Exploring Semantic Coexistentence in Open-Vocabulary Segmentation

Jianhong Tu Erdong Chen Shuhan Zhang Ben Mueller

Abstract

Semantic segmentation is essential in real-world applications such as autonomous driving. Open-vocabulary semantic Segmentation models advance conventional methods by extending pixel label classes to any arbitrary sets of text labels. One recent two-stage approach relies on a class-agnostic mask model and a pretrained vision-language classifier to assign a text label to each mask proposal flexibly. However, this pipeline fails to consider the implicit co-existing relationship of the segmentation targets for more accurate segmentation. This paper proposes CoSeg, a novel open-vocabulary segmentation modeling framework with a pretrained vision-language model and referral segmentation architecture that exploits semantic coexistence in the joint visual-linguistic space. Despite a simple architecture, our no-training and fully-supervised models achieve competitive performance in a cross-dataset evaluation, especially in a contextually rich environment. We believe our method establishes a foundation for future exploration of semantic modeling in images.

1. Introduction

Conventional semantic segmentation models [14, 19] are typically designed for datasets with a rigid class definition having no more than hundreds of labels. The sheer number of possible text labels with potentially ambiguous semantics presents a significant challenge to collecting large-scale, noiseless human annotations. In contrast, as a zero-shot learning task, open-vocabulary semantic segmentation entails a model capable of learning a generalizable strategy to accurately categorize every pixel using an arbitrary set of natural language labels.

Learning high-level transferable visual embeddings is enabled by the recent development of weakly supervised vision-language models, such as CLIP [10], which projects corresponding pairs of image and text descriptions into similar vector presentations in a joint embedding space. However, direct application of CLIP in open-vocabulary segmentation is infeasible since CLIP models are trained to encode the entire image input, whereas segmentation requires fine-grained region-level representations. Previous works [11, 23] bridge this granularity gap by decoupling mask generation and label generation: a two-stage pipeline is proposed to employ a pretrained CLIP model [10] to classify a set of task-agnostic mask proposals generated by a separate finetuned fixed-class segmentation model. A limitation of this two-stage approach is the failure to account for the interaction between semantic entities, compromising the label assignment accuracy.

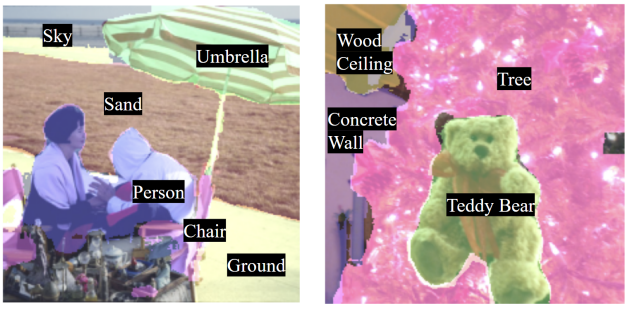

Figure 1. *CoSeg-Prefix* Results on 2 MS-COCO Images. The model is trained to segment common objects of interest in public segmentation benchmarks.

This work is inspired by the Distribution Semantic Theory, which originates in Natural Language Processing and states that entities that appear in similar contexts carry similar semantic meanings. In the context of semantic segmentation, we argue that segmentation masks of objects commonly appearing together in the same image are comparable to the linguistic tokens whose semantics can be efficiently learned from the co-occurring tokens [16], namely the "context." Intuitively, every image captured in the wild implicitly defines a broader context for the observed objects. When detecting all objects in the scene, reasoning about the setting associated with a particular object is potentially helpful for narrowing down the range of expected objects. For example, one would expect a "beach" seen in the outdoor scene to more likely co-exist with the "sky" and the "sea" than a "sofa" that often appears in the indoor scene. Yet learning such semantic relationships directly from segmentation masks is not trivial, for it introduces another problem: how to represent the masks efficiently? Therefore, we propose exploring coexistence information in the visual-linguistic space with a vision-language model, where semantic coex-

Figure 2. Qualitative Comparison of CoSeg. Our model is capable of segmenting both foreground and background pixels.

istence may be tractable from linguistic tokens.

