a sigmoid function \( \sigma \), resulting in \( P_k(h, w) = \sigma(W_k^T Z(h, w)) \). The pretrained CLIP model [47] uses a visual and a text encoder which we denote as: \( f_v \) and \( f_t \), respectively. Instead of passing the raw input image \( X \) to the CLIP image encoder, it is multiplied by \( P_k \), with the goal of focusing only on the class highlighted by its corresponding CAM. Moreover, to avoid interrogations from related background regions, the input image is also multiplied by \( (1 - P_k) \). Hence, we can create two different visual embeddings: the embedding of the target category \( (v_{ib}^k) \), and its background \( (v_{ib}^k) \), which are formally given by:

\[
v_{ib}^k = f_v(X \cdot P_k), \quad v_{ib}^k = f_v(X \cdot (1 - P_k)). (3)
\]

Now, for all the potential object classes \( k \) and their corresponding text inputs \( t_{io}^k \), we obtain their text embeddings, which are referred to as \( v_{io}^k \):

\[
v_{io}^k = f_t(t_{io}^k). (4)
\]

Note that to generate these embeddings, we just need to provide the different text inputs to the trained CLIP text encoder \( f_t(\cdot) \). Following the reasoning behind the training of CLIP, the foreground image embedding \( v_{io}^k \) should be highly correlated to the text embedding \( v_{io}^k \) of that particular class. In contrast, the background image embedding \( v_{ib}^k \) should have a much lower correlation with the object classes. This can be modeled by using the following objective function:

\[
L_{Cont} = -\alpha \sum_{k=1}^{K} y_k \cdot \log(s_{io}^k) - \beta \sum_{k=1}^{K} y_k \cdot \log(1 - s_{ib}^k), (5)
\]

where \( s_{io}^k = \text{sim}(v_{io}^k, v_{io}^k) \) and \( s_{ib}^k = \text{sim}(v_{ib}^k, v_{io}^k) \) represent the object-to-object and background-to-object similarities between visual and text embeddings, computed as a cosine similarity. Both terms in Eq. 5 act together to ensure that the activation map \( P_k \) covers the maximum possible region of the target, while excluding related background.

### 3.3. Category and image-driven prompt generation

#### Finding potential category related embeddings

To generate the object text representation \( v_{io}^k \) for a class \( k \), the standard input prompt given to the text encoder has the following format: a context token [CTX] followed by the class name token [CLS] and ended by a punctuator (‘.’). While the literature in prompt learning for large-scale visual language pre-trained models focuses on learning the context [CTX] [76], CLIMS [66] uses a fixed prompt, where the [CLS] token corresponds to the categorical label of the image. Contrary to these works, we hypothesize that modifying the input text prompts by optimizing only the [CLS] token has a greater impact on the generated CAMs. Indeed, as we will show empirically in the results section, using the ground truth class as a [CLS] token does not necessarily always results in the best segmentation performance.

Let us suppose that we take an input image \( X \) with its corresponding image class label \( y \), which indicates the \( k \) categories present on the image. For each category \( k \) in \( y \), we obtain a set of similar words, in terms of closeness in the semantic space, using chatGPT [1]. More concretely, we provide the following query as input to chatGPT "Give me \( m \) semantically similar words for [CLS] and also print the cosine similarity scores based on CLIP model", where [CLS] is a class name. This returns a list of \( m \) words along with their similarity scores for that particular [CLS]. This means that for each class [CLS], we can derive a set of \( m \) closest words, denoted as \( S = [CLS_1, CLS_2, ..., CLS_m] \). With this set of related categories, we can create a set of \( m + 1 \) potential text prompts \( T = [t_{io}^0, t_{io}^1, t_{io}^2, ..., t_{io}^m] \).
where $t_{k}^{o}$ is the text prompt containing the categorical ground truth label for class $k$, i.e., [CLS] followed by a fixed [CTX] token, and $t_{k1}, ..., t_{km}$ are composed of the fixed [CTX] followed by a variable [CLS] token chosen from set $S$. Now, we can extract an embedding for each of the prompts in $T$ from the CLIP text encoder, resulting in $V = [v_{1}^{o}, v_{2}^{o}, v_{3}^{o}, ..., v_{m}^{o}]$.

