A Closer Look at the Explainability of Contrastive Language-Image Pre-training

Yi Li[a], Hualiang Wang[a], Yiqun Duan[b], Jiheng Zhang[c], Xiaomeng Li[a,*]

[a]*Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China*
[b]*School of Computer Science, University of Technology Sydney, Ultimo, NSW, 2007, Australia*
[c]*Department of Industrial Engineering and Decision Analytics, The Hong Kong University of Science and Technology, Hong Kong, China*

Abstract

Contrastive language-image pre-training (CLIP) is a powerful vision-language model that has shown great benefits for various tasks. However, we have identified some issues with its explainability, which undermine its credibility and limit the capacity for related tasks. Specifically, we find that CLIP tends to focus on background regions rather than foregrounds, with noisy activations at irrelevant positions on the visualization results. These phenomena conflict with conventional explainability methods based on the class attention map (CAM), where the raw model can highlight the local foreground regions using global supervision without alignment. To address these problems, we take a closer look at its architecture and features. Based on thorough analyses, we find the raw self-attentions link to inconsistent semantic regions, resulting in the opposite visualization. Besides, the noisy activations are owing to redundant features among categories. Building on these insights, we propose the CLIP Surgery for reliable CAM, a method that allows surgery-like modifications to the inference architecture and features, without further fine-tuning as classical CAM methods. This approach significantly improves the explainability of CLIP, surpassing existing methods by large margins. Besides, it enables multimodal visualization and extends the capacity of raw CLIP on open-vocabulary tasks without extra alignment. The code is available at https://github.com/xmed-lab/CLIP_Surgery.

Keywords: CLIP, Explainability, CAM, Multimodal, Open-vocabulary

*Corresponding author (eexmli@ust.hk).

1. Introduction

Providing explanations for neural networks can enhance their transparency and credibility, which has become a crucial consideration across various fields of application. Among various explainability schemes [1], the class attention map (CAM) series methods [2, 3] explain the model via locating discriminative regions, which are widely used in applications such as semantic segmentation [4, 5], image retrieval [6] and generation [7], etc. Recently, contrastive language-image pre-training (CLIP) [8] has gained significant popularity and has been widely adopted in various downstream tasks such as segmentation [9], generation [7, 10]. Although some methods achieve reasonable visualizations through additional modules and alignments [11, 12], they require further tuning [13] and do not provide a feasible CAM to explain the raw model of CLIP. Therefore, there is a strong need to develop a CLIP-CAM model that can enhance the model's transparency and credibility, enabling its *direct application in various downstream tasks without requiring any fine-tuning and alignment.*

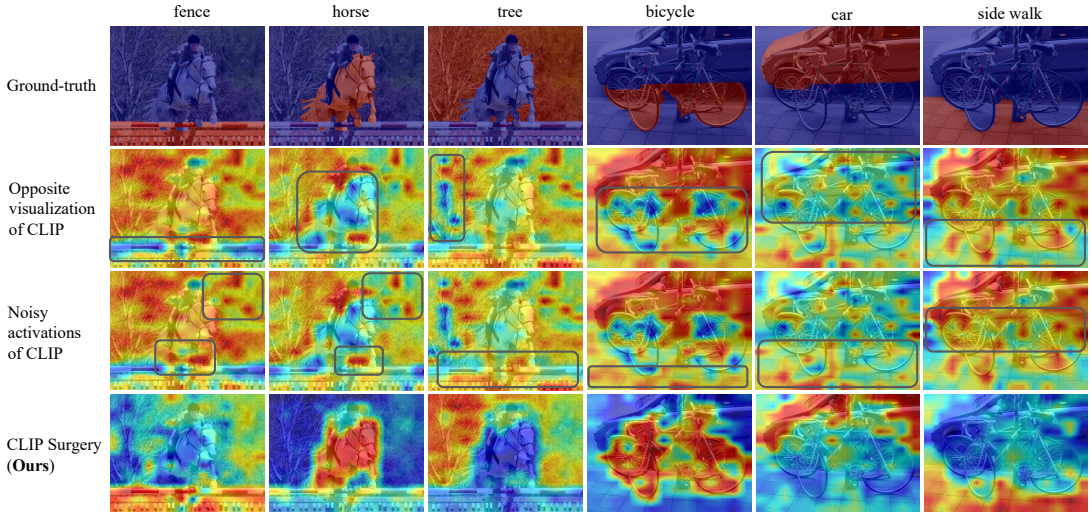


Figure 1: The CAM of CLIP exhibits opposite and noisy visualizations, as indicated by the black boxes in the second and third rows, respectively. In contrast, our CLIP surgery addresses these issues, resulting in improved visualizations, as shown in the last row. The foregrounds are shown in red, while the backgrounds are indicated in blue.

CAM explainability methods were initially designed for convolutional neural networks [2,

3] and vision transformers [14]. However, when directly applied to CLIP, these methods yield unsatisfactory results as shown in Fig. 9, including some recent works for multimodal models [15] and CLIP object localization [16]. This is mainly because the fundamental techniques do not work on CLIP without local alignment [11], rendering subsequent improvement methods ineffective. Specifically, direct application of the basic CAM to CLIP reveals a tendency of CLIP to prioritize background regions over foregrounds, leading to "noisy activations" with "opposite visualizations" as shown in Figure 1. Besides, these phenomena also occur in the basic Grad-CAM [3] as Fig. 9, which is widely used in gradient-based CAM methods [15, 16].

To address the observed explainability issues and generate high-quality CAM, we take a closer look at the architecture and features of CLIP to analyze how these phenomena happen. For the architecture, we observe that the self-attention layers build relations among inconsistent semantic regions (see. Fig. 4), resulting in the opposite visualization. Besides, not all the layers are beneficial and close to the final predictions as Fig. 5. For the features, we find the noisy activations are usually irrelevant to labels, appearing with empty textual input (see Fig. 6), which suggests some features are redundant among categories. Based on these insights, we proposed the CLIP Surgery for reliable CAM, an approach that allows surgery-like modifications to the inference architecture and output features on the raw CLIP, called CLIP architecture surgery and CLIP feature surgery, respectively. To be specific, the architecture surgery is designed to solve the opposite visualization via reforming a consistent self-attention module using original parameters, and aggregate partial beneficial modules via a dual paths structure. Besides, we identify common features across classes as redundant features and mitigate them in the feature surgery to mitigate noisy activations.

Extensive experiments demonstrate the outstanding effectiveness of the proposed CLIP Surgery. Compared with CLIP in terms of explainability, the average mIoU improvements for varied backbones range from 22.11% to 35.95% on multiple datasets as Tab. 1. Notably, the metric mSC (score difference between foregrounds and backgrounds) indicates CLIP prefers background than foreground, while our method solves this issue with average mSC improvements over 47.72%. It also significantly surpasses state-of-the-art CAM methods [15–17] as Tab. 2) (e.g., more than 20% mIoU improvements on VCO 2012 dataset [18], even beyond that using extra alignment [11]. Besides, our method shows wide applicability on open-vocabulary tasks, such as semantic

segmentation, interactive segmentation, multi-label recognition and multimodal visualization.

In summary, this paper has three main contributions:

- We observe that CLIP exhibits opposite visualization and noisy activations. Then, we discover that these phenomena are accompanied by inconsistent self-attention and redundant features among categories, respectively.
- Based on these insights, we propose the CLIP Surgery for reliable CAM, consisting of architecture and feature surgery without fine-tuning.
- The proposed method greatly improves the explainability of CLIP across various backbones and datasets, with wide applicability on multimodal visualization and open-vocabulary tasks.

2. Related Works

2.1. Explainability of CLIP

Recently, CLIP has emerged as a powerful pre-training model supervised by natural language [8]. Before CLIP, traditional explainability methods such as CAM [2], Grad-CAM [3], etc. [19] are designed for convolutional neural networks (CNNs). Recent methods [14] have focused on explainability for vision transformers [20] based on gradient. Besides, Bi-Modal [15] and gScore-CAM [16] were introduced for explainability on multimodal models and CLIP's object localization task, respectively. However, these class attention map (CAM) based methods show unsatisfactory results on CLIP as shown in Table 2. Besides the above CAM-based methods, some recent works generate reasonable visualizations via extra alignments, such as self-supervised mask [11], VQA-based alignment [12], extra bounding box [21] or prompt learning [13]. However, these methods do not explain the original CLIP model since the usage of extra fine-tuning, models, or layers, and they are not as convenient and practical as our CAM-based methods. Besides, the proposed CLIP Surgery is much more effective on CLIP compared with existing CAM-based methods including CAM [2] Grad-CAM [3], LRP [17], Bi-Modal [15], gScoreCAM [16] in Tab. 2, with extensive abilities on downstream tasks. Besides, it surpasses alignment methods ECLIP [11], etc. [12, 21] on practicalness and flexibility. Importantly, we aim to explain the explainability behaviors of the

original CLIP model, while additional models, layers or fine-tuning in alignment methods violate this goal.