In this paper, we present a new open-vocabulary segmentation framework (CoSeg) by exploiting semantic coexistence in the text-image domain to accurately predict a diverse set of segmentation masks with labels from an arbitrary set. Two major pipeline components are a vision-language model that predicts segmentation targets and a conditional mask generation model that outputs a sequence of binary masks accordingly. The modular design allows for flexible adaptation of wider architecture choices. In the scope of this study, we focus on the popular CLIP vision-language model and a simple transformer-based UNet-like segmentation architecture.

Following the zero-shot evaluation protocol as [8], we evaluate our method in a cross-dataset setting. We implemented a simple *no-training* baseline method with off-the-shelf pretrained models and also trained two variants of the proposed model with mask and text supervision. Our models achieve competitive results in benchmarks with rich contexts. We provide evidence that contextual information benefits multi-target segmentation. Our method also offers an intuitive interpretation of the model's behavior. Our research serves as a baseline for future efforts dedicated to models that more effectively use coexistence relations to learn image semantics.

2. Related Work

Semi-Supervised Large-Scale Vision-Language Pretraining Weakly supervised vision-language models are proposed to learn a general representation of images and texts in a unified embedding space. Modern efforts to this end [1, 20] adopt frozen, pretrained, transformer-based language models and project images into the linguistic space with vector quantization or pretrained vision encoder, such as ResNet [5] and Vision Transformer [3], with a lightweight multiplayer perceptron adapter network. The latest works, represented by CLIP [10] and ALIGN [6], have developed models capable of open-vocabulary classification by end-to-end pretraining vision-language models on large-scale internet datasets. The impressive generalizability of these models makes them a good entry point for building networks that solve a wide range of cross-modal downstream tasks, including vision question answering [24] and large vision-language model assistant [9].

Our proposed pipeline is also based on the public CLIP checkpoints and involves the CLIP contrastive learning paradigm to extend its object recognition ability to multi-target semantic segmentation.

Referral Segmentation Model As another branch of the open-vocabulary task, referral segmentation is a one-shot dense label prediction task that conditions its mask generation on a provided natural language phrase. Previously proposed models [7, 15] also rely on pretrained vision-language models to align image and text modalities. In particular, CLIPSeg [15] offers a purely transformer-based conditional mask decoder capable of generalizing to out-of-distribution classes and even abstract text descriptions. Note that past referral segmentation benchmarks do not require zero-shot ability, making the CLIPSeg decoder desirable for our task since new unseen categories will be introduced in the testing phase.

Open-Vocabulary Semantic Segmentation Recent approaches for open-vocabulary semantic segmentation take advantage of pretrained vision-language models that align images to short phrases mentioned above. One simple strategy effectively turns the image semantic problem into a region-level classification task. ZSSeg [23] adapts a fine-tuned MaskFormer model, which is not capable of zero-shot segmentation, to generate a fixed number of mask proposals without labels and then extract a sub-image for every mask proposal to assign a label using CLIP as an open-vocabulary image classifier. Subsequently, MaskCLIP [11] improves upon the two-stage pipeline by finetuning the CLIP on masked object patches, thus boosting classification accuracy. However, such a method heavily relies on a large mask proposal network, and its capacity for recognizable objects inherently limits the zero-shot ability. In contrast, our one-stage method combines a segmentation network and a classifier into an end-to-end model supervised by both masks and text labels to model semantic coexistence distribution, allowing for better bi-modal alignment.

3. Approach

We introduce CoSeg, a simple open-vocabulary segmentation model based on an open-source vision-language

