Zero-guidance Segmentation Using Zero Segment Labels

Pitchaporn Rewatbowornwong[2*] Nattanat Chatthee[2*] Ekapol Chuangsuwanich[1] Supasorn Suwajanakorn[2]

[2]VISTEC, Thailand [1]Chulalongkorn University, Thailand

{pitchaporn.r_s18, nattanat.c_s23, supasorn.s}@vistec.ac.th ekapol.c@chula.ac.th

Abstract

The joint visual-language model CLIP has enabled new and exciting applications, such as open-vocabulary segmentation, which can locate any segment given an arbitrary text query. In our research, we ask whether it is possible to discover semantic segments without any user guidance in the form of text queries or predefined classes, and label them using natural language automatically? We propose a novel problem ***zero-guidance segmentation*** *and the first baseline that leverages two pre-trained generalist models, DINO and CLIP, to solve this problem without any fine-tuning or segmentation dataset. The general idea is to first segment an image into small over-segments, encode them into CLIP's visual-language space, translate them into text labels, and merge semantically similar segments together. The key challenge, however, is how to encode a visual segment into a segment-specific embedding that balances global and local context information, both useful for recognition. Our main contribution is a novel attention-masking technique that balances the two contexts by analyzing the attention layers inside CLIP. We also introduce several metrics for the evaluation of this new task. With CLIP's innate knowledge, our method can precisely locate the Mona Lisa painting among a museum crowd (Figure 1). More results are available at* `https://zero-guide-seg.github.io/`.

1. Introduction

Semantic segmentation is a core computer vision problem that seeks to partition an image into semantic regions. Traditionally, the semantic classes of interest need to be predefined and are limited in number [22]. Earlier methods thus cannot generalize beyond the training classes. With recent advances in joint vision-language representation learning, e.g., CLIP [25], newer methods [21, 19, 36] can successfully predict segments corresponding to arbitrary text queries in a novel task called open-vocabulary segmentation. These segmentation methods are guided by a text query, which describes what already exists in the image and must be provided by the user. Another meaningful milestone, however, is how we can segment an image *without* user input or guidance like text queries or predefined classes, and label such segments automatically using natural language. Our work provides the first baseline for this novel problem, referred to as *zero-guidance segmentation.*

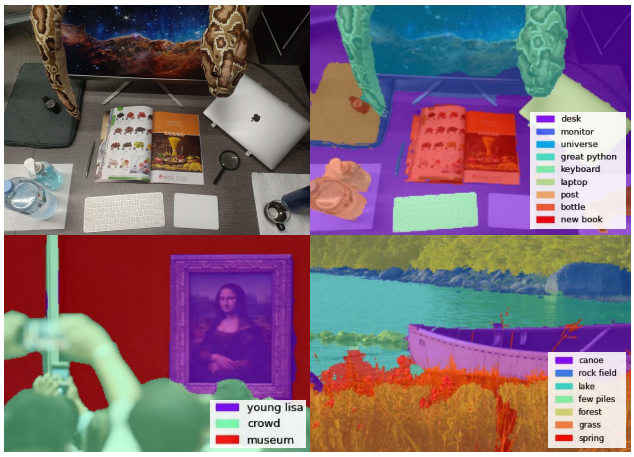

Figure 1. **Zero-guidance Segmentation** segments input images and generated text labels for all segments without any guidance or prompting. Our method produces these results using only pre-trained networks with no fine-tuning or annotations.

Our work is inspired by a recent research direction that solves segmentation by leveraging CLIP [36, 39]; however, our key distinction is that we require no segmentation datasets, no text query guidance, and no additional training or fine-tuning. This problem is challenging partly because CLIP has been trained with image captions that globally describe the scenes and provide no spatially specific information for learning segmentation. Surprisingly, we show that it is possible to distill the learned knowledge from two generalist models: a self-supervised visual model, DINO [2], and a visual-language model, CLIP [25], to solve zero-guidance segmentation without further training.

The overall idea is to first over-segment an image into small segment candidates, then translate each segment into

*Equal contribution

words, and finally join semantically similar segments to form the output segments. In particular, we identify segment candidates by clustering deep pixel-wise features from a DINO that takes as input our image. Despite using no training labels, DINO has been shown to produce class-discriminative features, allowing unsupervised segmentation of the primary object in an image [2] or part co-segmentation across images [1]. However, our main challenge lies in the next step, which maps each segment into a meaningful representation that can be later translated to words. We leverage CLIP's joint space to model this intermediate representation.

A naive way to project a segment to CLIP's joint space is to input the segment directly into CLIP's image encoder, but this entirely ignores the surrounding context needed for object recognition and disambiguation [38, 5, 4]. Alternatively, masking can be applied inside the attention layers, as done with a transformer-based segmentation network [6]. However, when applied to CLIP's encoder, which was not trained for segmentation, these techniques struggle to produce segment-specific embeddings due to the domination of global context information. Evidence in [40, 37, 13] also suggests that a transformer trained with image-level annotations, such as CLIP, may lose local information in its tokens in later layers. We discovered similar issues: masking in earlier layers removes global contexts and hurts recognition, whereas masking in later layers fails to focus the embedding on a given segment, resulting in all embeddings describing the same dominant object in the image.

Another difficulty in balancing global and local contexts is that different objects may require different degrees of context balancing. For small objects, their CLIP embeddings can be dominated by global contexts, which describe other prominent objects in the scene. This phenomenon matches the characteristics of CLIP's training captions, which often ignore unimportant objects in the image. As a result, less prominent objects may require less of the global contexts to highlight their semantics and local contexts.