2.2. Applications of CLIP

In general, CLIP is used as the pre-training model for downstream tasks such as zero-shot recognition [8], segmentation [9], detection [22], generation [7, 10] etc. These applications are achieved by task-specific designs out of the raw CLIP model. For open-vocabulary semantic segmentation, additional models or supervisions besides texts are used, such as extra fully-supervised proposal models [23]. Besides, additional fine-tuning or training are involved, like token grouping in GroupViT [24] or self-training in MaskCLIP+ [25], and other sophisticated methods [26]. For interactive segmentation like segment anything model (SAM) [27], it performs poorly with text prompts from CLIP alone, and the authors suggest combining text with manual points for better results. For CLIP based multi-label recognition, the alignment between visual and textual features are necessary for studies like Dual Modality [28], DualCoOp [29], TaI-DPT [30].

Unlike these methods built on task-specific alignments, we observe that the raw CLIP without further alignment can achieve comparable performances on some applications with the help of our CLIP Surgery. Specifically, the high-quality CAM from our method can be directly used to generate segmentation results, or generating points from CAM to replace manual points for text-based SAM [27]. Besides, the performances of multi-label recognition are improved when we mitigate the redundant features in the feature surgery, even CAM-based methods [2, 3] are not responsible for classification improvements. Our method is also capable of multimodal visualizations to explain the learning process of CLIP. Notably, our method enhances the transparency of CLIP and enables it to serve multiple tasks simultaneously as shown in Fig. 7, beyond those fine-tuning methods for a certain specific task.

3. Method

In this section, we first introduce the visualization of CLIP from its raw predictions in Sec. 3.1, with descriptions of observed opposite visualization and noisy activation. Then, we analyze the architecture of CLIP in Sec. 3.2, and discover how the opposite visualization happens, with our

solution: consistent self-attention and dual paths. In Sec. 3.3, we observe that the noisy activation is related to redundant features among categories, then propose the feature surgery to mitigate it. The overall framework of the CLIP Surgery is depicted in Fig. 2 before detailed descriptions.

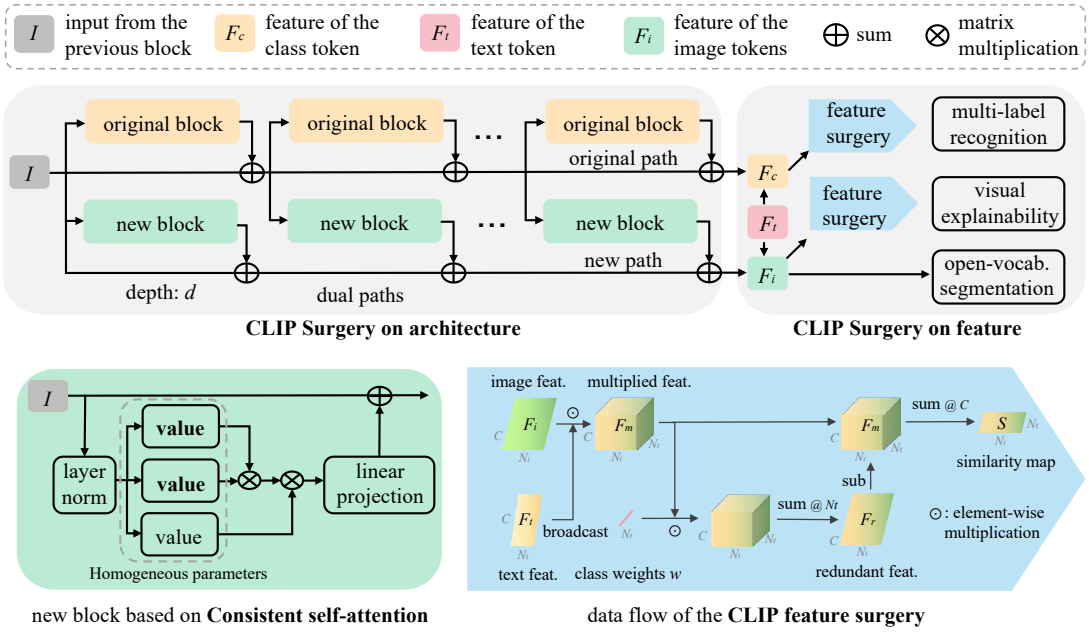


Figure 2: The proposed CLIP Surgery contains two parts as the top part. For the architecture surgery, it is built on a dual paths structure, where the new blocks (left bottom part) are based on the consistent self-attention using homogeneous parameters without feed-forward networks (FFN). The data flow of feature surgery is given in the right bottom part to mitigate redundant features across texts.

3.1. Visual Explainability of CLIP

Foremost, we visually explain the CLIP from the similarity map as the class attention map from its raw predictions. Specifically, the raw predictions are the similarity distances between the text feature and image features of multiple image tokens. This similarity map is the most fundamental and direct explainability cue of CLIP, since it does not require any extra operation like back-propagation in previous gradient-based CAM methods [3, 14]. Here we define the similarity map $\boldsymbol{M} \in \mathbb{R}^{H \times W \times N_t}$ as:

$$\boldsymbol{M} = \mathcal{N}(\mathcal{I}(\mathcal{R}(\boldsymbol{F}_i \boldsymbol{F}_t^{\top}))), \tag{1}$$

where the L2-normalized image feature $\boldsymbol{F}_i \in \mathbb{R}^{N_i \times C}$ and transposed text feature $\boldsymbol{F}_t^\top \in \mathbb{R}^{N_t \times C}$ are multiplied to get the similarity map (N_i, N_t, C are number of image token, text token and channel, respectively). Note this similarity map is processed by functions $\mathcal{R}, \mathcal{I}, \mathcal{N}$ to reshape, interpolate, and min-max normalize to the shape of the raw image ($H \times W$).

Subsequently, we generate similarity maps of CLIP, as shown in Figure 3, and observe that the most prominent problems are the opposite visualization and noisy activations. More specifically, when identifying a target category, CLIP tends to prioritize background regions over foreground regions, which contradicts human perception. Besides, there are many noisy activations at class irrelevant positions. These phenomena are observed across various backbones (ResNets [31] and Vision Transformers [20]), and also occur in multiple explainability methods as shown in Fig. 3 and Fig. 9. These figures indicate these phenomena are common instead of an isolated case.

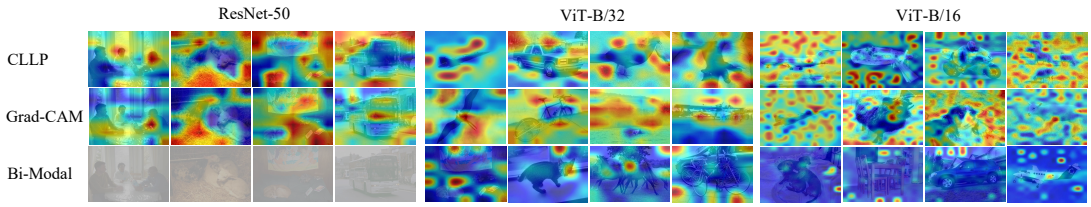


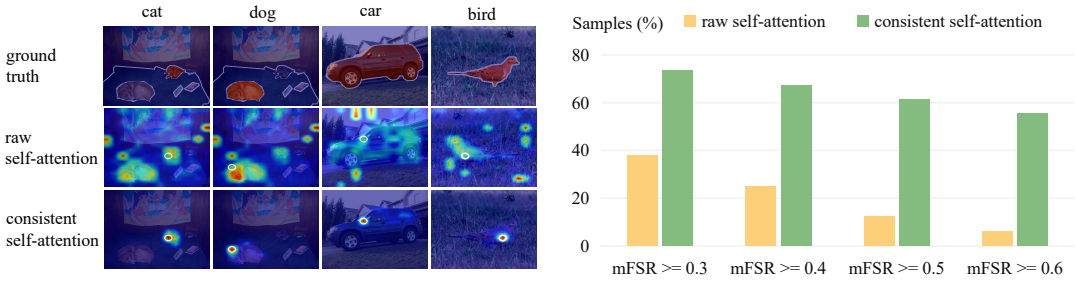
Figure 3: CLIP shows opposite visualization with noisy activations, which are common on varied backbones (ResNet [31], ViT [20]) and methods (Grad-CAM [3], Bi-modal [15]). Note, Bi-modal for ViT is not applicable to ResNet.

3.2. CLIP Architecture Surgery

Consistent self-attention: Firstly, let us draw the conclusion regarding the reason for the opposite visualization: the original self-attention layers in CLIP build relations among inconsistent semantic regions. To substantiate our claim, we present both qualitative and quantitative evidence in Fig. 4. In this figure, raw self-attention $\boldsymbol{A}_{raw}$ uses heterologous parameters ϕ_q, ϕ_k which are different from the parameter ϕ_v for output value as:

$$\boldsymbol{A}_{raw} = \sigma(s \cdot \boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V}, \tag{2}$$

where σ is the softmax function and s indicates the learnable scale, $\boldsymbol{Q} = \phi_q(\boldsymbol{X})$, $\boldsymbol{K} = \phi_k(\boldsymbol{X})$, $\boldsymbol{V} = \phi_v(\boldsymbol{X})$ using learnable linear parameters ϕ_q, ϕ_k, ϕ_v, respectively.