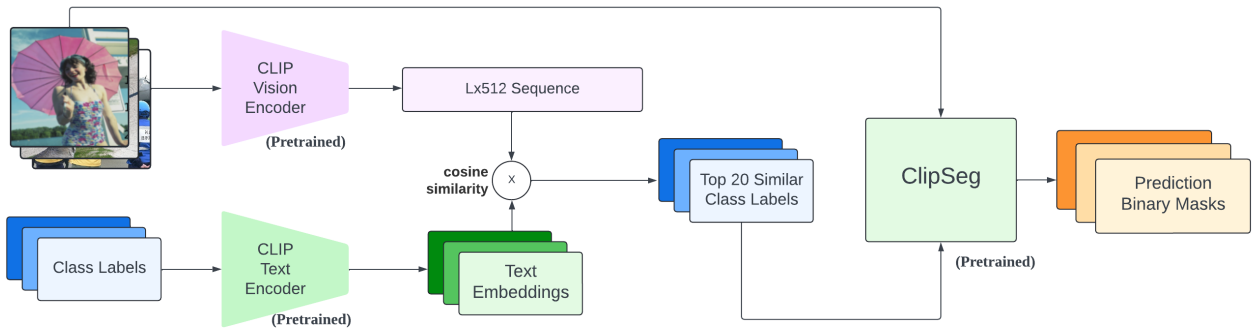

Figure 3. Baseline No-Training Pipeline. A CLIP model computes L image embeddings and pools them to identify the top-k similar labels based on cosine similarity from pairs of one image embedding and all text label embeddings. A CLIPSeg model generates k binary masks by taking the input image and the k labels.

model and segmentation architecture that aims to establish coexistence relationships between segmentation masks. Our method enhances the vision-language model to recognize correlated segmentation targets and leverages a conditional mask generator to produce the corresponding set of binary masks. We explore two distinct vision-language model architectures and offer a *no-training model* with out-of-the-box pretrained models.

3.1. Problem Definition

We mathematically set up the open-vocabulary semantic segmentation task: During training, we have access to the training data D_{train} that contains segmentation annotations y and corresponding text labels $x \in C_{seen}$. At the inference (test) time, the testing dataset D_{test} has ground truth masks with labels from an arbitrary set $C_{test} = C_{unseen} \bigcup C_{seen}$, where C_{unseen} and C_{seen} are mutually exclusive. The modeling goal is to learn a function f that maps a normalized image $I \in [0,1]^{H \times W}$ to the a sequence of predicted binary masks $\hat{y} = \{0,1\}^{N \times H \times W}$ and corresponding labels $[\hat{x_1}...\hat{x_N}]$ such that $\forall i \; \hat{x_i} \in C_{test}$, where H and W denote the fixed height and width. A successful model f should be able to predict masks for unseen classes with non-trivial accuracy.

3.2. CLLPSeg Decoder

Both no-training and main architecture build upon the CLIPSeg decoder [15], a UNet-like purely decoder-only transformer-based conditional mask generator with stacked transposed convolutional layers. Thus, we briefly discuss its modeling choice. The decoder network selects M intermediate transformer layers of CLIP vision encoder to pull out sequences of activations $A_{clip} \in \mathbb{R}^{L \times d_{clip}}$, dropping the "[CLS]" token. The network architecture consists of M transformer decoder blocks and M fully connected projection layers to compress the activations to a lower dimension $d_{reduce} << d_{clip}$.

$$A_{reduce} = f_{reduce}(A_{clip}) \in \mathbb{R}^{L \times d_{reduce}}$$

where f_{reduce} is applied to every row and d_{reduce} is the CLIPSeg decoder's hidden dimension. The first input activation $A_{dec,0}$ is simply initialized to a 0 sequence. At each transformer block, we sum up the last decoder activation $A_{dec,i}$ and the i-th reduced CLIP activation $A_{reduce,i}$ and feed the result into the transformer block to compute the intermediate activation $A_{int,i+1}$:

$$A_{int,i+1} = Transformer(A_{dec,i} + A_{reduce,i})$$

To enable conditional generation, the CLIPSeg decoder incorporates a Feature-wise Linear Modulation layer [18], featuring two affine transformations as functions of x, the CLIP token to condition on:

$$\gamma = f_{mul}(x) \in \mathbb{R}^{d_{reduce}} \qquad \beta = f_{add}(x) \in \mathbb{R}^{d_{reduce}}$$

The two FiLM values above modulate each decoder block's intermediate activation to obtain the final activation:

$$A_{dec,i+1} = FiLM(A_{int,i+1}, \gamma, \beta) = A_{int,i+1} * \gamma + \beta$$

where $*$ and $+$ indicate element-wise multiplication and addition applied to every row. At the end of the forward pass, the final activation is reshaped into a tensor of shape $\frac{H}{16} \times \frac{W}{16} \times d_{reduce}$, assuming the CLIP vision encoder computes patch embeddings of size 16×16, and then fed into a simple stacked transposed convolutional network to generate a binary image of shape $H \times W$ as desired.