**How to select the best [CLS]?** Given the input image $X$ and its generated class activation map $P_k$ for class $k$, we can obtain an image focusing on discriminative regions for that category by simply doing $M_k = X \cdot P_k$. The resulting image, $M_k$, is given to the CLIP image encoder to get a compressed representation of $X$, i.e., $v_{k}^{o}$. Similar to the steps described in Section 3.2, in contrast, we now obtain a correlation between each of the closest words in set $S$ for the class $k$ and the CAM activated image $M_k$. In particular, this correlation is found by computing a similarity score between the visual and text embeddings: $v_{k}^{io}$ and every $v_{kj}^{io}$ for $j \in \{0, 1, 2, 3, ..., m\}$, given as:

$$\text{sim}(v_{k}^{io}, v_{kj}^{io}) = \frac{v_{k}^{io} \cdot v_{kj}^{io}}{|v_{k}^{io}| \cdot |v_{kj}^{io}|},$$

which generates a vector containing the similarities between the visual encoding and each of the text encodings $[s_{k0}, s_{k1}, ..., s_{km}]$. From this similarity vector, we select the most correlated [CLS] token, which corresponds to the text embedding $v_{k}^{io}$ with the highest similarity with $v_{k}^{io}$, computed with the $\arg\max$ operator.

**3.4. Weakly supervised adaptors**

Following the success of adaptors in pre-trained language-vision models for classification tasks [17,49,74], we propose to further improve our segmentation network by integrating image and text adaptors. In particular, and similar to [17] in classification, we introduce two MLP layers $A_{v}(\cdot)$ and $A_{t}(\cdot)$ to transform the embeddings in the image side and text space, respectively, which is formulated as:

$$v_{k}^{io} = r_{v} \cdot A_{v}(v_{k}^{io}) + (1 - r_{v}) \cdot v_{k}^{io} \quad (7)$$

$$v_{k}^{io} = r_{t} \cdot A_{t}(v_{k}^{io}) + (1 - r_{t}) \cdot v_{k}^{io} \quad (8)$$

with $A_{v}(v_{k}^{io}) = ReLU(v_{k}^{io}W_{1}^{v})W_{2}^{v}$ and $A_{t}(v_{k}^{io}) = ReLU(v_{k}^{io}W_{1}^{t})W_{2}^{t}$, where $W$ represents the learnable parameters of the MLP layers. Furthermore, the parameters $r_{v}$ and $r_{t}$ are learnable vectors of the same shape as the original embeddings, and are used to selectively suppress or amplify the refinement of each feature through the MLP layers. Note that this contrasts with standard adaptors, i.e., [17,49,74], whose balancing weight is a fixed hyperparameter. Our hypothesis is that using a fixed scalar to control the importance of each embedding is suboptimal, as the features refinement process may differ across images as well as depend on the class variation of the dataset.

**4. Experiments**

**4.1. Experimental Settings**

**Dataset and evaluation protocol.** Following CLIMS [66], as well as other recent WSSS works [32], we conduct experiments on the popular PASCAL VOC 2012 [14] benchmark. This dataset contains images with 20 foreground classes, which are split into 1,464 for training, 1,449 for validation and 1,456 for testing. The training set is augmented with 10,582 images and their associated image-level annotations from SBD [19]. To evaluate the performance of the proposed method, we resort to the mean intersection over union (mIoU). Last, while the results reported in the ablation studies are obtained on the training set, the results for the validation and testing sets of PASCAL VOC are obtained from the official evaluation server.

**Implementation Details.** We followed the default settings of CLIMS [66] for training. In particular, input images are randomly rescaled and then augmented by random cropping to $512 \times 512$, as well as by horizontal flipping. We use SGD as the default optimizer, with a cosine annealing policy for scheduling the learning rate, and a batch size of 16 images. The model is trained for 10 epochs, with an initial learning rate of 0.00025 and a weight decay of 0.0001. We follow [4] to adopt ResNet-50 [20] as backbone network for the generation of initial CAMs. All models are implemented in PyTorch and trained on NVIDIA A100 GPU with 40 GB memory. Furthermore, as the initial CAMs coarsely covers the target object, we further perform a refinement step with IRNet [2], to improve their quality before using them as pseudo ground-truth masks, a common practice in WSSS.