(a) qualitative self-attntion comparison

(b) quantitative self-attntion comparison

Figure 4: The raw self-attention pays more attention on backgrounds, while the proposed consistent self-attention corrects it. (a) Qualitative comparison between the raw self-attention and our consistent self-attention at the token of the highest score on the last layer. The raw self-attention pays much attention to the opposite semantic regions, while our consistent self-attention links nearby tokens at the same semantics. (b) The quantitative comparison for self-attention focuses on foregrounds. Note, the y-axis indicates the sample ratio whose mFSR is higher than a threshold, and mFSR measures the attention degree on the foregrounds as Eq. 4.

The raw self-attention uses heterologous parameters to build global relations. However, these different parameters build relations to inconsistent semantic regions as Fig. 4. This is owing to the lack of local supervision to align the right relation, thus varied parameters may link to context or redundant features. After locating where the problem occurs, we proposed our solution: consistent self-attention $\boldsymbol{A}_{con}$ via homogeneous parameters as:

$$\boldsymbol{A}_{con} = \sigma(s \cdot \boldsymbol{V}\boldsymbol{V}^{\top})\boldsymbol{V}, \tag{3}$$

where the self-attention matrix $\boldsymbol{V}\boldsymbol{V}^{\top}$ uses the homogeneous parameter ϕ_v as that for output feature.

The motivation for consistent self-attention is that the self-attention matrix based on a homogeneous parameter builds relations for consistent semantics. Specifically, there are no heterologous parameters, so the features are the same, where the token itself stands for the highest cosine similarity, and nearby or similar tokens are ranked next. Qualitative results in Fig. 4(a) support this claim. Besides, we introduce a metric mFSR in Eq. 4(b) to quantitatively prove it.

$$mFSR = m_c(m_s(\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{A}_{i,j} \cdot \boldsymbol{G}_{i,j}}{\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{A}_{i,j}})) \tag{4}$$

Specifically, mFSR quantitatively measures the ratio of attention on the foregrounds, where

m_c, m_s count the mean values along classes and samples (every positive label on all images), respectively. Herein, $\boldsymbol{A} \in \mathbb{R}^{H \times W} (N_t = H \times W)$ is the self-attention averaged along the head dimension from the last layer (belonging to the token at the highest score on the similarity map), and $\boldsymbol{G}$ is the foreground binary ground-truth of each sample whose size is $H \times W$. Results in Fig. 4(b) suggest that most examples of the proposed consistent self-attention focus on the foregrounds for varied mFSR thresholds, while a few samples are passed for the raw self-attention.

Dual paths: Besides the above analysis for self-attention, we observe that not all the layers are close to the final predictions and hurt the explainability. To measure the affinity $a(\cdot)$ between the final prediction and that of intermediate layers, we calculate the average cosine similarity (angle) between L2-normalized text feature $\boldsymbol{F}_t \in \mathbb{R}^{C \times N_t}$ and image feature $\hat{\boldsymbol{F}}_c \in \mathbb{R}^C$ at the class token of the targeted module as Eq. 5 (including self-attention modules and feed-forward networks (FFN) summed in the residual). Note that each image feature is multiplied with the last linear projection layer to get $\hat{\boldsymbol{F}}_c$, and the class token is used to extract image-level features for the average affinity $a(\cdot)$ among positive texts.

$$a(\boldsymbol{F}_t, \hat{\boldsymbol{F}}_c) = \frac{\sum_{i=1}^{N_t} \boldsymbol{F}_t^{(i)} \hat{\boldsymbol{F}}_c}{N_t} \tag{5}$$

Based on this metric, we set analysis for image-level intermediate predictions for multiple blocks involved in the residual at varied depths, then we draw the affinity for each module in Fig. 5. From this figure, we find FFN modules have larger gaps than self-attention modules and closed to negatives. Especially, the last FFN returns the feature at cosine 0.1231, which is much farther than the cosine of negative labels. The features of the first three FFNs are very close to the features of negative labels. This finding suggests that FFNs push features towards negatives when identifying positives, thus hurting the model. Tab. 4 experimentally prove this claim in the explainability task. *Therefore, we only take features from partial consistent self-attention modules without FFNs.*

Guided by the above analysis, we aggregate partial self-attention modules for consistent results to the final prediction. To avoid model collapse when deleting FFN and partial layers, a technique called dual paths is proposed. Specifically, we skip FFN in the new path as analyzed in Fig. 5. Then, we define the architecture surgery by the update of dual paths as Eq. 6 and 7. Here, $\hat{x}_{i+1}$ is

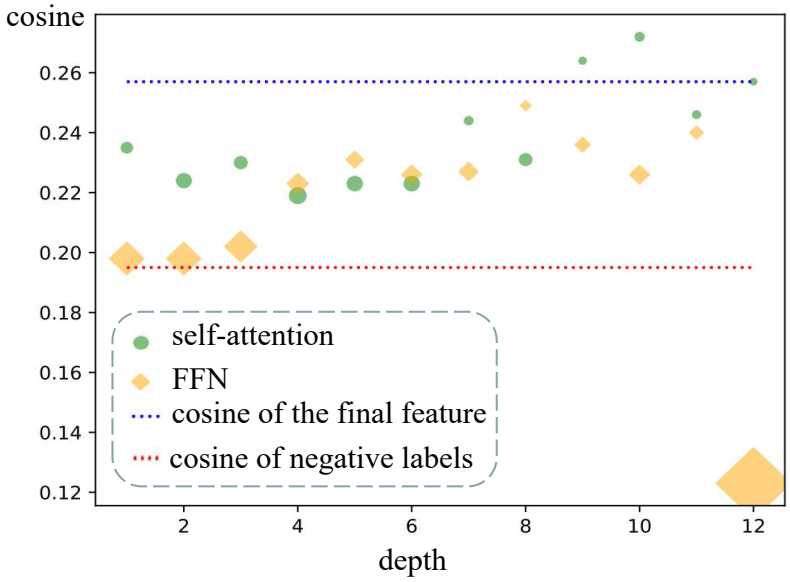


Figure 5: Analysis for intermediate self-attention blocks (green) and feed-forward networks (FFNs, colored in yellow) at different depths via the affinity in Eq. 5. The blue line indicates the mean cosine of all positive labels on the VOC 2012 dataset [18] using backbone ViT-B/16, and the red line is that of negative labels. All scatters are the mean cosine of positive labels and are expected to be close to the blue line. Larger scatters indicate the features of this block are more inconsistent with the final prediction.

the new path:

$$\hat{x}_{i+1} = \begin{cases} None & i < d \\ f_{\boldsymbol{A}_{con}}(x_i, \{\phi_v\}) + x_i & i = d \\ f_{\boldsymbol{A}_{con}}(x_i, \{\phi_v\}) + \hat{x}_i & i > d \end{cases}, \ \forall T\&A. \tag{6}$$

where i is the index of the block, and depth d controls the start of the new path. For shallow layers under d, there is only the original path, and the new path returns "None". For the first reformed self-attention ($i = d$), we merge x_i from the original path with the output of $f_{\boldsymbol{A}_{con}}(x_i, \{\phi_v\})$, which consists of the consistent self-attention in Eq. 3 using parameter ϕ_v and a linear projection layer without Feed-Forward Networks f_{FFN}). For the following modules $i > d$, the x_i is turned to $\hat{x}_i$, where outputs of deeper self-attentions are merged only. This path is only applicable to Transformers and Attention Pooling ($T\&A$) of CLIP. For the original path x_{i+1}, the operations are

not modified:

$$x_{i+1} = \begin{cases} f_{FFN}(x_i') + x_i', & \\ \quad s.t.\ \ x_i' = f_{\boldsymbol{A}_{raw}}(x_i, \{\phi_q, \phi_k, \phi_v\}) + x_i & , \ \forall T\&A \\ f_{res}(x_i) + x_i & , \ \forall Res, \end{cases} \tag{7}$$

where f_{res} is residue blocks and f_{FFN}, ϕ_q, ϕ_k are kept.

3.3. CLIP Feature Surgery

As shown in Fig. 3, the predicted similarity map of CLIP presents many noisy activations in spot shapes at unexpected positions, undermining the credibility of CLIP. We find that noisy activations are caused by redundant features among categories. Because CLIP learns to recognize numerous categories using natural language, leading to only a few features being activated for a specific class, while other features remain non-activated for the remaining classes. Consequently, these non-activated features become redundant and occupy a substantial portion of the feature space, shown as noises. The evidence is given in Fig. 6, where we draw the similarity maps with positive texts and an empty string. For the empty string, all the output features are regarded as redundant features without connection to any category. We can see from this figure that activations of redundant features (empty string) are very similar to noises from positive texts for both visual (top part) and quantitative results (bottom bars), which are powerful pieces of evidence.

We aim to mitigate the problem of noisy activations by removing redundant features. Motivated by the above observation, we aim to count the mean features along the class dimension to identify redundant features. Besides, in our observation, some categories are influenced by the classes at high scores, which leads to false activations. Thus, we give extra emphasis to these obvious classes, when measuring the redundant features.