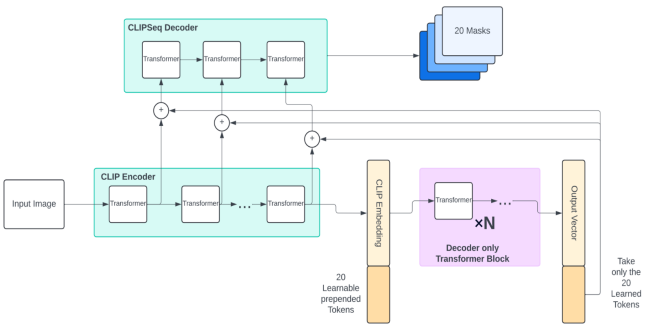

Figure 4. CoSeg (Prefix) Architecture. A prefix vision-language model prepends L learnable tokens, which are used to predict L CLIP embeddings as segmentation targets after the forward pass. A retrained CLIPSeg decoder produces L masks conditioned on the CLIP embeddings.

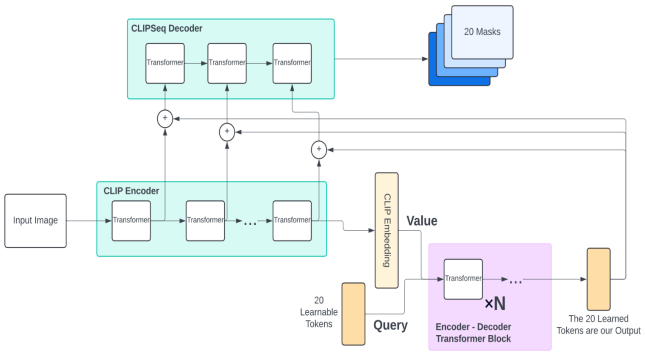

Figure 5. CoSeg (Cross-attention) Architecture. The vision-language model encodes the final activations of a CLIP vision encoder and decodes L clip embeddings using L learnable tokens and the cross-attention mechanism. A retrained CLIPSeg decoder generates L masks in the same way as the prefix model.

3.3. No-training Baseline Model

One approach we will be using for baseline builds upon pretrained Clip [10] and pretrained CLIPSeg [15]. We utilize a two-stage pipeline to perform image segmentation without additional training.

Feature Extraction Clip text encoder processes a short textual input (in our case, the image class labels). The tokenized prompt sequences are wrapped by [SOS] and [EOS] and passed through Transformer [21] layers to obtain contextualized, high-level representations. To get the final representation of the prompt, we pool the last activations by extracting the text embedding at the [EOS] token position.

Clip vision encoder tokenizes input images by converting them into a grid of image patches with a convolution layer, and image patches are flattened into a high-dimensional embedding space. [CLS] pooling adds a learnable embedding token and takes the Transformer [21] predictions of [CLS] token only as the representation of the whole image sequence.

Cosine Similarity Cosine similarity is computed between the normalized embedding of text and image to reflect how closely the image content matches the semantic meaning of the text. In our No-training setting, for each image in the COCO-Stuff training dataset [2], the encoded representation is compared with the text embedding of all class labels.

$$\text{Cosine Similarity}(x, y) = \frac{x \cdot y}{||x||||y||}$$

where x represent the image embedding, and y is one of the all class labels.

Top-k sampling We hypothesize the model has learned the delicate semantic meanings of diverse images and prompts during CLIP pretraining on internet image-caption data. Thus, the semantic coexistence information can be readily extracted from the pooled representation. Specifically, we selected labels with the highest top-k cosine similarities by comparing the predicted embedding of an image against the embeddings of all relevant labels to identify the segmentation target. The choice of k in top-k sampling depends on the distribution of ground truth class number per image for different segmentation datasets. The chosen top-k labels are saved for subsequent mask generation.

Pertained CLIPSeg The CLIPSeg model [15] can produce a single binary mask conditioned on an arbitrary text prompt. During inference, we predict a sequence of masks conditioned on the given k most probable class labels in an iterative pattern.