**4.2. Results**

**How effective is prompt learning for weakly supervised segmentation?** In this section we assess whether modifying the input text prompt leads to performance differences, which corresponds to our first question. To do this, we first manually selected two popular prompts: “A photo of [CLS]”, and “An image of [CLS]”, which are employed over all the images of the whole dataset. Note that the original CLIMS [66] used the first prompt, and it is therefore considered as the baseline model. Furthermore, inspired by the recent advances on prompt learning, we also evaluate the impact different strategies, which model the different tokens with learnable continuous vectors: CoOp [77] ([CTX] tokens), target optimization baseline in [58] ([CLS] tokens) and a modified version of DeFo [58]. These results, which are reported in Table 1, reveal that the text input, i.e., prompt, of the pre-trained vision-language model plays an important role on the segmentation performance. Indeed, we can observe that depending on the prompt employed, performance differences may diverge up to 3%, particularly on the final generated CAM (last column).

**Context vs. category in prompt learning.** Once we have
Table 1. Does prompt learning improve the performance of weakly supervised segmentation? Comparison of the quality of initial CAMs and refined pseudo ground-truth masks obtained by different prompt learning strategies (with R50 as backbone), where either [CTX] or [CLS] tokens are modified. Evaluation is reported on the train set of PASCAL VOC2012, and refinement of the pseudo-masks is performed using RW (IRN [2]). [CTX] and [CLS] are used to indicate which part of the prompt is optimized in each approach. Furthermore, in the approaches optimizing the [CLS] token, ‘V’ indicates a continuous learnable vector, whereas CLS* represents the class selected among a set of potential classes.

<table>
<thead>
<tr>
<th>Method</th>
<th>[CTX]</th>
<th>[CLS]</th>
<th>Prompt</th>
<th>CAMs +RW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual [66]</td>
<td>✓</td>
<td>✗</td>
<td>“A photo of [CLS].”</td>
<td>56.6</td>
</tr>
<tr>
<td>Manual</td>
<td>✓</td>
<td>✗</td>
<td>“An image of [CLS].”</td>
<td>56.5</td>
</tr>
<tr>
<td>CoOp [76]</td>
<td>✓</td>
<td>✗</td>
<td>[V1</td>
<td>V2</td>
</tr>
<tr>
<td>DeFo [58]</td>
<td>✓</td>
<td>✗</td>
<td>[V1</td>
<td>V2</td>
</tr>
<tr>
<td>Target optimization</td>
<td>✓</td>
<td>✗</td>
<td>“A photo of [CLS]..”</td>
<td>56.8</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✗</td>
<td>“A photo of [CLS]..”</td>
<td>58.7</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the quality of initial CAMs and refined pseudo ground-truth masks using RW (PSA [4]) on PASCAL VOC2012. The mIoU values here are reported on the train set. Backbone denotes the backbone network to generate CAMs. Best approaches (CAM and refined CAM) highlighted in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>CAMs +RW</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSA_CVPR'2018 [3]</td>
<td>WR38</td>
<td>48.0 61.0</td>
</tr>
<tr>
<td>SC-CAM_CVPR'2020 [6]</td>
<td>WR38</td>
<td>50.9 63.4</td>
</tr>
<tr>
<td>SEAM_CVPR'2020 [60]</td>
<td>WR38</td>
<td>55.4 63.6</td>
</tr>
<tr>
<td>AdvCAM_CVPR'2021 [32]</td>
<td>R50</td>
<td>55.6 68.0</td>
</tr>
<tr>
<td>MCTFormer_CVPR'2022 [69]</td>
<td>ViT</td>
<td>61.7 69.1</td>
</tr>
<tr>
<td>SIPE_CVPR'2022 [9]</td>
<td>R50</td>
<td>58.6 64.7</td>
</tr>
<tr>
<td>RECAM_CVPR'2020 [10]</td>
<td>R50</td>
<td>54.8 70.5</td>
</tr>
<tr>
<td>AdvCAM+W-OoD_CVPR'2022 [33]</td>
<td>R50</td>
<td>59.1 72.1</td>
</tr>
<tr>
<td>AFA_CVPR'2022 [53]</td>
<td>MiT</td>
<td>68.7</td>
</tr>
<tr>
<td>CLIMS_CVPR'2022 [66]</td>
<td>R50</td>
<td>56.6 70.5</td>
</tr>
<tr>
<td>VWL_ICCV'2022 [52]</td>
<td>R101</td>
<td>57.3</td>
</tr>
<tr>
<td>AEFT_ECCV'2022 [71]</td>
<td>WR38</td>
<td>56.0</td>
</tr>
<tr>
<td>ViT-PCM_ECCV'2022 [50]</td>
<td>ViT-B16</td>
<td>– 71.4</td>
</tr>
<tr>
<td>ESOL_newECCV'2022 [38]</td>
<td>R50</td>
<td>53.6</td>
</tr>
<tr>
<td><strong>POLE (Ours)</strong></td>
<td>R50</td>
<td>59.0</td>
</tr>
</tbody>
</table>