To solve this, we introduce a novel attention-masking technique called *global subtraction*, which helps adjust the influence of global contexts in the output embedding. The key idea is to first estimate the saliency or the presence of a given segment in the global contexts by analyzing CLIP's attention values. Then, this saliency value will be used to determine how much global contexts should be attenuated in the segment's embedding. The resulting embedding in CLIP's joint vision-language space allows us to readily translate it to text labels with an existing image-to-text generation algorithm [30]. And finally, we merge semantically similar segments with simple thresholding by considering both their visual and text similarities.

To evaluate our algorithm that can output arbitrary text labels, we also propose new evaluation metrics. Evaluating an algorithm under this setup is not straightforward as predicted labels may not necessarily match predefined labels in the test set but can still be correct. This may result from the use of synonyms, such as "cat" vs. "feline," or differences in label granularity, such as "cat" vs. "orange cat" or "cat's nose" or "kitten." Generally, there is no single correct level of granularity, and each dataset may arbitrarily adopt any level. To address this, we propose to first map the predicted semantic labels to the existing ones in a given test set. After that, we can use standard measurements, such as segment IoU, to evaluate the results as if the algorithm performs segmentation with the predefined test classes. We also introduce Segment Recall, which measures how often ground-truth objects are discovered, and Text Generation Quality, which tests the quality of our embedding technique given ground-truth oracle segmentation.

Our technique can automatically segment an image into meaningful segments as shown in Figure 1 without any supervision or text guidance. There are still performance gaps between our technique and other supervised methods or methods fine-tuned on segmentation datasets—but none can specifically solve our problem that lacks user guidance. Nonetheless, we provide a detailed analysis on obstacles that lie ahead as well as ablation studies for the first approach to this problem. In summary, our contributions are:

- We introduce the first baseline to a novel problem, *zero-guidance segmentation*, which aims to segment and label an input image in natural language without predefined classes or text query guidance. Our method does not require a segmentation dataset or fine-tuning.
- We propose a novel attention-masking technique to convert a segment into an embedding in CLIP's joint space by balancing global and local contexts.
- We present evaluation metrics for the proposed setup.

2. Related Work

Open-vocabulary segmentation. This problem aims to predict segments in an input image that correspond to a set of input texts not necessarily seen during training. Prior solutions often involve a shared latent space between the image and text domains. OpenSeg [11] uses datasets of images, captions, and class-agnostic segmentation masks to train a mask proposal network before matching the predicted masks with nouns in the captions using a shared latent space. OVSeg [21] built a segmentation pipeline that fine-tunes CLIP on masked images to make it more suitable for masked image classification. Xu et al. [36] proposed a zero-shot segmentation baseline by matching CLIP's embeddings of masked images to text embeddings of classes. ZegFormer [8] performs class-agnostic pixel grouping to create segments and uses CLIP to classify them. Lseg [19]

trains an image pixel encoder that encodes each pixel into an embedding that is close to the corresponding text labels' embeddings in CLIP's space. These methods show impressive results but still demand expensive segmentation labels.

To avoid the use of segmentation datasets, GroupViT [34] proposes a new method based on hierarchical vision transformers where the visual tokens represent arbitrary regions instead of patches in a square grid. By using only image-caption pairs, GroupVit can match each region from its visual tokens to input text prompts. Zhou et al. [39] modifies CLIP for text-guided segmentation and employs self-training to improve the results. In contrast, our work requires neither additional training nor text prompts but can discover semantic segments and label them automatically.

Attention masking in transformer. Masking self-attention is a common practice in NLP to input a word sequence more efficiently [31]. In computer vision, few explorations exist: Mask2Former [6] solves supervised segmentation by masking self-attention layers of a transformer decoder, achieving state-of-the-art results. Unlike Mask2Former, which is trained on specific segmentation datasets, our method and the base models we used (CLIP and DINO-ViT) do not have any explicit segmentation supervision. We found that using the masking mechanism of Mask2Former yields noisy CLIP embeddings, which are often heavily biased toward the foreground objects. This can be solved by our proposed global subtraction technique.

Image segmentation with DINO. DINO [2] is a model that uses self-distillation to learn rich features of an input image with no supervision and has been used as a pretrained network or representation extractor in many tasks [32, 33, 29, 14]. Caron et al. [2] demonstrated that DINO's features effectively capture object boundaries and scene layout [2], and Hamilton et al. [14] further showed that these features can perform segmentation of not only foreground objects but also other elements in the background, such as the sky. Our method uses a simple DINO-based clustering, inspired by Amir et al. [1], which requires no training and offers reasonable results. Note that our key contribution in attention masking is orthogonal to this clustering choice.

Image-to-text generation with CLIP. The recent advent of CLIP leads to new approaches in text-image tasks, including generating text from an input image. ClipCap [23] trains a mapping network that joins CLIP with a pretrained language model, GPT-2, and performs image captioning with faster training. ZeroCap [30] performs zero-shot image caption by optimizing the value matrix V in each attention module in GPT-2 to guide the embedding of the output text toward the target image's embedding. The output texts display knowledge learned from CLIP's vast and diverse training set, such as names of celebrities and pop culture references. This is a new ability unseen in older image captioning methods. Note that our contribution is not directly in text generation, rather we focus on inferring semantic segments and mapping them to CLIP's latent space.

3. Approach

Given an input image, our goal is to partition this image into semantic segments and label each segment using words in natural language. Our framework consists of four stages: 1) we identify segment candidates based on clustering deep per-pixel features of DINO-ViT [2], 2) we map each segment to an embedding in the CLIP's visual-language space using our proposed attention masking technique, 3) we translate each CLIP embedding into words by optimizing a generative language model with an existing technique, ZeroCap [30], and 4) we merge segments with similar semantics.