In terms of formulations, we firstly obtain the multiplied features $\boldsymbol{F}_m \in \mathbb{R}^{N_i \times N_t \times C}$ in Eq. 8, by element-wise multiplication $\odot$ between the L2 normalized features $\boldsymbol{F}_i \in \mathbb{R}^{N_i \times C}$ and the normalized text features $\boldsymbol{F}_t \in \mathbb{R}^{N_t \times C}$ with expand operation $\mathcal{E}$ to broadcast in the same shape.

$$\boldsymbol{F}_m = \mathcal{E}(\boldsymbol{F}_i) \odot \mathcal{E}(\boldsymbol{F}_t) \tag{8}$$

Then, we need the image similarity $\boldsymbol{s}$ of each text to pay more attention on the influential class. In Eq. 9, μ_s indicates the mean value of $\boldsymbol{s}$, and s is obtained from the L2 normalized image feature

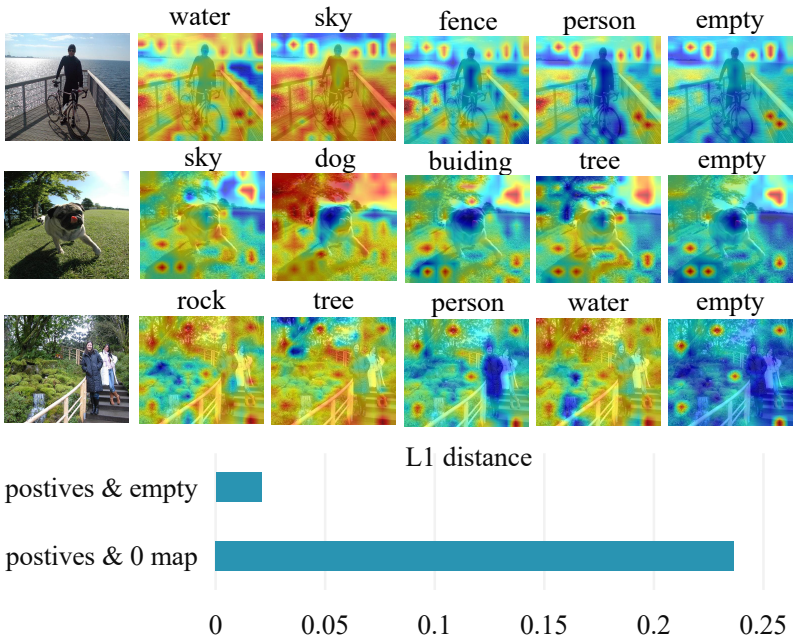


Figure 6: Similarity maps from positive labels are similar to the map from an empty string, which produces redundant features irrelevant to any category. The bar chart indicates the overall L1 distance between positive maps vs. empty map or 0 map (no activation) on the PASCAL Context dataset [32], suggesting the predictions of CLIP among categories are highly related to redundant features.

$\boldsymbol{F}_c$ and transposed text feature $\boldsymbol{F}_t^\top$ with logit scale τ and softmax σ.

$$\begin{aligned} \boldsymbol{w} &= \frac{\boldsymbol{s}}{\mu_s} \\ \boldsymbol{s} &= \sigma(\tau \cdot \boldsymbol{F}_c \boldsymbol{F}_t^\top). \end{aligned} \tag{9}$$

Then, we broadcast the weight w and multiply to $\boldsymbol{F}_m$ in Eq. 8 to count the mean value along the category dimension N_t as the redundant feature $\boldsymbol{F}_r \in \mathbb{R}^{N_i \times C}$ in Eq. 10.

$$\boldsymbol{F}_r = mean(\boldsymbol{F}_m \odot \mathcal{E}(\boldsymbol{w})) \tag{10}$$

lastly, we use multiplied features $\boldsymbol{F}_m$ to subtract the expanded redundant feature $\mathcal{E}(\boldsymbol{F}_r)$ for the removal of redundant features. Then sum all features along the channel dimension C to get the cosine similarity $\boldsymbol{S} \in \mathbb{R}^{N_i \times N_t}$:

$$\boldsymbol{S} = sum(\boldsymbol{F}_m - \mathcal{E}(\boldsymbol{F}_r)) \tag{11}$$

Note that Eq. 11 is specifically designed for the explainability task, and the final similarity map is obtained from $\boldsymbol{M} = \mathcal{N}(\mathcal{I}(\mathcal{R}(\boldsymbol{S})))$ as Eq. 1. For multi-label recognition tasks, we only need to replace $\boldsymbol{F}_i$ in Equation 8 with the image-level features of the class token $\boldsymbol{F}_c \in \mathbb{R}^{1\times C}$ for similarity distance. Note, CLIP Feature Surgery is not suitable to rank base methods, like argmax in segmentation and top-1 accuracy in single-label classification. This is because $\boldsymbol{F}_r$ can be considered as a common bias, which does not affect the ranking of categories within a single image. Instead, it adjusts the scores across images or pixels.

4. Experiments

4.1. Setup

Datasets: The proposed CLIP Surgery is operated on the inference stage without training, thus we don't need any training datasets. To evaluate our method, we use the validation split of each dataset. Specifically, for explainability task, we use PASCAL VOC 2012 [18], MS COCO 2017 [33], PASCAL Context [32], and ImageNet-Segmentation-50 [34], where single-label, multi-label, object and stuff are included for comprehensive evaluation. We also use the same datasets as the explainability task to evaluate the interactive segmentation task. For the open-vocabulary semantic segmentation task, besides PASCAL Context [32], we test our CLIP Surgery on two widely used semantic segmentation datasets: COCO Stuff [35] and CityScapes [36]. For the open-vocabulary multi-label recognition task, we use PASCAL Context [32], and the most used dataset NUS-Wide [37]. Besides, we use image-text pairs in GCC3M dataset [38] for the multimodal visualization task.

Implementation: We implement the CLIP Surgery via the official CLIP. All the parameters are copied from it with modified architecture and feature operations. We elaborate the applicability of each module for varied tasks as Fig. 7. Specifically, semantic segmentation task and single-label recognition tasks don't apply the feature surgery, since their results are obtained from the argmax operation, while feature surgery doesn't change the prediction rank. It is also the reason why single-label recognition keeps the same results as before. Notably, it is not a shortage of our method, because **all the CAM-based methods are not responsible for classification improve-**

ments, including ours. So this paper does not involve single-label classification datasets like ImageNet. Even though our method works in the multi-label recognition task.

Tasks	Architecture Surgery	Feature Surgery	Notes
Explainability	√	√	-
Multimodal Visualization	√	√	-
Interactive Segmentation	√	√	-
Semantic Segmentation	√	✗	Feat. surgery does not change argmax results.
Single-label Recognition	✗	✗	CAM based methods do not improve recognition.
Multi-label Recognition	✗	√	Feat. surgery helps to adjust scores among images.

(a) The applicability of CLIP Surgery on multiple tasks

Applilicability	CAM Methods	Downstream Methods	Ours
Explain Raw CLIP	√	✗	√
No Training	√	✗	√
Good Performance	✗	√	√
Multiple Tasks	✗	✗	√

(b) Applicability comparison between ours and other methods

Figure 7: (a) Applicability of the architecture surgery and feature surgery on varied tasks. (b) Comparison of applicability among our method, conventional CAM methods, and downstream methods.

For the explainability task, we set experiments on the official CLIP with 5 backbones, namely ResNet-50, ResNet-101, ViT-B/32, ViT-B/16, and ViT-L/14. For ResNets [31], CLIP adds an attention pooling with self-attention in the last layer, and their output size is 7×7 under input resolution 224 (all the datasets are resized to 224×224 without crop or other augmentation). For the output size of ViTs [20], it depends on its patch size. Specifically, the output sizes are 7, 14, and 16 for ViT-B/32, ViT-B/16, and ViT-L/14, respectively. As for the hyperparameters of CLIP Surgery, the depth d in Eq. 6 is set to 7 according to analysis in Fig. 5, and softmax scale τ in Eq. 9 is set to 2. Note that these hyperparameters are not sensitive with small variations of results. For example, on COCO 2017 [33] dataset using backbone ViT-B/16, the variation of results at metric mSC is 0.42% for $d \in [1, 10]$, and variation for $\tau \in [1, 10]$ is 0.12%. As to the textual prompt, we deploy the 85 templates in CLIP [8], and combine templates with the names of categories. Then, we mean the text features along the template dimension as the final text features F_t, and this prompt ensemble is applied to all methods for fair comparison.

For open-vocabulary semantic segmentation, we resize the images of PASCAL Context [32] and COCO Stuff [35] to 512×512, and crop each image of Cityscapes to 8 patches at 512×512 from 2048×1024 without overlap. For fair comparisons, all the compared methods use the same backbone ViT-B/16, except grouping methods [24, 39] whose backbones are specially designed. Note, the original ReCo [40] uses ResNet50x16 which is much larger than ViT-B/16,

and the implemented results based on ViT-B/16 are from the official code. Since ViL-Seg [41] is not released, we report its results from the paper. We reproduce the results of MaskCLIP [25] at the same settings as ours, without its post-processing methods, for fair comparison with other works. For open-vocabulary multi-label recognition, we take the prediction from the original path (the same as the original CLIP at input size 224), and apply feature surgery to replace the softmax operation of CLIP. For other CLIP-based zero-shot methods [28, 29], we report their results from the papers. Besides, we implement TaI-DPT [30] from the official code as a baseline, and apply the feature surgery on its predictions to verify the complementarity of our method.