observed empirically that modifying input prompts results in performance differences, one question that naturally arises is which component of the prompt must be changed. Table 1 shows that replacing a standard [CTX] token (i.e., ‘A photo of’) by a similar sequence (i.e., ‘An image of’) brings 0.5% difference. Nevertheless, if the [CTX] token is optimized for the whole dataset, e.g., CoOp [76], these differences are further increased, with similar results if [CLS] token is optimized as a continuous vector (e.g., DeFo [58] and Target Optimization). Last, we can observe that only optimizing the [CLS] prompt, based on a set of pre-defined closely-related categories, actually provides the best performance across all the methods. These findings align with recent observations in Natural Language Inference [43], which suggest that hand-crafted prompts conveying meaningful instructions outperform automatically optimized prompts.

Comparison to state-of-the-art. Previous experiments empirically demonstrated that modifying the input prompt can significantly improve the performance of weakly supervised segmentation models. These observations motivated the proposed approach, which we benchmark against state-of-the-art models to show its superiority. Note that in what follows, the proposed approach, POLE, is composed of the [CLS] token selection process and the adapter with learnable weights (eq. 7). First, Table 2 reports the results of state-of-the-art methods in the generation of pseudo-masks. We can observe that, even compared to very recent models, the proposed approach brings substantial improvements, ranging from 2 to 6%. We interpret that the performance gain observed comes from the highest correlation between the selected category name and the content of the CAM-activated region in the image., which may capture larger discriminant areas of semantic objects. More interestingly, the proposed approach also outperforms recent methods that use additional information, for example in the form of extra saliency maps, which are typically trained on a supervised foreground-background detection dataset. Thus, based on these results we can argue that our approach represents an effective alternative to generate initial CAMs.

While the quality of the initial CAMs was evaluated in the previous section, we now assess their impact on the semantic segmentation task. In particular, Table 3 reports the segmentation performance of the proposed approach compare to state-of-the-art methods in the validation and testing sets of PASCAL VOC2012 dataset. Compared to approaches that resort to the same supervision, our method provides very satisfactory results, ranking second if all the approaches are considered. Nevertheless, it has been found recently that vision transformers provide much better quality CAMs than conventional convolutional neural networks. Thus, if we just consider the approaches that leverage CNNs for the CAM generation, our approach achieves the best performance, with improvement gains ranging from 0.7% to 3% compared to very recent methods (e.g., RECAM, SIPE or CLIMS [66]). Furthermore, even benchmarking POLE against recent approaches that use additional supervision, e.g., saliency maps, it yields very competitive performance.

How many synonyms are sufficient? And from which Corpus? Previous results have demonstrated empirically that the proposed approach brings substantial improvements by just replacing the categorical name on the ground truth by a closely related synonym. Thus, we now want to evaluate the impact of the corpus selected, as well as the amount of synonyms needed to achieve satisfactory results. In particular, we evaluate the performance of the proposed approach (Fig 3), without adapters, i.e., ‘A photo of [CLS*]’, when the optimal [CLS] token is selected from a set of potential synonyms extracted from different corpus: British National Corpus [11], Google News [29] and English Wikipedia. The first observation is that, while the use of different cor-
Further exploration on the choice of the synonyms across corpus can be found in Supplemental Material.

### Table 3. Evaluation results on PASCAL VOC2012 val and test sets.

The best results are in **bold**. Sup. denotes the weak supervision type. \( F \) denotes full supervision. \( S \) denotes saliency map supervision. \( T \) denotes image-level supervision. Seg. denotes the segmentation network. Bac. denotes the backbone network for CAMs generation. V1: DeepLabV1. V2: DeepLabV2. V16: VGG-16 [58]. R50: ResNet-50 [20]. WR38: WideResNet38 [64].