3.1. Finding segment candidates with DINO

The goal of this step is to partition the input image into small over-segments, which will be merged in the final step. To do so, we first extract spatial features of the input image from DINO-ViT. In particular, we use the "key" values from the last attention layer as the features (following [1]), which have a total dimension of (#patch$\times C$). Unlike in standard use of ViT, we use a small stride of two instead of the patch size, resulting in a dense feature map ($\frac{H}{2} \times \frac{W}{2} \times C$).

Given this dense feature map, we initially assign each feature vector ($1 \times C$) its own cluster and perform agglomerative clustering by repeatedly merging any two clusters with the smallest combined feature variance. We stop this process when the target number of clusters $n = 20$ is reached, and we additionally merge clusters with similar feature vectors based on their cosine similarity, detailed in Appendix B. The output segments from this step may break single objects into small parts, which lack semantic meanings by themselves. However, our decision to oversegment first allows merging in the semantic space of CLIP later on, which takes into account both vision and language semantics and can be done with simple thresholding.

3.2. Transforming segment candidates into CLIP's vision-language embeddings

To map a given segment to CLIP's vision-language space, our idea is to feed the entire input image into CLIP's image encoder while masking some of the encoder's attention layers with an alpha mask corresponding to the given segment. One major consideration is which layers should the masking be applied to properly balance global and local contexts. This turns out to be challenging: masking in earlier layers destroys global contexts, whereas masking in later layers eliminates local contexts. This finding agrees with several studies [40, 37, 13] showing that vision transformers trained for classification suffer from an "attention collapse," where the attention in deeper layers becomes near

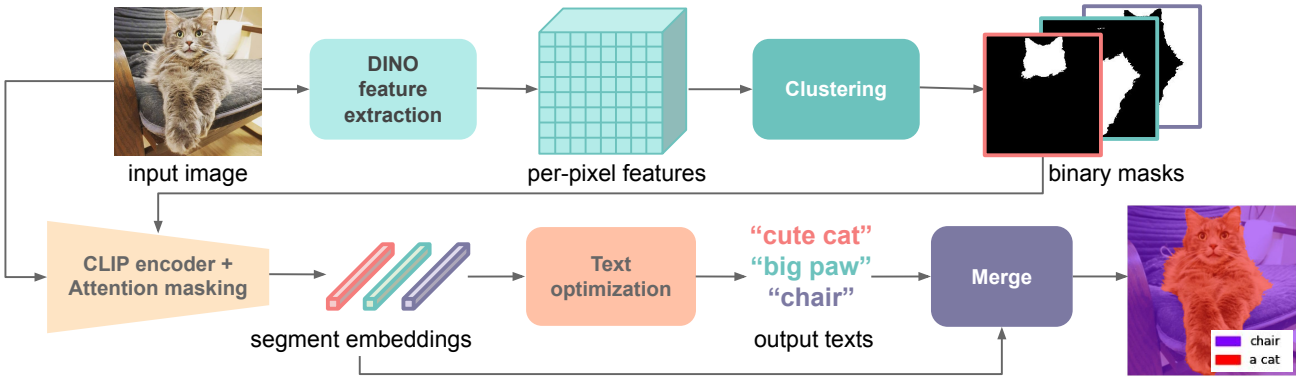

Figure 2. **Pipeline Overview.** Our method first segments an input image by clustering learned per-pixel features extracted from DINO. The input image is then fed into CLIP's image encoder. In this step, the produced segmentation masks are used to modify CLIP's attention to provide embeddings that are more focused to each segment. The resulting embeddings are then used to optimize a trained language model to generate texts closest to these embeddings. Lastly, segments with similar embeddings and text outputs are merged together to form more coherent segmentation results.

uniform and all tokens converge to the same value. Another study [28] also suggests that CLIP may lack the ability to maintain local information. In Appendix A, we show how CLIP's attention maps become less localized in later layers.

Masking in the middle layers also performs poorly because different objects still require different degrees of context balancing depending on how salient they are in the scene. For example, the embedding of a small, obscure object in the background can be dominated by global contexts, which describe other prominent objects in the scene. As a result, these small objects may require more de-emphasis of the global contexts for their semantics to emerge.

Based on this observation, we propose a simple technique to estimate the saliency of each segment and use it to modulate how much global information should be removed or subtracted from individual tokens during attention masking. We next explain how we apply masking to the attention module, and then our *global subtraction* technique.

3.2.1 Masking in self-attention module

Given a logit vector $x \in \mathbb{R}^n$ and a flattened mask $M \in [0,1]^n$, we first define the masked softmax operator as:

$$\text{MaskedSoftmax}(x, M) = \frac{e^x \odot M}{\sum_{j=1}^{n} (e^{x_j} \times M_j)}, \quad (1)$$

where $\odot$ denotes the element-wise multiplication. To mask a standard attention module, we compute $A_i^{\text{masked}} = \text{MaskedSoftmax}\left(Q_i K^T / \sqrt{d_k}, M\right) V$ for *every* token i. In practice, when the mask size is larger than the visual patch grid, we first downsample M to the same size using area interpolation. We also prepend one extra element to the flattened M for the global token, which is always set to one in our algorithm, and thus #tokens = #patches + 1.