For the interactive segmentation, we convert text to points for the Segment Anything Model (SAM) [27]. It helps to replace the cost of manual labeling and avoids the bad performance of SAM using text prompts only. Specifically, we pick points whose scores are higher than 0.8 from the similarity map, and take the same number of points ranked last as background points. Note that there is only one text prompt for SAM instead of multiple texts. Thus, we implement the feature surgery via the redundant feature $\boldsymbol{F}_{empty}$ from text features of an empty string to replace $\boldsymbol{F}_r$ in Eq. 11. This situation is the same as multimodal visualization, where the whole sentence is used as a text label without other categories. We use the string "[start][end]" (start flag and end flag, respectively) to extract redundant features.

Metrics: We use mean Intersection over Union (**mIoU**) for semantic segmentation, and mean Average Precision (**mAP**) for multi-label recognition. Besides, mIoU is used to evaluate each positive label independently for interactive segmentation and explainability. The new metric for explainability is mean Score Contrast (**mSC**), which refers to the score difference between foregrounds and backgrounds, ranging from -100% to 100%. This metric can reflect the problem of opposite visualization when the value is lower than 0, but mIoU cannot. Besides, mSC measures the difference of scores, while mIoU measures the mask without information of confidence, which provides new insights. We give the formula definition as:

$$mSC = m_c(m_s(m_p(\boldsymbol{M}^s \cdot \boldsymbol{G}) - m_p(\boldsymbol{M}^s \cdot \neg\boldsymbol{G}))) \,, \tag{12}$$

where m_c, m_s, m_p compute the mean values along classes (macro average), samples (the average over all images for a class), and pixels (spatial average), respectively. Note, $\boldsymbol{M}^s$ is the similarity

map for a certain sample, $\boldsymbol{G}$ is the corresponding binary foreground ground-truth, and $\neg\boldsymbol{G}$ indicates the background ground-truth.

4.2. Results of Explainability

Table 1: Results comparison between "CLIP" and "Ours" on four datasets and five backbones. mSC (%) ranges from -100% to 100%, reflecting the score contrast between foreground and background. Note, "Δ" indicates our average improvements over five backbones.

		ImageNet-S50		VOC 2012		PASCAL Context		COCO 2017	
Method	Network	mIoU ↑	mSC ↑	mIoU ↑	mSC ↑	mIoU ↑	mSC ↑	mIoU ↑	mSC ↑
CLIP	ResNet50	28.18	-26.50	17.78	-27.88	16.98	-14.02	10.53	-19.17
Ours	ResNet50	66.05	43.28	53.85	44.60	38.50	32.44	29.24	33.80
CLIP	ResNet101	28.17	-23.30	18.06	-23.48	17.52	-9.90	10.66	-17.31
Ours	ResNet101	65.51	43.22	52.51	44.07	38.03	32.21	29.89	35.50
CLIP	ViT-B/32	28.05	-21.86	17.56	-24.06	16.37	-17.68	10.15	-21.02
Ours	ViT-B/32	59.24	36.72	51.14	41.15	40.10	32.90	29.22	31.35
CLIP	ViT-B/16	27.87	-18.84	17.36	-19.86	15.76	-16.73	9.74	-23.37
Ours	ViT-B/16	62.41	36.50	55.78	41.64	46.28	34.32	35.23	35.43
CLIP	ViT-L/14	27.89	-18.34	17.24	-24.42	15.62	-20.26	9.64	-27.51
Ours	ViT-L/14	61.72	28.25	54.47	34.89	42.71	28.11	37.67	32.54
Δ	-	**34.96**	**59.36**	**35.95**	**65.21**	**24.67**	**47.72**	**22.11**	**55.40**

Effectiveness: Table 1 shows the comparison between the proposed CLIP Surgery and the original CLIP model. The comparison is performed on four datasets using five backbones. Our results consistently outperform the CLIP baselines in each setting, with significant improvements. On average, our method achieves a higher mIoU than CLIP by 22.11% to 35.95%, and a higher mSC by 47.72% to 65.21%. Notably, the mSC of CLIP is lower than 0, indicating that the model tends to favor the background over the foreground, our CLIP Surgery corrects this issue, yielding results far above 0. These results provide strong evidence for the effectiveness of our proposed method.

Besides quantitative results, we draw visualization results for different datasets in Fig. 8. These

results demonstrate that our proposed method effectively addresses the two discussed problems: opposite visualization and noisy activations. Notably, the proposed method enables us to produce clear and interpretable visualizations based solely on the original CLIP, without any additional training or complex back-propagation.

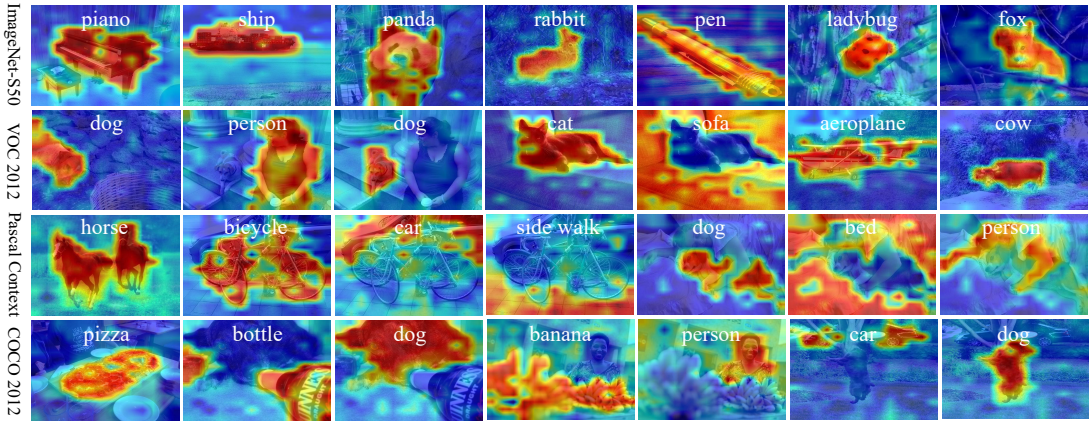


Figure 8: Our CLIP Surgery solves the explainability problems and provides good visualizations on varied datasets.

Compared with Previous Works: We compare CLIP Surgery with previous explainability methods, including the similarity map of original CLIP, Grad-CAM [3] for CNN, pLRP [17] implemented by Chefer et al. [14], Bi-Modal based on ViT, and explainability methods by [11, 16]. We conduct the comparison using the ViT-B/16 and ResNet-50 from the official codebase. Tab. 2 displays the results, where CLIP Surgery achieves the best performance across all datasets, metrics, and backbones, surpassing other methods by significant margins. ECLIP [11] ranks second for ViT-B/16, but our CLIP Surgery outperforms it by a maximum of 15.94% in mIoU and 19.89% in mSC without any training. RCLIP [11] is the top method without extra training, and our results surpass it by 18.42% in mIoU and 23.13% in mSC. Other methods exhibit larger performance gaps compared to ours. Notably, some methods [15, 17] for ViT are not applicable to ResNet, while we also achieve much higher results on ResNet-50 in this table.

Table 2: Results compared with previous state-of-the-art explainability methods. Besides, "CLIP" indicates the similarity map of CLIP. "'-" means this method is not applicable to ResNet, "†" notes the model requires extra fine-tuning, and "⋆" indicates that back-propagations are required for each label at lower efficiency. Note, mIoU (%) measures the mean intersection over union for positive labels, and mSC indicates the mean score contrast in Eq. 12

	ImageNet-S50		VOC 2012		PASCAL Context		COCO 2017	
Method	mIoU ↑	mSC ↑	mIoU ↑	mSC ↑	mIoU ↑	mSC ↑	mIoU ↑	mSC ↑
ResNet-50								
CLIP [8]	28.18	-26.50	17.78	-27.88	16.98	-14.02	10.53	-19.17
Grad-CAM⋆ [3]	34.39	1.30	22.93	1.76	19.85	0.76	13.11	1.89
pLRP⋆ [17]	-	-	-	-	-	-	-	-
Bi-Modal⋆ [15]	-	-	-	-	-	-	-	-
gScoreCAM⋆ [16]	62.21	33.80	48.53	33.69	34.68	23.27	12.98	13.32
RCLIP [11]	54.45	26.50	41.17	27.88	28.48	14.02	21.87	19.17
ECLIP†[11]	62.49	30.49	50.04	31.14	36.28	21.77	27.27	23.43
CLIP Surgery (Ours)	**66.05**	**43.28**	**53.85**	**44.60**	**38.50**	**32.44**	**29.24**	**33.80**
ViT-B/16								
CLIP [8]	27.87	-18.84	17.36	-19.86	15.76	-16.73	9.74	-23.37
Grad-CAM⋆ [3]	28.59	-11.05	17.90	-14.51	16.04	-14.68	9.89	-18.91
pLRP⋆ [17]	46.76	10.33	31.73	8.88	25.61	6.24	21.06	11.22
Bi-Modal⋆ [15]	43.37	6.77	30.64	6.76	24.31	3.95	18.33	7.99
gScoreCAM⋆ [16]	24.75	7.47	11.33	1.59	13.26	0.35	13.59	4.45
RCLIP [11]	48.00	16.14	37.36	18.51	33.25	18.21	26.12	22.41
ECLIP†[11]	58.59	26.32	48.46	28.83	30.34	14.43	24.67	18.95
CLIP Surgery (Ours)	**62.41**	**36.50**	**55.78**	**41.64**	**46.28**	**34.32**	**35.23**	**35.43**