<table>
<thead>
<tr>
<th>Sup. Method</th>
<th>Seg.</th>
<th>Bac.</th>
<th>val</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>DeepLabV2 2x ResNet101 [7]</td>
<td>-</td>
<td>-</td>
<td>77.6</td>
</tr>
<tr>
<td></td>
<td>WideResNet38 2x ResNet101 [65]</td>
<td>-</td>
<td>-</td>
<td>80.8</td>
</tr>
<tr>
<td>T + S</td>
<td>NSROM_CVP_22 [70]</td>
<td>V2_R101</td>
<td>V16</td>
<td>68.3</td>
</tr>
<tr>
<td></td>
<td>DRS_CVP_21 [27]</td>
<td>V2</td>
<td>V16</td>
<td>70.4</td>
</tr>
<tr>
<td></td>
<td>EPS_CVP_21 [35]</td>
<td>V2_R101</td>
<td>WR38</td>
<td>70.9</td>
</tr>
<tr>
<td></td>
<td>EDAM_CVP_21 [63]</td>
<td>V2_R101</td>
<td>WR38</td>
<td>70.9</td>
</tr>
<tr>
<td></td>
<td>AuxSegNet_CVP_21 [68]</td>
<td>WR38</td>
<td>-</td>
<td>69.0</td>
</tr>
<tr>
<td></td>
<td>SANCE_CVP_22 [68]</td>
<td>V2_R101</td>
<td>R50</td>
<td>72.0</td>
</tr>
<tr>
<td>T</td>
<td>SEAM_CVP_20 [69]</td>
<td>V3_R38</td>
<td>WR38</td>
<td>64.5</td>
</tr>
<tr>
<td></td>
<td>BES_ECC_20 [8]</td>
<td>V2_R101</td>
<td>R50</td>
<td>65.7</td>
</tr>
<tr>
<td></td>
<td>SC-CAM_CVP_20 [6]</td>
<td>V2_R101</td>
<td>WR38</td>
<td>66.1</td>
</tr>
<tr>
<td></td>
<td>A^2GNN_CVP_21 [72]</td>
<td>V2_R101</td>
<td>WR38</td>
<td>66.8</td>
</tr>
<tr>
<td></td>
<td>VWE_ECC_21 [51]</td>
<td>V2_R101</td>
<td>R50</td>
<td>67.2</td>
</tr>
<tr>
<td></td>
<td>AdvCAM_CVP_21 [12]</td>
<td>V2_R101</td>
<td>R50</td>
<td>68.1</td>
</tr>
<tr>
<td></td>
<td>VWS_ECC_21 [52]</td>
<td>V2_R101</td>
<td>R101</td>
<td>70.6</td>
</tr>
<tr>
<td>I</td>
<td>SIPE_CVP_22 [9]</td>
<td>V2_R38</td>
<td>R50</td>
<td>68.2</td>
</tr>
<tr>
<td></td>
<td>CLIMS_CVP_22 [66]</td>
<td>V2_R101</td>
<td>R50</td>
<td>70.4</td>
</tr>
<tr>
<td></td>
<td>AdvCAM_W_OCC_22 [33]</td>
<td>V2_R101</td>
<td>R50</td>
<td>69.8</td>
</tr>
<tr>
<td></td>
<td>SIPE_CVP_22 [9]</td>
<td>V2_R101</td>
<td>R50</td>
<td>68.8</td>
</tr>
<tr>
<td></td>
<td>RECAM_CVP_22 [10]</td>
<td>V2_R101</td>
<td>R50</td>
<td>68.5</td>
</tr>
<tr>
<td></td>
<td>AMIN_CVP_22 [34]</td>
<td>V2_R101</td>
<td>R50</td>
<td>70.7</td>
</tr>
<tr>
<td></td>
<td>Spatial_BCE_ECC_22 [62]</td>
<td>V2_R101</td>
<td>R38</td>
<td>68.5</td>
</tr>
<tr>
<td></td>
<td>ESOL_ECC_22 [38]</td>
<td>V2_R101</td>
<td>R50</td>
<td>69.9</td>
</tr>
<tr>
<td></td>
<td>POL_ECC_22 [60]</td>
<td>V2_R101</td>
<td>R50</td>
<td><strong>71.5</strong></td>
</tr>
<tr>
<td>I</td>
<td>APA_CVP_22 [53]</td>
<td>MTC_B1</td>
<td>-</td>
<td>66.0</td>
</tr>
<tr>
<td></td>
<td>MCT_ECC_22 [69]</td>
<td>V1_R38</td>
<td>DeiT-S</td>
<td><strong>71.9</strong></td>
</tr>
<tr>
<td></td>
<td>ViT-PCM_ECC_22 [50]</td>
<td>V2_R101</td>
<td>ViT-B/16</td>
<td>70.3</td>
</tr>
</tbody>
</table>

### Figure 3. Impact of the Corpus choice and number of synonyms selected.

ChatGPT offers the richer variety of synonyms, yielding the best results across other corpus. Furthermore, increasing the number of synonyms (from 2 up to 4) further improves the results. Note that the number of synonyms includes the categorical name from the ground truth and the requested close synonyms.