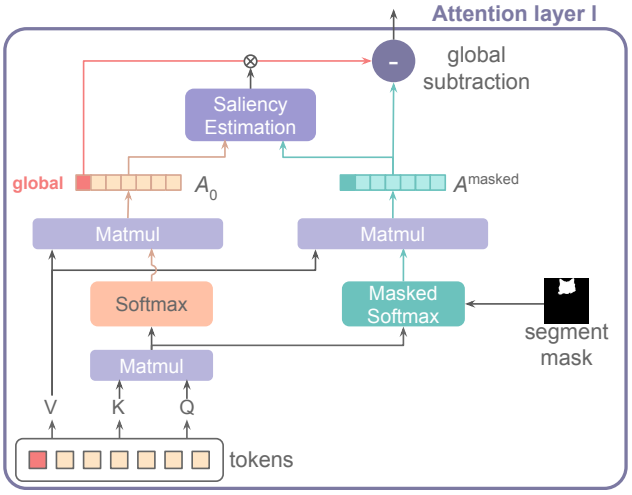

Figure 3. **Attention masking and global subtraction.** To encode a segment into CLIP's space, we pass the input image into CLIP's image encoder and mask self-attention map in some layers with the segment's mask. We apply this masking inside masked softmax function while still computing normal softmax. Cosine similarity between masked and unmasked output is used to estimate the saliency of the region. This similarity determines how much global context needs to be reduced in global subtraction.

3.2.2 Global subtraction

To balance global and local contexts in our output embedding, we design a proxy function that estimates the "saliency" of each segment or the segment's presence in the global contexts. This value will be used to determine how much global contexts should be removed from the attention output. Note that our saliency value is only defined with respect to the attention mechanism in CLIP and is unrelated to other uses of "saliency" in the literature [16, 12].

We perform the following operations separately for each attention layer l using its own saliency value $\mathcal{S}_l$. We compute $\mathcal{S}_l$ as the cosine similarity between the masked atten-

tion output A^{masked} and unmasked attention output $A = \text{Softmax}(QK^T/\sqrt{d_k})V$ at layer l, averaged over all tokens (we omit the subscript l from A and A^{masked} for simplicity):

$$\mathcal{S}_l = \frac{1}{\#\text{tokens}} \sum_{i=1}^{\#\text{tokens}} \text{cossim}(A_i, A_i^{\text{masked}}). \quad (2)$$

The output of the attention layer l is computed by subtracting the unmasked attention output of the global token A_0 from the masked attention A^{masked}:

$$A^{\text{out}} = A^{\text{masked}} - wA_0, \quad (3)$$

$$\text{where } w = \exp(-(\mathcal{S}_l + 1)^2/2\sigma^2). \quad (4)$$

This global subtraction weight w is computed by applying a Gaussian function with a standard deviation σ to $(\mathcal{S}_l + 1)$, making w highest when $\mathcal{S}_l = -1$. In other words, when an object is *not* salient, we remove *more* global contexts from its attention output. We apply these masking operations starting from attention layer 21, which is chosen empirically, to the last layer 24. Finally, the embedding of our segment is the output from our masked CLIP's encoder, which additionally applies linear projection to the global token value from the last attention layer.

3.3. Text generation from CLIP's embedding

To translate our segment embedding in CLIP's joint space into words, we use an existing image-to-text generation algorithm, ZeroCap [30]. This method uses a pre-trained language model GPT-2 [26] along with CLIP to optimize for a sentence that describes an input image. This is done by optimizing specific activations of GPT-2 (K and V matrices) to complete an initial prompt of "Image of a ..." and minimizing the difference between the output sentence and the input image in CLIP's joint space.

3.4. Merging segment candidates

In this step, we merge segments that are semantically similar or small segments that may not be so meaningful by themselves from our oversegmentation. We compute the similarity score between two segments using the average of two measures: 1. the cosine similarity between their visual embeddings (Section 3.2) and 2. the cosine similarity between their predicted texts' embeddings computed from CLIP's text encoder. In our implementation, we also reduce the number of merging combinations by limiting the pairing option. In particular, we first continue the agglomorative clustering in the first step (Section 3.1) until there is a single cluster representing the entire image. By keeping track of the merging history, we obtain a binary tree where each node represents a segment and each parent is the merged segment of its children. We limit the pairing to only between siblings in this tree and recursively merge segments up the tree when their similarity score is at least τ_{merge}. The final embedding of a merged segment is computed by passing the corresponding merged mask through the embedding pipeline (Section 3.2) and is then used to generate the final predicted text.

4. New Evaluation Protocol

This section introduces new metrics to evaluate the quality of the output segments and their corresponding text labels. To overcome the evaluation challenges due to the use of synonyms or the difference in label granularity, such as "car" vs "wheel," we first map the predicted labels to the predefined ones in the test set (Section 4.1) and verify the reassignment using thresholding or human evaluation (Section 4.2) before applying standard metrics such as IoU.

4.1. Label reassignment

Given a predicted segment S_i and its predicted text T_i, our goal is to relabel S_i with T_i^*, which should be one of the test labels. We describe two reassignment techniques based on text-to-text and segment-to-text similarity.

Text-to-text similarity (TT). This technique relabels S_i with the ground-truth label that is closest to T_i in the embedding space of Sentence-BERT [27], a pre-trained text encoder widely used in NLP for computing text similarity [10, 20, 7]. Formally, the new label T_i^* is computed by

$$T_i^* = \arg\max_{t \in T^{\text{gt}}} \left[\text{cossim}^{\text{SBERT}}(T_i, t)\right], \quad (5)$$

Segment-to-text similarity (ST). This technique relabels S_i with the ground-truth label that is closest to S_i in the CLIP's joint image-text space [15, 17]. That is,

$$T_i^* = \arg\max_{t \in T^{\text{gt}}} \left[\text{cossim}^{\text{CLIP}}(S_i, t)\right], \quad (6)$$

where $\text{cossim}^{\text{CLIP}}(s, t)$ uses CLIP's image encoder for s and text encoder for t. Note that this relabeling is commonly used in open-vocabulary settings [21, 11], but it does not consider our predicted label T_i during relabeling. Nonetheless, this technique is still valuable as it offers a complementary assessment that does not involve text generation, which is based on prior work (Section 3.3), or text-to-text mapping, which can be challenging and ambiguous even for human evaluators (Section 6).