3.4. CoSeg

CoSeg concatenates a vision-language model to a retrained ClipSeg decoder. The former module is responsible for identifying up to N segmentation targets represented by CLIP tokens, and the latter generates N binary masks conditioning on each of the tokens. Both modules share one pretrained CLIP vision encoder to reduce the computation cost. This section focuses on the design of the language model component. To model the coexistence relationship in images, we enhance the CLIP model with additional transformer blocks and introduce an additional embedding layer with N learnable tokens as an extension of the "[CLS]" token used for image-level classification. We experiment with 2 variants of the language model as illustrated in Figure 4 and 5.

In the forward pass, the model first passes the image through the CLIP vision encoder and the visual projection layer to obtain L CLIP tokens. Similarly, the "[CLS]" token is dropped. Without applying any attention mask, we allow the new tokens to freely attend to any CLIP tokens as well as each other and to learn 1) identifying distinct semantic

entities in the scene, including a special "[unlabled]" token, and 2) finding the most probable set of semantic entities according to the mutual information. The modeling strategy conceptually resembles a transformer language-only model. Given a normalized image $I \in [0,1]^{W \times H}$, the vision-language model predicts a matrix $X = [x_1, ..., x_n]^T \in \mathbb{R}^{N \times d_{clip}}$ as the detected objects.

However, predicting a set of semantic entities differs from language modeling since the order of the tokens is unimportant. We offer a naive solution by sorting the ground truth labels according to the area of their segmentation masks in decreasing order. This practice attempts to establish an imperfect order for the semantic tokens, making a sequence-to-sequence modeling style more suitable. In the case where there are less than N ground truths, we pad the sequence with empty masks and "[unlabeled]" tokens.

Both encoder-decoder and decoder-only transformers are widely explored in the natural language processing domain. To search for a more effective architecture for our vision-language model, we also design two variants with prefix and cross-attention models that perform the same task.

Prefix Language Model This model employs the decoder-only transformer block, which lacks the encoder network and the cross-attention layer compared to the original transformer architecture [21]. The newly added N learnable tokens are position encoded and pretended to the L CLIP embeddings as prefixes. The entire sequence of $N+L$ tokens is then passed through the self-attention layer and the multilayer perceptron layer in the transformer decoder. Finally, only the N prefix tokens are extracted and outputted as the final prediction.

Cross-Attention Language Model This model employs the vanilla transformer architecture with a full encoder and decoder. The encoders encode the L clip tokens. In the decoder layers, We treat the N new tokens as the $Query$ and L clip tokens as the Key and $Value$, allowing the query tokens to attend to all key tokens freely. The decoders finally output N tokens as the final prediction.

Mask + Text Supervision At the end of the pipeline, the CLIPSeg decoder predicts N masks, and we extract the N language outputs as the label predictions. We intend to use the language outputs to identify the actual text label by computing all-pair cosine-similarity between N predictions and $|C_{label}|$ tokens, similar to CLIP's open-vocabulary classification pipeline [10]. Therefore, before training, we encode all text labels with a CLIP text encoder as a necessary preprocessing step. At the training time, the encoded labels serve as the *anchor* for our contrastive learning objective. We keep the encoded ground truth tokens invariant and maximize the cosine similarity between the matched predicted tokens and the ground truths. The language loss

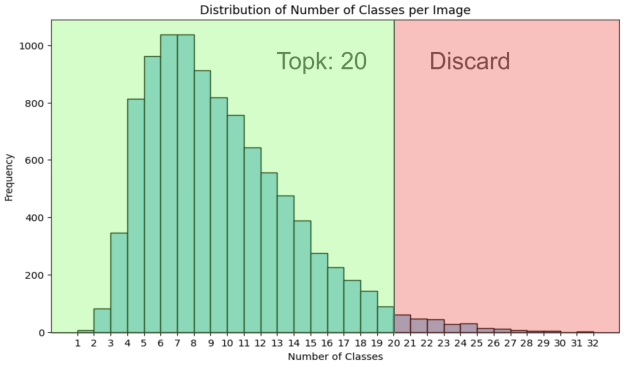

Figure 6. Distribution of the number of classes for each image in the COCO-Stuff training dataset. We set up 2 standard deviations to derive our k for top-k sampling.

value for a sequence of labels is calculated as follows:

$$\mathcal{L}_{lang}(x, \hat{x}) = -\frac{1}{L}\sum_{i=1}^{L} log \frac{exp(\hat{x}_i^T x_i/\tau)}{\sum_{j=1}^{L} exp(\hat{x}_i^T x_j/\tau)}$$

where τ is a hyperparameter and x_i and $\hat{x}_i$ are the l_2 normalized predicted token and ground truth token at the i-th position.