### Figure 4. What does CLIP think about the best [CLS]?

*Is the ground truth category chosen everytime? How likely is it that CLIP will select something different?* The plot summarises the percentage of cases where the ground truth category was chosen for an instance of that class. Thus, an inward point on the radial plot indicates that the number of instances where the ground truth category was chosen as the best [CLS] token is considerably low.

**On the impact of the different components.** We observed in Table 1 that the proposed yet simple strategy to optimize the category name achieves better performance than arguably more complex techniques that attempt to optimize the whole text prompt. In this section, we empirically motivate the use of the proposed adapters, as well as the choice of adding learnable parameters to control the importance of each term in Eq. 7, unlike the fixed hyperparameter used in the existing literature. In particular, Table 4 reports these results, where the first observation is that adding the proposed adaptors results in slight improvement compared to...
Figure 5. Qualitative results of the initial class activation maps. Green dotted lines ellipses are used to indicate missed regions by previous approaches (original CAMs and CLIMS [66]) compared to the proposed method. No refinement on the obtained CAMs is done (e.g., RW) to better illustrate the impact of our approach.

Table 4. Ablation on the main components. Empirical results that validate the different components of the proposed methodology. A) Image label as [CLS], i.e., CLIMS [66]; B) Optimal [CLS] selected; C) Fixed adaptor; D) Learnable adaptor. Results are performed on the train set of PASCAL VOC2012.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>74.2 (+3.7)</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>73.8 (+3.3)</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>73.6 (+3.1)</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>70.5</td>
</tr>
</tbody>
</table>

Qualitative results of the obtained CAMs, compared to standard CAM generation and the related CLIMS [66] are depicted in Figure 5. We can observe that despite CLIMS somehow alleviates the under-segmentation problem in conventional CAMs, it still fails to cover larger target regions (see for example first and second columns). In contrast, POLE typically identifies better larger semantic regions related to the target class, resulting in more complete CAMs compared to related approaches.

5. Conclusions

In this work, we have investigated the potential of prompt tuning, an emerging strategy to adapt large pre-trained language-vision models, in the challenging task of weakly supervised semantic segmentation. Our empirical observations have demonstrated that simply replacing the text-token associated with the category name yields better segmentation performance than more complex prompt learning strategies focusing on optimizing the context, which dominate the literature in adapting models. More interestingly, we have observed that employing the corresponding image-level ground truth does not always lead to the best segmentation performance, and closely-related synonyms can indeed result in further performance gains. In light of these findings, we have introduced a simple yet efficient approach, POLE, that selects the most correlated class for a given image in order to generate a better text prompt. Comprehensive experiments have shown that the proposed approach can generate high quality pseudo-labels for WSSS, and achieve state-of-the-art performance in a popular WSSS benchmark.
References

[1] Information provided by chatgpt, an artificial intelligence language model developed by openai. Accessed on [March 23rd, 2023]. Generated text should be treated as such. 4


[50] Simone Rossetti, Damiano Zappia, Marta Sanzari, Marco Schaefer, and Fiora Pirri. Max pooling with vision transform-


ference on Computer Vision and Pattern Recognition, pages
2921–2929, 2016. 2

[76] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei
Liu. Conditional prompt learning for vision-language mod-
els. In Proceedings of the IEEE/CVF Conference on Com-
puter Vision and Pattern Recognition, pages 16816–16825,
2022. 1, 2, 3, 4, 6, 13

[77] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei
Liu. Learning to prompt for vision-language models. Inter-
national Journal of Computer Vision, 130(9):2337–2348,
2022. 2, 3, 5, 13

[78] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yi-
fan Liu. Zegclip: Towards adapting clip for zero-shot se-
matic segmentation. In Proceedings of the IEEE/CVF Con-
ference on Computer Vision and Pattern Recognition, pages
11175–11185, 2023. 3