4.2. Reassignment verification

For evaluation, we need to verify that the reassigned label T_i^* is sufficiently close to the original label T_i or its segment S_i. We provide two kinds of verification. The first is based on simple thresholding on the cosine similarity using τ_{SBERT} and τ_{CLIP} for TT and ST reassignments, respectively (Appendix C). The second involves human judgement, in

which we ask human evaluators to rate how well the reassigned label describes its segment on a scale of 0-3, ranging from 0: incorrect, 1: partially correct, 2: correct but too general/specific, 3: correct. The full definitions are in Appendix F. Multiple thresholds will be used to report scores.

4.3. Metrics

Segmentation IoU evaluates the quality of the output segments in terms of Intersection-over-Union (IoU) against the ground-truth segments in each test image. Given a set of predicted segments with reassigned labels T^*, segments with the same label T^* are merged to form a single segment for the label. Then, IoU for each image can be computed using a standard protocol [9].

Segment Recall measures how many objects labeled in the ground truth are discovered. This metric disregards any extra labels predicted by our method that are not part of the ground truth labels. We consider each merged segment of the same reassigned label a True Positive if its IoU against the corresponding ground-truth segment is greater than τ_{IoU}. Segment Recall is the rate of True Positive over the number of grounding segments.

Text Generation Quality measures the quality of text generation given an oracle segmentation. That is, we feed each *ground-truth* segment into our model and compute the cosine similarity between our predicted label and the ground-truth label. If the value is higher than τ_{SBERT}, it is considered a True Positive. The score is the True Positive rate over the entire test set. This metric evaluates our attention-masking and text generation components independent of the segment generation process (Section 3.1).

5. Experiments

Datasets. We evaluate our results on two commonly used segmentation datasets: Pascal Context [24] and Pascal VOC 2012 [9]. Pascal Context contains 5,000 validation images with segmentation ground truths of 459 object classes for scene segmentation task. We use PC-59, the commonly used subset with 59 most common objects, following [11, 34, 21], as well as the full PC-459. Pascal VOC (PAS-20) is a segmentation dataset with 1,500 image-segment validation pairs of 20 object classes. For both datasets, we report our results on the validation splits, as the test splits are not publicly available, but the validation splits were never used for hypertuning. For comparison with our own variations and a crop-and-mask baseline, we test on the first 1,000 images of Pascal Context dataset and full 1,500 image for Pascal VOC dataset (Section 5.3). We use the full datasets when compared to prior work (Section 5.4).

Table 1. Quantitative results on 1,000 random images from PAS-59's validation split. We use constants (τ_{SBERT} and τ_{CLIP}) and multiple human verification scores (h) for thresholding.

	Text-text reassign.		**Segment-text reassign.**			
Threshold:	const.	$h \geq 1$	const.	$h = 3$	$h \geq 2$	$h \geq 1$
IoU	11.2	11.0	19.3	14.2	20.9	22.7
Recall	10.3	9.8	18.0	13.2	18.0	19.4

Table 2. Distribution of human rating scores on the quality of the predicted labels (0: incorrect, 1: partially correct, 2: correct but too general/specific, 3: correct).

Human rating	0	1	2	3
% of labels	36.0	20.8	23.9	19.3

5.1. Zero-guidance segmentation results

We present our qualitative results in Figure 1, 4. Our method can discover semantic segments and densely label them with diverse types of labels, including names of objects, animal breeds, facial expressions, and places. More results are in Appendix G. In Table 1, we report IoU and Recall scores using different reassignment and verification techniques (Section 4.2), computed on 1,000 randomly sampled images from PC-59. We observe that ST tends to perform reassignment better than TT, as evident by its higher scores. Reassigning words like 'leg' to the correct animal class in the ground-truth set can be challenging when relying solely on text (TT), as it lacks any additional context. But ST can access other visual information within the segment, which better facilitates reassignment.

5.2. User study

We evaluate the quality of predicted labels using human evaluation. Each segment and its predicted label were shown to three distinct human evaluators, who were asked to rate how well the label describes the segment on a scale of 0-3, similar to the process in Section 4.2 except we show the predicted label T_i instead of the reassigned label T_i^*. Full details and the score definitions are in Appendix F.

Table 2 shows that human evaluators found about 43% of our results to be 'correct' or 'correct but too generic/specific' and 64% to be at least 'partially correct.'

We provide example images and their scores given by the human evaluators in Figure 6. According to the result, most of our score-0 labels are single-word adjectives, such as 'black', or collective nouns, such as 'group'. Another kind of score-0 labels is caused by biases toward stereotypical appearances of objects, such as when a pet dog was mislabeled as 'stray' due to its shabby appearance (row 4). Some of score-1 labels correspond to descriptions or abstract nouns that are related to their segments but may not fully describe them, such as 'reflection', 'dining', and 'sunny', and some other labels describe specific but incor-