For the mask supervision, we choose the binary cross entropy loss as in [15]:

$$\mathcal{L}_{mask}(y, \hat{y}) = -\frac{1}{M}\sum_{i}^{M} [y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)]$$

where y and $\hat{y}$ are the binary ground truth mask value and predicted probability, respectively, supposing there are M pixel-level predictions. Then, the final loss function is:

$$\mathcal{L}_{all} = \mathcal{L}_{mask} + \lambda \mathcal{L}_{lang}$$

λ is a hyperparameter introduced to balance the two loss values.

3.5. Experiments

In this section, we describe the cross-dataset evaluation protocol and the implementation of the three models: *CoSeg-no-training*, *CoSeg-prefix*, and *CoSeg-xatten*. The training dataset employed in this study is the COCO-stuff dataset [2], which enhances the original COCO dataset [12] with pixel-level annotations for various "stuff" categories. There are 118K contextually rich training images from diverse 80 thing classes and 91 stuff classes. Incorporating foreground and background objects and tens of semantic labels per image makes it ideal for exploring semantic coexistence in context. Following [8], we choose the following benchmarks: COCO validation set [13] with 5000 images and 80 thing classes, Pascal VOC 2012 validation set [4]

Model	COCO	Pascal VOC	Pascal Context 59	Pascal Context 459
MaskCLIP	12.8	26.8	22.8	10.0
GroupViT	27.5	50.8	23.7	-
OpenSeg	38.1	77.2	45.9	-
S-Seg	**81.8**	42.4	27.2	8.7
SAN	-	**94.6**	**57.7**	**12.6**
OVSeg	-	94.5	55.7	11.0
CoSeg (Prefix)	***50.0**	65.5	52.7	**45.5**
CoSeg (Xatten)	*49.3	**67.1**	**54.2**	44.6
CoSeg (No-training)	38.9	61.1	31.2	24.9

Table 1. Comparison of Model Performances. Scores are computed by following the cross-dataset evaluation protocol. Performance of other models is collected from the variants with CLIP-B/16 encoder, if applicable, as reported in [8] and [22]. * indicates that COCO-Stuff is used for training. High scores are better.

with 1449 images and 20 classes, and Pascal Context validation set with 5105 images and 459 classes. Pascal Context also provides a subset of the 59 most common labels for testing. The general "unlabeled" background class is excluded from all datasets. The training dataset, COCO-stuff, already includes all classes in COCO, making it no longer a zero-shot challenge. Nonetheless, we report scores on COCO for completeness. We justify the dataset choices by referencing the similar training/evaluation experiment by [22], who report similarity scores between the training and testing set to support this evaluation protocol.

As common in image segmentation benchmarks, we choose the mean of class-wise intersection over union (mIoU) as the performance metric.

3.6. Baseline Implementation

No-training implementation is divided into two stages: first generating *top-20 similar class labels* for each image, and then outputting image segmentation by predicting a sequence of object masks conditioned on *top-20 similar class labels*.

We choose k to be 20 by analyzing the number of label distributions on the COCO-Stuff dataset. For each image in the dataset, we plot the frequency vs. number of classes, as shown in Figure 6. The figure shows that most images have less than 20 labels. A high value for k likely decreases the true positive rate to detect the ground truth classes at the cost of more CLIPSeg forward passes. Therefore, segmenting at most 20 semantic entities per image avoids the excessive computation burden of redundant classes while capturing most labels.