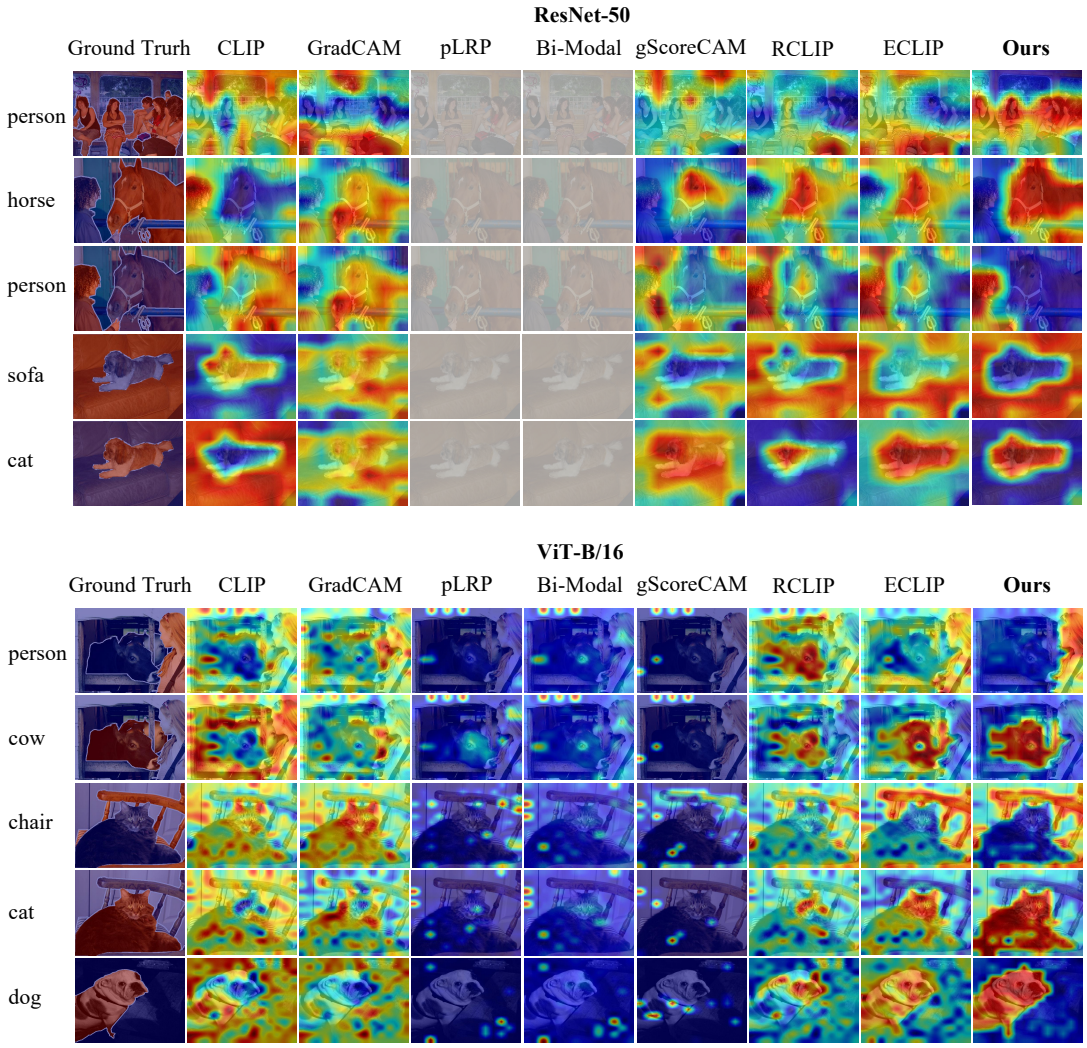


Figure 9: Visual comparison between our CLIP Surgery with state-of-the-art explainability methods on VOC 2012 [18]. Note the foregrounds are colored in red, and the white mask indicates this method is not applicable to this backbone. Our visualization quality is much better than other methods for both ResNet and ViT, without any fine-tuning like ECLIP [11] or back-propagation in GradCAM [3]. Besides, our method is not limited by certain backbones such as pLRP [17], Bi-Modal [15], and gScoreCAM [16].

In addition, we also present a visual comparison in Fig. 9. The results show that our CLIP Surgery provides better quality visualizations compared to existing explainability methods, with-

out problems of opposite visualization. Furthermore, our method produces fewer noisy activations than methods like pLRP [17], Bi-Modal [15], gScoreCAM [16]. Also, our method shows a more obvious score contrast than RCLIP and ECLIP [11] at better visualization quality. All the above improvements enhance the model's visual explainability and make it more credible.

4.3. Ablation Study

Table 3: Ablation study of CLIP Surgery. "no" is the original CLIP, "Archi." indicates the CLIP Architecture Surgery and "Feat." means the CLIP Feature Surgery. Note, mSC lower than 0 suggests scores of background is higher than foreground. The used dataset is the Pascal Context dataset [32] using CLIP with a ViT-B/16 backbone.

	Explainability		Multi-label
CLIP Surgery	mSC (%) ↑	mIoU (%) ↑	mAP (%) ↑
no	-16.73	15.76	47.09
Archi.	31.15	43.47	47.09
Archi. + Feat.	34.32	46.28	52.61

We conduct ablation studies on the Pascal Context dataset [32] using CLIP with a ViT-B/16 backbone. Tab. 3 presents the quantitative results of the ablation experiment for two aspects of CLIP Surgery. CLIP Architecture Surgery enhances explainability by 47.88% based on the mSC metric, while CLIP Feature Surgery further improves it by 3.17%. For the multi-label recognition task with features from the original path, the mAP remains unchanged. However, CLIP Feature Surgery significantly enhances it by 5.52%. These results indicate that feature surgery improves both the explainability and multi-label recognition tasks.

Tab. 4 presents the ablation study on multiple layers, dual paths, and FFN. Replacing the last raw self-attention (Eq. 2) with the new self-attention (Eq. 3) results in an mSC of 29.13% (row 1). However, applying the new self-attention to multiple layers leads to a decline of -7.73% due to the modified outputs causing model instability. This issue is resolved by the proposed dual paths, which stabilize the model at an mSC of 32.30% (40.03% higher than the original single path). The FFN alone is ineffective without self-attention, while the new self-attention performs better without FFN (-4.28% vs. 34.32%). The results highlight the importance of dual paths for

multi-layers, and using the new self-attention alone achieves the best result at 34.32%.

Table 4: Ablation study about multi-layers, dual paths, and FFN. "Last" indicates the Archi. Surgery is only applied on the last layer, and "Multi" means applying to multiple layers. "Dual" is the proposed dual paths. Besides, "FFN" indicates only feed-forward networks are used, and "Attn" only takes self-attentions.

Last	Multi	Dual	FFN	Attn	mSC (%) ↑
✗	✗	✗	✗	✗	-16.73
✓	✗	✗	✗	✗	29.13
✗	✓	✗	✗	✗	-7.73
✗	✓	✓	✗	✗	32.30
✗	✓	✓	✓	✗	-4.28
✗	✓	✓	✗	✓	34.32

4.4. Results on Open-vocabulary Tasks

Table 5: Our method helps the raw CLIP archives comparison results with some SoTA open-vocabulary semantic segmentation without any further training or alignment. Besides, we list more differences in Fig. 7(b) to compare these task-specific downstream methods. All methods use the same backbone ViT-B/16, without box supervision, post-processing, or self-training for fair comparison at metric mIoU (%).

Method	Weights	Training	Core Idea	PASCAL Context	COCO Stuff	Cityscapes
ViL-Seg [41]	CLIP	yes	contrasting and clustering	16.3	16.4	-
OVSegmentor [26]	CLIP	yes	learnable group tokens	20.4	-	-
ReCo [40]	CLIP	yes	co-segmentation with retrieval	22.3	14.8	21.1
GroupViT [24]	scratch	yes	tokens grouping	22.4	13.3	12.4
Seg-CLIP [39]	CLIP	yes	grouping with learnable centers	24.7	-	-
ZeroSeg [42]	CLIP	yes	visual distillation	20.4	20.2	-
MaskCLIP [25]	CLIP	no	modify attention pooling	22.4	15.3	22.7
CLIP Surgery	CLIP	no	improve explainability	**29.3**	**21.9**	**31.4**

Semantic Segmentation: We find that the high-quality visualization results from the CLIP Surgery method are well-suited for the task of semantic segmentation. Despite requiring no extra training, our method performs exceptionally well, and even surpasses some open-vocabulary segmentation

methods that require extra training. As shown in Tab. 5, our CLIP Surgery achieves the best results on three datasets, surpassing the second-best method by 4.6%, 1.7%, and 8.7%, respectively. Besides the noticeable effectiveness, we list the differences of core ideas among methods, and CLIP Surgery is the prior work to introduce explainability into open-vocabulary segmentation task with high novelty. Note, there are many other open-vocabulary segmentation methods listed in the related work. While they require partial mask annotations [9], additional supervised proposal network [43], extra supervision [23], etc. Thus, we do not compare them in Tab. 5.