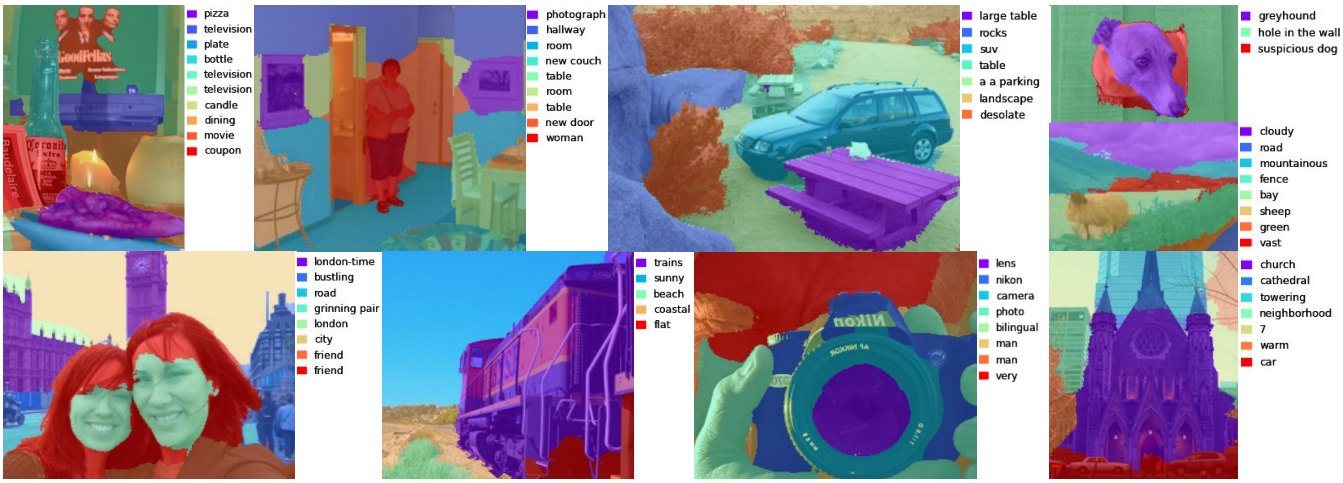

Figure 4. **Qualitative results.** Our method gives reasonable segments and output free-language text labels representing all regions. The labels can describe regions by different kinds of descriptions, such as object names, facial expressions, locations, car models, or even animal breeds.

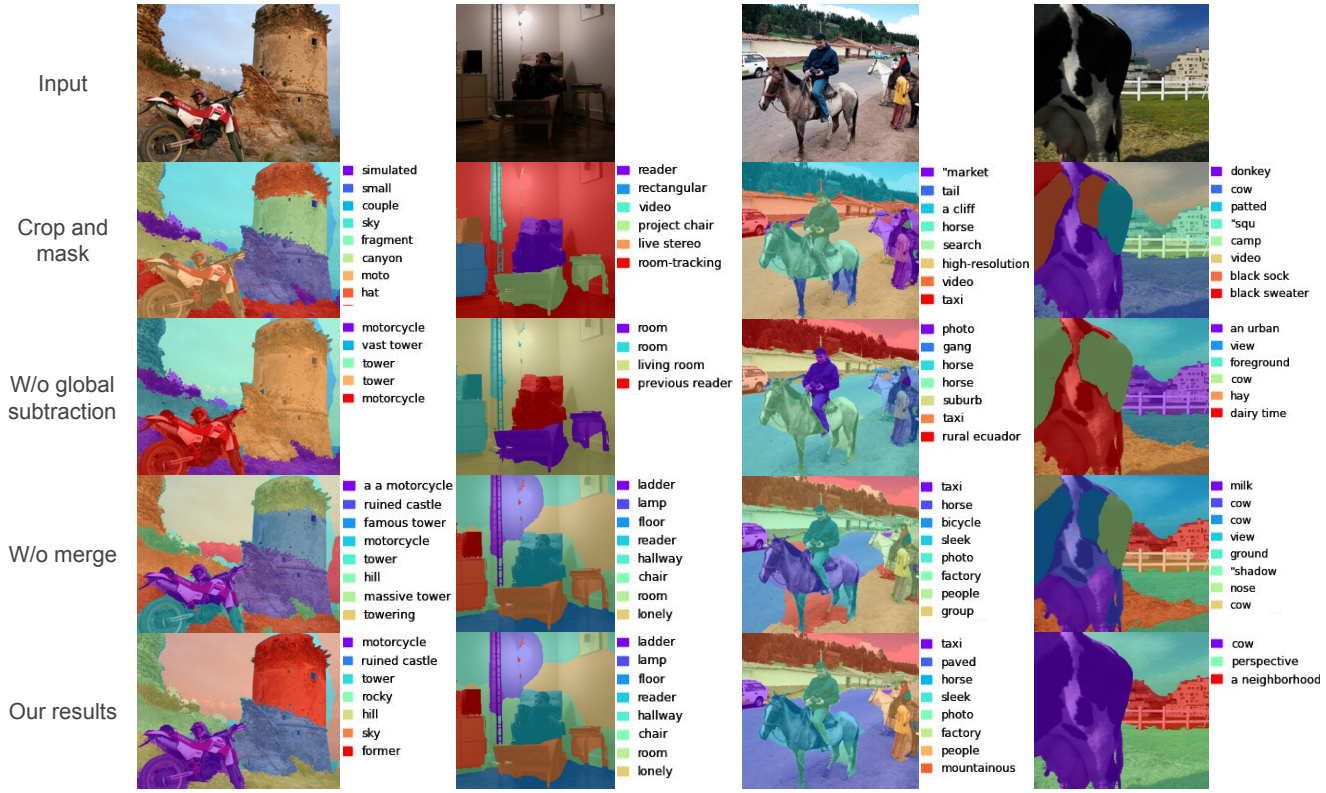

Figure 5. **Ablation results.** Comparison between the results from different segment encoding methods. The crop-and-mask baseline often outputs text labels that is not relevant to segments/input images. Our method without global subtraction suffers from global leak and often mislabels non-salient objects. Without semantic merging, text outputs look good, but it tends to over-segment.

rect types of objects, such as 'uber' or 'military'. Most of our score-2 labels are nearly accurate, but the segments may incompletely or excessively cover the referred objects, such as 'lush moss' and 'few puppies' (row 2). Most of our score-3 labels accurately represent their segments, such as 'plane', and they can be descriptive even on background objects, such as 'sandy beach' and 'crowd observing', unlike labels from traditional segmentation methods.