Our pipeline (Figure 3) starts with class labels from each dataset, where we first attach the prefix "a photo of $<$*class*$>$" to each class label. We then convert modified class labels to sequences of high-dimensional tokens using pretrained Clip [10] text encoder. Meanwhile, the vision encoder converts each image in the dataset into a 512 embedding in the same representation space. We calculate the cosine similarity between the image embedding and text embeddings of all existing class labels for each image, sorting the result with descending cosine similarity. Finally, the top 20 similar class labels are saved with image IDs and 20 similarity scores. In the next stage, we pass the image and top 20 similar class labels pretrained CLIPSeg [15] to derive binary mask predictions.

3.7. Main Model Implementation

We initialize the CLIP-B/16 model from the *huggingface* checkpoint and freeze the pretrained network throughout the training session. As a prepossessing step, we use the standard clip template "a photo of a $<$*label*$>$" to wrap up all 182 labels in the COCO-stuff dataset and compute the pooled CLIP embedding with the pretrained text encoder. Since the image encoder is pretrained on images with a resolution of 224×224, we prepare every training image using center crop to extract a square image and resizing to reduce the spatial resolution to 224×224, maintaining the aspect ratio same as the original image. The ground truth mask is separated into a sequence of binary masks, truncated if containing more than 20 items and padded with "unlabeled" and empty mask if containing less than 20 items.

We trained both *prefix* and *xatten* variants of our model to compare their performance. Both language models are created with 8 additional transformer blocks with 4 multi-attention heads: The *prefix* model has 8 decoders, while the *xatten* model has 4 encoders and 4 decoders. As with the no-training method, we decide the number of new tokens N to be 20 and initialize 20 learnable tokens from the normal distribution. We choose $d_{reduce} = 128$ and extract the activations from the $[3, 6, 9]$ CLIP transformer layer for the CLIPSeg decoder. Both networks are optimized with the Adam optimizer with a cosine learning schedule at an initial learning rate of 1e-4 and a minimum value of 1e-6, a batch size of 32, and a total of 100 epochs. We use $\tau = 0.08$, same as CLIP, and λ = 0.08. We threshold the per-pixel probability at 0.3 for mask generation.

4. Results

We report the performance of our three models and a few other state-of-the-art models in Tab. 1. Due to a shortage of time, we are unable to re-implement the other methods in our environment. Rather, we compare our scores to those reported in [22] and [8]. The baseline training method performs reasonably, beating MaskCLIP, the two-stage method. Our fully supervised models yield competitive scores, especially on the Pascal Context [17], which features more contextually rich annotations than Pascal VOC [4]. It is worth noting that our *CoSeg-Xatten* and *CoSeg-Prefix* variants fall behind the SOTA method by -3.5 MIoU in Pascal Context with 59 classes but surpass SOTA by a large margin of +32.9 mIoU. Across all settings, our methods show a more robust performance, though the peak performance is not as strong. We argue that our method avoids overfitting and is stable against perturbation in the labels, thus having a more generalization capability. However, a more detailed investigation of the exact behavior of each model is needed to provide more evidence to support the claim.

5. Discussion and Conclusions

In this work, we contribute the CoSeg, an intuitive framework to build robust Open-Vocabulary semantic segmentation models by leveraging the frozen CLIP model and a referral segmentation decoder. Our method shows consistent performance compared to previous methods and is insensitive to label variance. The main models are trained end-to-end to capture semantic coexistence in the CLIP embedding space to the greatest extent. The better performance over the no-training baseline with the weakly supervised CLIP classifier and CLIPSeg segmentation model provides evidence that introducing context is beneficial for segmentation tasks involving multiple targets.

However, we acknowledge that the reported performance scores are not collected strictly in the same environment, despite our effort to follow the previous evaluation protocol as closely as possible. Another deficiency is the presence of prediction bias towards background classes. Our training procedure involves sorting the ground truth masks according to their relative sizes, but the mask decoder is blind to this information. In future work, we plan to explore better modeling options to inject the "size" information into the decoder and replace the simple binary cross entropy loss with the dice loss. Additionally, our study entails a comprehensive ablation study to support that our method is capable of modeling semantic coexistence, and segmentation effectively benefits from such information.