Multi-label Recognition: We compare our method with existing CLIP-based zero-shot multi-label recognition in Tab. 6. Our method significantly improves the mAP of the raw CLIP by 11.61% and 7.24% for ViT-B/16 and ResNet-50, respectively, on the NUS-Wide [37] dataset. Besides, our results have already beyond some methods requiring fine-tuning. After combining with TaI-DPT [30], we achieve new state-of-the-art result on NUS-Wide [37] at mAP 48.55% using ResNet-50, which is 1.56% higher than the implemented TaI-DPT from its official code. These results suggest our method is effective, also it's complementary to methods at zero-shot settings which require fine-tuning on seen categories.

Table 6: Results of CLIP-based multi-label recognition on NUS-Wide dataset [37]. Our method is complementary to zero-shot methods in the bottom part.

Method	Network	Fine-tuning	mAP (%)
CLIP [8]	ViT-B/16	no	35.58
Ours	ViT-B/16	no	**47.19**
CLIP [8]	ResNet-50	no	32.75
Ours	ResNet-50	no	**39.99**
DualModal [28]	ViT-B/32	yes	36.56
MKT [44]	ViT-B/16	yes	37.6
DualCoOp [29]	ResNet-50	yes	43.6
TaI-DPT [30]	ResNet-50	yes	46.99
TaI-DPT + Ours	ResNet-50	yes	**48.55**

Interactive Segmentation: Interactive segmentation [27, 45] involves segmenting a target object

from an image with user guidance. **Segment Anything Model (SAM)** [27] is a new paradigm of it. SAM enables interactive segmentation via text prompts in an open-vocabulary manner. However, it performs poorly with text prompts alone, and the authors suggest combining text with manual points for better results. Our motivation is to replace the need for manual points entirely by using CLIP Surgery with text-only inputs. Our proposed method provides pixel-level results from text input (orange points in Fig. 10 indicate predicted foregrounds and blue points are background prompts), which can be readily converted to point prompts for the SAM model. The advantages include (1) low manual efforts without combination with manual point; (2) one text for all images without interactions on every image;

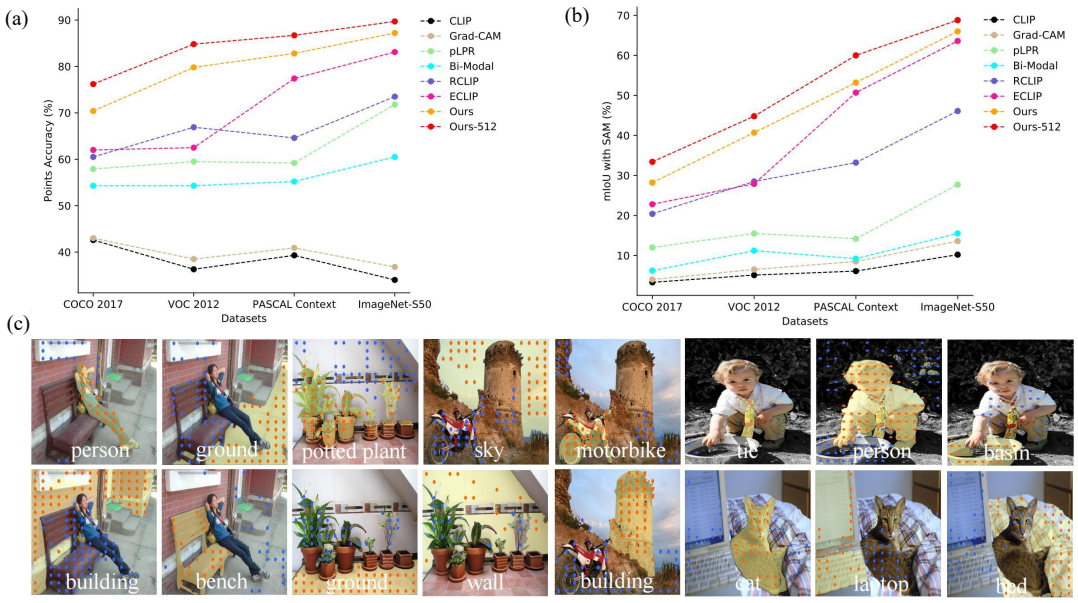


Figure 10: Comparison of points accuracy (a) and mIoU (b) with SAM [27] among explainability methods on four datasets. Points and masks of our method are shown in (c). Note, all the input size is 224, except "Ours-512" at 512.

We compare our method with other explainability methods in the **text-to-points solution**. Using ViT-B/16 as the backbone, we select points with scores above 0.8 as positives and use the same number of lowest-ranked negatives as input prompts for SAM. Fig. 10(b) illustrates the mIoU after SAM processing and the qualitative results, and Fig. 10(a) shows the accuracy of

points compared with other CAM-based methods. Note that the mIoU is evaluated independently for each positive label. Our method outperforms others at large margins and performs well visually. Especially, there are over three times improvements compared with the original CLIP and explainability methods for CNN and ViT.

Multimodal Visualization: Besides above visualization results on image modality for varied tasks, we also explain the learning process of CLIP by the multimodal visualization. Specifically, we visualize the image-text pairs during training, where the whole sentence is used as a textual label for visual explanation. At the same time, we show the top text tokens whose scores are ranked in front. For the implementation, we use image-text pairs (training data) from the GCC3M dataset [38], since the training data of CLIP is private and not available. Then we draw the similarity map via our CLIP Surgery method and mark the high-response text tokens. Specifically, the feature of the class token $\boldsymbol{F_c}$ is used to compute similarity scores for each text token, and the text at max similarity is served for the generation of similarity map with image tokens $\boldsymbol{F_i}$.

We draw multimodal visualization results as Fig. 11. From these multimodal visualization results, we observed some interesting phenomena. For the visual results, we summarize two points: (1) Not all the objects or stuff are highlighted in the image, because only the text token at the highest cosine similarity is picked for training (red texts in Fig. 11). Thus, we believe CLIP learns partial context from one image. (2) CLIP can recognize the texts from an image to some extent, as shown in the last two images of Fig. 11. Since the highlights are corresponded with text tokens (e.g., day, enjoy, relationship). For the textual visualization results, there are two findings: (1) The end token is the most common activated text token, and some non-object words are at high response too (e.g., "in", ".", "of"). It suggests there are also redundant tokens in the vocabulary dictionary. (2) The object-based words also occur frequently with corresponding salient objects in the image. While their cosine similarities often rank second or third behind the end token "[end]". These findings are interesting, also they reveal some characteristics of image-text pairs of CLIP, thus providing potential value to further improvement of CLIP's training process.