5.3. Ablation study

We compare our method with alternative attention-masking methods, which include 1) cropping and masking the input image to fit the segment region [36], 2) our method without global subtraction, and 3) our method without the

Figure 6. Examples of segmentation results that were evaluated by human evaluators with scores ranging from 3 (correct) to 0 (incorrect). The definitions of the scores can be found in Appendix F.

merging step. Note that we also apply the same merging in the crop-and-mask baseline for a fair comparison. We show the results in Figure 5 and in Table 3.

In Figure 5, the crop-and-mask baseline often returns text labels that are unrelated to the segments, like 'video' in column 2-4. Without global subtraction, our method often fails to recognize objects in the background due to the leak of global contexts. For example, the sky in column 1 is labeled 'tower', and almost everything in column 2 is labeled 'room'. Our full pipeline yields reasonable results, and can label 'ladder', 'lamp', 'floor', and 'reader' correctly.

In Table 3, we report results based on ST reassignment and constant thresholding. Our method outperforms all alternative masking techniques in terms of IoU, Recall, and Text Generation Quality scores on PC-59. Global subtraction also helps improve both IoU_{CLIP} by 3.0-3.1 points. On PAS-20, our method achieves a slightly lower IoU than not using global subtraction (1.1 lower). Upon inspection, we observe that the 20-class PAS-20 tends to label only a few foreground objects while ignoring much of the background (see Appendix D), and not using global subtraction may preserve the embedding of these few objects better. This bias toward primary objects, however, would not be beneficial if the goal is to discover *all* semantic objects.

5.4. Comparison to zero-shot open-vocab baseline

As a reference, we provide a comparison with GroupVit [34], which solves a related but different segmentation problem, *open-vocabulary segmentation*. This task requires text queries to specify which objects to segment, although the queries can be arbitrary or unseen during training. Our method, on the contrary, predicts arbitrary text labels at inference time and is not directly comparable using the same standard benchmarks. Nonetheless, our proposed relabeling procedure can allow useful comparative analysis against open-vocabulary baselines on the same benchmarks.

Table 3. Ablation study of our CLIP's mask attention technique. IoU and Recall are computed with ST and constant thresholding.

Method	PC-59			PC-459	PAS-20
	IoU_c	Recall_c	TGQ	IoU_c	IoU_c
Crop and Mask	12.1	10.2	16.9	5.4	14.0
Ours w/o glob sub.	14.5	15.0	11.8	7.2	21.2
Ours w/o merge	16.4	11.8	-	10.2	18.3
Ours	17.5	15.0	19.0	11.3	20.1

Table 4. Comparison to GroupVit [34], which solves a related but different segmentation problem and requires input text queries. *denotes scores computed on 1,000 random test images. IoU_c and IoU_h are IoU with constant thresholding or human verification. GroupVit's numbers have been updated according to [35]

Method	PC-59			PC-459	PAS-20
	IoU_c	$\text{IoU}_{\text{h}\geq 2}$	$\text{IoU}_{\text{h}\geq 1}$	IoU_c	IoU_c
GroupVit [34]	25.9	-	-	4.9	50.7
Ours	19.6	20.9*	22.7*	11.3	20.1

Table 4 shows that GroupVit obtains a better IoU on PAS-20 with 20 classes. However, our method is significantly narrowing the gap on PC-59, especially with human-threshold IoU, and our IoU with constant-threshold even surpasses GroupVit's on challenging PC-459, which has much more classes (459). Figure 7 shows that our method can discover more objects and provide more fine-grained labeling, while GroupVit labels only a few objects and does not label every part of the image.

5.5. Comparison to supervised baselines

Table 5 presents an IoU comparison with existing supervised open-vocabulary baselines on three datasets based on the numbers presented in [21]. Unlike our approach, these methods require segmentation annotations (or pretrained segmentation models) during training and text queries to guide the segmentation. Our IoU scores in Table 5 are computed using segment-to-text IoU with a constant threshold $\tau_{\text{CLIP}} = 0.1$ and human score ≥ 1. There is still a gap in performance between our unsupervised method and these supervised baselines, though our method performs only slightly worse on the more challenging PC-459.

6. Discussion and Analysis

Mismatched text labels during evaluation. Evaluation in our new setup is still challenging, despite using label reassignment. For example, in Figure 8, our algorithm breaks down the 'building' ground-truth segment into 'roof' and 'pub', which are correct. But ST reassignment assigns 'pub' to 'sign', which is still technically correct but not counted toward our IoU score for 'building'. Another problematic