6. Statement of Individual Contribution

Jianhong(Tovi) Tu As the project Lead, Tovi initiated the idea of enhancing image segmentation performance by integrating linguistic semantic meanings. He proposed the thesis and designed the Coseg model. He actively explored multiple architecture designs, including the initial autoregressive vision-language model, which was proven to perform poorly in empirical evaluation. Tovi managed the model-building process and led the final model training sessions using Lab GPUs. His leadership and technical experiences were pivotal in guiding the project's direction and integrating teamwork.

Erdong(Anthony) Chen Anthony contributed significantly by managing the initial model training on a smaller scale using personal GPUs in early-stage testing development. He worked closely with Tovi to explore the applicability of the CLIP architecture for our project. Additionally, Anthony implemented the baseline model, which operates without further training. The baseline model demonstrated the feasibility of our hypothesis.

Shuhan(Steven) Zhang Steven served as the head of the data team in this project. He compiled a list of datasets and evaluation methods by reading and researching other papers. Subsequently, Steven developed Python code for the Data Loader pipeline for each dataset. Upon completion of training, he executed scripts to generate qualitative evaluation graphics for both the trained and untrained models for all datasets used.

Ben Mueller Ben worked with Steven on the data team for this project. He aided in researching, implementing, and debugging evaluation methods for the various models and datasets while also working on the data loading pipelines. After the training was completed, he worked with Steven to write and debug scripts to generate needed evaluation graphics for various combinations of our models and datasets.

Group coordination In the early stages of the project, we set up in-person meetings in Olin Library on a weekly basis. Near the milestone, presentation, and paper due, we had additional group meetings to ensure everything went smoothly.

7. External Resources Used

This project builds upon several open-source libraries. We hereby explain their functions and provide external references.

PyTorch Deep Learning Framework All deep learning models are implemented with Python PyTorch library of version 2.0.0 https://pytorch.org/. We utilize the provided data utilities to streamline data loading and preprocessing, the torch-vision toolkit for image augmentation, and the "nn" package for modeling building and automatic

gradient calculation.

HuggingFace Checkpoints HuggingFace https://huggingface.co/ is an open-sourced platform that hosts a large number of models. We rely on its Python package to access pretrained model checkpoints. Specifically, we use the clip-vit-base-patch16 variant at https://huggingface.co/openai/clip-vit-base-patch16 and the clipseg-rd64-refined model at https://huggingface.co/CIDAS/clipseg-rd64-refined.

Training & Testing Data We access COCO-Stuff at https://github.com/nightrome/cocostuff, Pascal VOC 2012 at http://host.robots.ox.ac.uk/pascal/VOC/, and Pascal Context at https://cs.stanford.edu/~roozbeh/pascal-context/.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 2

[2] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1209–1218. Computer Vision Foundation / IEEE Computer Society, 2018. 4, 5

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2

[4] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. 5, 7

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 2

[6] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 2021. 2

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3992–4003. IEEE, 2023. 2

[8] Zihang Lai. Exploring simple open-vocabulary semantic segmentation. *CoRR*, abs/2401.12217, 2024. 2, 5, 6, 7

[9] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 2

[10] Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. Clip-event: Connecting text and images with event structures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16399–16408. IEEE, 2022. 1, 2, 4, 5, 6

[11] 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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 7061–7070. IEEE, 2023. 1, 2

[12] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 5

[13] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 5

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. 1

[15] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7076–7086, 2022. 2, 3, 4, 5, 6

[16] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. 1

[17] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. The role of context for object detection and semantic segmentation in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 891–898. IEEE Computer Society, 2014. 7

[18] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3942–3951. AAAI Press, 2018. 3

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015. 1

[20] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7463–7472. IEEE, 2019. 2

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. 4, 5

[22] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 2945–2954. IEEE, 2023. 6, 7

[23] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXIX*, volume 13689 of *Lecture Notes in Computer Science*, pages 736–753. Springer, 2022. 1, 2

[24] Ziyi Yin, Muchao Ye, Tianrong Zhang, Jiaqi Wang, Han Liu, Jinghui Chen, Ting Wang, and Fenglong Ma. Vqattack: Transferable adversarial attacks on visual question answering via pre-trained models. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 6755–6763. AAAI Press, 2024. 2

A. Appendix