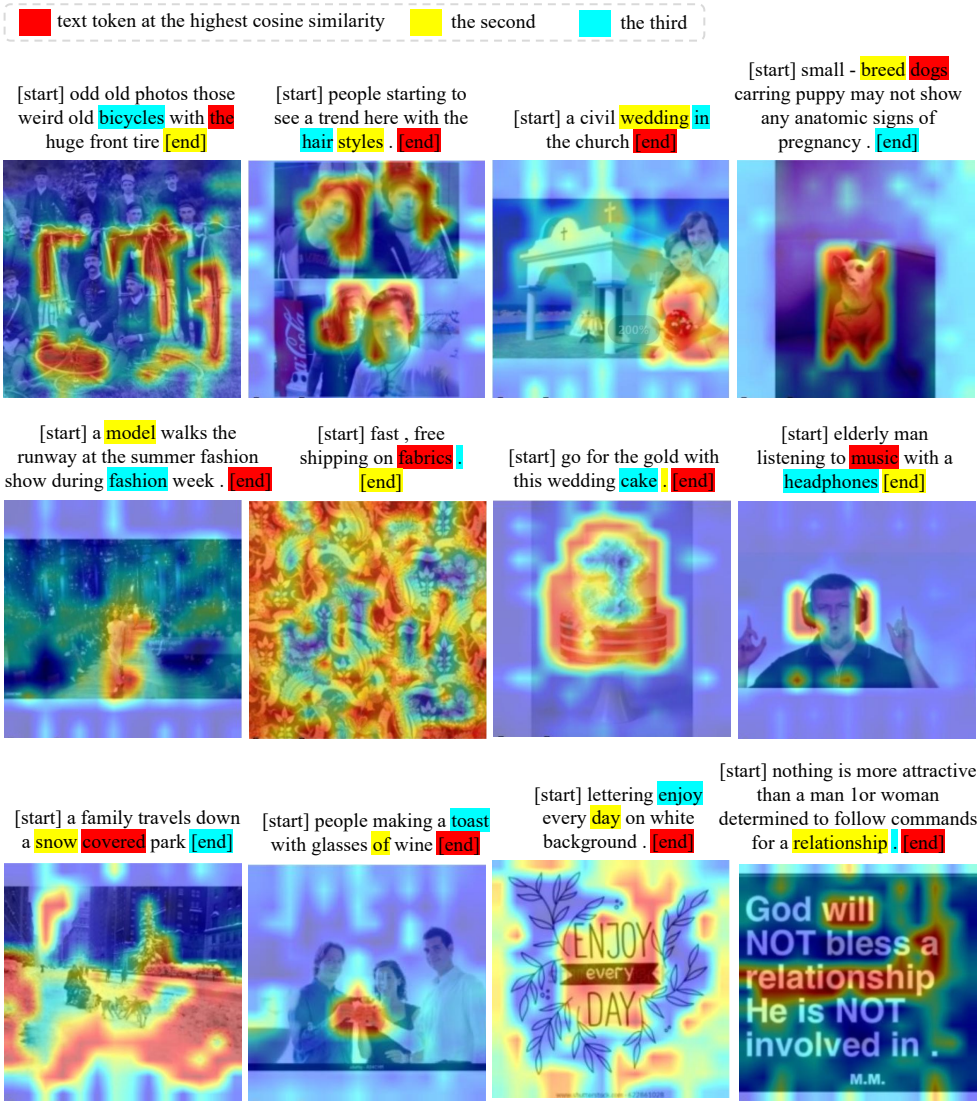


Figure 11: Multimodal visualization to explain the image-text pairs used in the training process of CLIP. The visual explainability results are from the proposed CLIP Surgery, and textual explainability results are based on the cosine distance of each text token. [start] indicates the start text token and [end] means the end text token. Note that we mark the text token at the highest cosine similarly in red, then draw the second in yellow and blue for the third.

5. Conclusion

In this study, we investigate two observations related to CLIP's explainability: opposite visualization and noisy activations. We discover that the raw self-attentions build relations on inconsistent semantic regions, resulting in the opposite visualization. To address it, we propose the CLIP architecture surgery, merging the consistent self-attentions with a dual paths structure. For the noisy activation in CLIP, it arises from redundant features among categories, then we introduce the CLIP feature surgery on output features to mitigate the common but redundant activations.

The proposed method significantly enhances the visual explainability of CLIP for reliable CAM, which plays a crucial role in promoting model transparency. Moreover, our method further enhances downstream tasks like semantic segmentation, interactive segmentation, and multi-label recognition, showing remarkable improvements. Besides, it provides valuable insights into the architecture, features, and learning process of CLIP, which boosts our understanding of CLIP and benefits it further improvements. Overall, the proposed CLIP Surgery offers a promising solution to generate high-quality CAM for CLIP, with wide applicability and valuable insights.

CRediT authorship contribution statement

Yi Li: Conceptualization, Investigation, Methodology, Validation, Visualization, Writing. **Hualiang Wang**: Conceptualization, Investigation, Methodology. **Yiqun Duan**: Conceptualization, Investigation. **Jiheng Zhang**: Resources. **Xiaomeng Li**: Conceptualization, Investigation, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

We declare that this manuscript has not been published before and is not currently being considered for publication elsewhere. We confirm that the manuscript has been approved by all authors for publication, and no conflict of interest exits in the submission of it.

Data availability

All data used in the research are publicly available.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China un-

der Grant 62306254, and also by grants from Foshan HKUST Projects under Grants FSUST21-HKUST10E and FSUST21-HKUST11E.

References

[1] L. Baur, K. Ditschuneit, M. Schambach, C. Kaymakci, T. Wollmann, A. Sauer, Explainability and interpretability in electric load forecasting using machine learning techniques–a review, Energy and AI (2024) 100358.

[2] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921–2929.

[3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.

[4] Y. Li, Z. Kuang, L. Liu, Y. Chen, W. Zhang, Pseudo-mask matters in weakly-supervised semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6964–6973.

[5] L. Yu, W. Xiang, J. Fang, Y.-P. P. Chen, L. Chi, ex-vit: A novel explainable vision transformer for weakly supervised semantic segmentation, Pattern Recognition 142 (2023) 109666.

[6] H. Yu, H. Lu, M. Zhao, Z. Li, G. Gu, Gradient aggregation based fine-grained image retrieval: A unified viewpoint for cnn and transformer, Pattern Recognition 149 (2024) 110248.

[7] O. Bar-Tal, D. Ofri-Amar, R. Fridman, Y. Kasten, T. Dekel, Text2live: Text-driven layered image and video editing, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV, Springer, 2022, pp. 707–723.

[8] 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.

[9] M. X. et al., A simple baseline for open vocabulary semantic segmentation with pre-trained vision-language model, Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV) (2022).

[10] Z. Tan, X. Yang, Z. Ye, Q. Wang, Y. Yan, A. Nguyen, K. Huang, Semantic similarity distance: Towards better text-image consistency metric in text-to-image generation, Pattern Recognition 144 (2023) 109883.

[11] Y. Li, H. Wang, Y. Duan, H. Xu, X. Li, Exploring visual interpretability for contrastive language-image pre-training, arXiv preprint arXiv:2209.07046 (2022).

[12] Y. Huang, A. Jia, X. Zhang, J. Zhang, Generic attention-model explainability by weighted relevance accumulation, in: Proceedings of the 5th ACM International Conference on Multimedia in Asia, 2023, pp. 1–7.

[13] F. Zheng, J. Cao, W. Yu, Z. Chen, N. Xiao, Y. Lu, Exploring low-resource medical image classification with weakly supervised prompt learning, Pattern Recognition 149 (2024) 110250.

[14] H. Chefer, S. Gur, L. Wolf, Transformer interpretability beyond attention visualization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 782–791.

[15] H. Chefer, S. Gur, L. Wolf, Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 397–406.

[16] P. Chen, Q. Li, S. Biaz, T. Bui, A. Nguyen, gscorecam: What objects is clip looking at?, in: Proceedings of the Asian Conference on Computer Vision, 2022, pp. 1959–1975.

[17] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking clever hans predictors and assessing what machines really learn, Nature communications 10 (1) (2019) 1096.

[18] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.

[19] Y. Li, H. Liang, H. Zheng, R. Yu, Cr-cam: Generating explanations for deep neural networks by contrasting and ranking features, Pattern Recognition 149 (2024) 110251.

[20] 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 (2021).

[21] R. Paiss, H. Chefer, L. Wolf, No token left behind: Explainability-aided image classification and generation, in: European Conference on Computer Vision, Springer, 2022, pp. 334–350.

[22] R. Huang, X. Pan, H. Zheng, H. Jiang, Z. Xie, C. Wu, S. Song, G. Huang, Joint representation learning for text and 3d point cloud, Pattern Recognition 147 (2024) 110086.

[23] H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Yang, L. Zhang, A simple framework for open-vocabulary segmentation and detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 1020–1031.

[24] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, X. Wang, Groupvit: Semantic segmentation emerges from text supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18134–18144.

[25] C. Zhou, C. C. Loy, 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.

[26] J. Xu, J. Hou, Y. Zhang, R. Feng, Y. Wang, Y. Qiao, W. Xie, Learning open-vocabulary semantic segmentation models from natural language supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2935–2944.

[27] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al., Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026.

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

[29] X. Sun, P. Hu, K. Saenko, Dualcoop: Fast adaptation to multi-label recognition with limited annotations, Advances in Neural Information Processing Systems 35 (2022) 30569–30582.

[30] Z. Guo, B. Dong, Z. Ji, J. Bai, Y. Guo, W. Zuo, Texts as images in prompt tuning for multi-label image recognition, in: CVPR, 2023, pp. 2808–2817.

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

[32] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 891–898.

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[34] S. Gao, Z.-Y. Li, M.-H. Yang, M.-M. Cheng, J. Han, P. Torr, Large-scale unsupervised semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).

[35] H. Caesar, J. Uijlings, V. Ferrari, Coco-stuff: Thing and stuff classes in context, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1209–1218.

[36] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213–3223.

[37] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: a real-world web image database from national university of singapore, in: Proceedings of the ACM international conference on image and video retrieval, 2009, pp. 1–9.

[38] P. Sharma, N. Ding, S. Goodman, R. Soricut, Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 2556–2565.

[39] H. Luo, J. Bao, Y. Wu, X. He, T. Li, Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation, in: International Conference on Machine Learning, PMLR, 2023, pp. 23033–23044.

[40] G. Shin, W. Xie, S. Albanie, Reco: Retrieve and co-segment for zero-shot transfer, Advances in Neural Information Processing Systems 35 (2022) 33754–33767.

[41] Q. Liu, Y. Wen, J. Han, C. Xu, H. Xu, X. Liang, Open-world semantic segmentation via contrasting and clustering vision-language embedding, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX, Springer, 2022, pp. 275–292.

[42] J. Chen, D. Zhu, G. Qian, B. Ghanem, Z. Yan, C. Zhu, F. Xiao, S. C. Culatana, M. Elhoseiny, Exploring open-

vocabulary semantic segmentation from clip vision encoder distillation only, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 699–710.

[43] 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. 16793–16803.

[44] S. He, T. Guo, T. Dai, R. Qiao, X. Shu, B. Ren, S.-T. Xia, Open-vocabulary multi-label classification via multi-modal knowledge transfer, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 808–816.

[45] M. Ô. V. Ngc, E. Carlinet, J. Fabrizio, T. Géraud, The dahu graph-cut for interactive segmentation on 2d/3d images, Pattern Recognition 136 (2023) 109207.