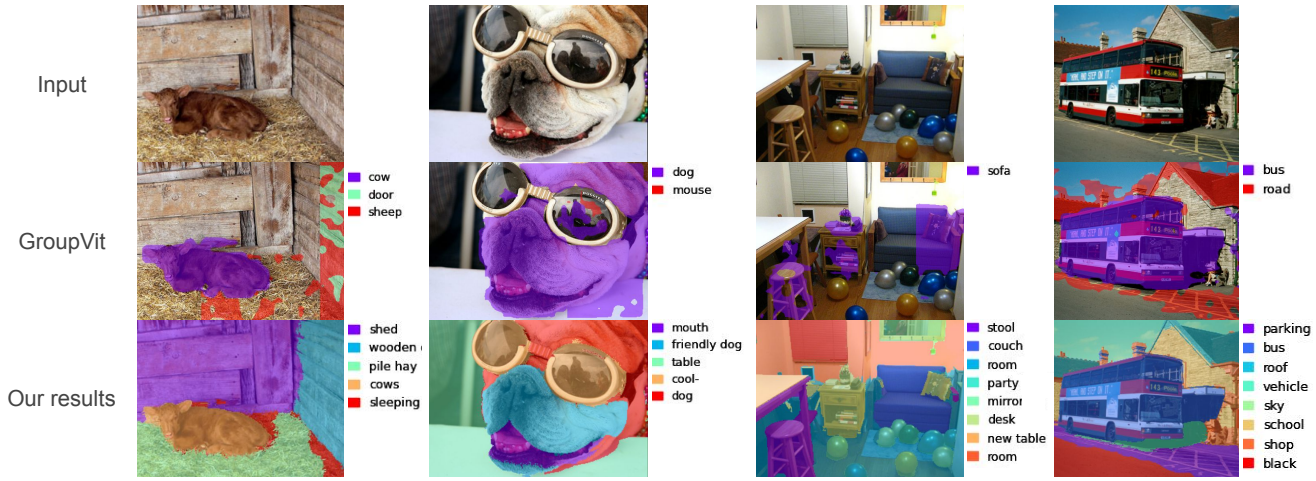

Figure 7. **Qualitative comparison to GroupVit [34].** Despite achieving lower IoU scores, our method can discover objects beyond the labels in the dataset, such as 'hay' and 'mirror', and can provide more fine-grained labels, such as 'stool'.

Table 5. IOU scores comparison between supervised open-vocabulary segmentation baselines (trained with segmentation labels) and our unsupervised method.

Method	PAS-20	PC-59	PC-459
Lseg [19]	47.4	-	-
SimBaseline [36]	74.5	-	-
ZegFormer [8]	80.7	-	-
OpenSeg [11]	-	42.1	9.0
OVSeg [21]	94.5	55.7	12.4
Ours - IoU_c	20.1	19.6	11.3
Ours - $IoU_{h \geq 1}$	-	22.7	-

Figure 8. **Label reassignment issue.** Our predicted labels 'roof' and 'pub' are correct but are not matched to the ground-truth class 'building' during label reassignment.

class is 'person' whose parts like 'face', 'hair', 'shirt' appear distinct in CLIP's space and may not be mapped to 'person' (Figure 9). To overcome this challenge, we may need a new kind of embedding space that understands the hierarchical nature of object parts.

Global context leakage. Some background segments that share boundaries with primary objects can be mislabeled due to the influence of global contexts as shown in Figure 9. Another problem that can cause context leakage is the low-resolution 24x24 image grid of CLIP visual tokens. As we downsample our segment masks to fit this grid, we lose masking precision and information can leak between neighboring segments.

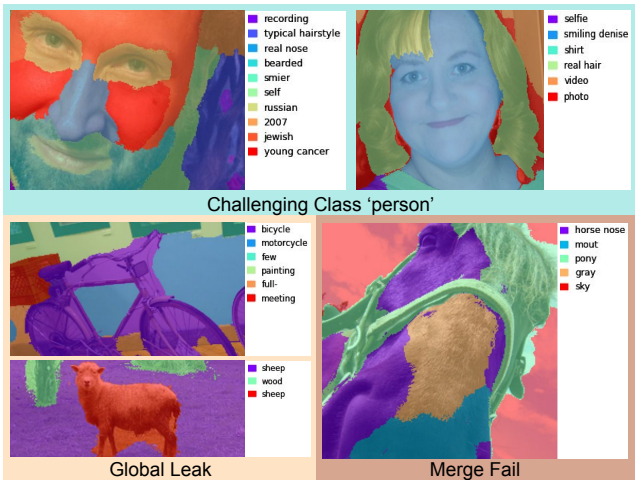

Figure 9. **Failure cases.** 1) Classes that have many visually distinct parts, such as 'person', are difficult to reassign labels correctly. 2) Background regions that share boundaries with salient objects are still prone to *global context leakage*. 3) Semantic merging may fail when the text outputs of the same object give different descriptions.

Merge fail due to different labels of the same object. Our over-segment outputs may use a wide variety of descriptions for the same object, such as car model and car color. These segments may fail to merge into a single object during the merging step (see Figure 9).

Conclusion. We have presented the first framework for *zero-guidance segmentation*, a novel problem that seeks to segment and label an image using natural language automatically. We leverage two generalist models, DINO and CLIP, and propose a technique to map a given segment to CLIP's joint space by masking CLIP's attention, allowing zero-shot segmentation without the need for any segmentation dataset or fine-tuning. We also introduce a new evaluation protocol for this problem and will release our code.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[3] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semantic textual similarity-multilingual and cross-lingual focused evaluation. In *Proceedings of the 2017 SEMVAL International Workshop on Semantic Evaluation (2017). https://doi. org/10.18653/v1/s17-2001*, 2017.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 06 2016.

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

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

[7] Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International conference on pattern recognition (ICPR)*, pages 5482–5487. IEEE, 2021.

[8] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[10] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[11] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021.

[12] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012.

[13] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Vision transformers with patch diversification. *arXiv preprint arXiv:2104.12753*, 2021.

[14] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022.

[15] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[16] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[17] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022.

[18] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

[19] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[20] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pretrained language models. *arXiv preprint arXiv:2011.05864*, 2020.

[21] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. *arXiv preprint arXiv:2210.04150*, 2022.

[22] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

[23] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[24] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan 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*, pages 891–898, 2014.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[26] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[27] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[28] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

[29] Oriane Sim'eoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick P'erez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021.

[30] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[33] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, 2022.

[34] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.

[35] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.

[36] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. *arXiv preprint arXiv:2112.14757*, 2021.

[37] Fuzhao Xue, Jianghai Chen, Aixin Sun, Xiaozhe Ren, Zangwei Zheng, Xiaoxin He, Xin Jiang, and Yang You. Deeper vs wider: A revisit of transformer configuration. *arXiv preprint arXiv:2205.10505*, 2022.

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

[39] Chong Zhou, Chen Change Loy, and Bo Dai. Denseclip: Extract free dense labels from clip. *arXiv preprint arXiv:2112.01071*, 2021.

[40] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[41] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.