USE: Universal Segment Embeddings for Open-Vocabulary Image Segmentation

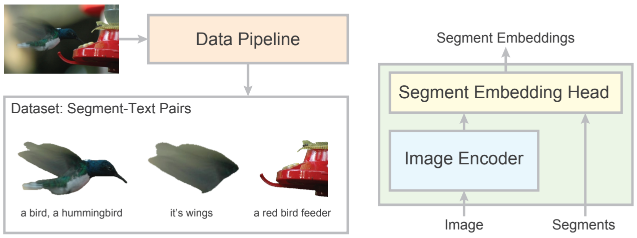
Xiaoqi Wang[1,2,3] Wenbin He[1,2] Xiwei Xuan[1,2,4] Clint Sebastian[2] Jorge Piazentin Ono[1,2]
Xin Li[1,2] Sima Behpour[1,2] Thang Doan[1,2] Liang Gou[1,2] Han-Wei Shen[3] Liu Ren[1,2]
[1]Bosch Research North America [2]Bosch Center for Artificial Intelligence (BCAI)
[3]The Ohio State University [4]University of California Davis
wang.5502@osu.edu wenbin.he2@us.bosch.com xwxuan@ucdavis.edu, clint.sebastian@de.bosch.com
{jorge.piazentinono, xin.li9, sima.behpour, thang.doan, liang.gou}@us.bosch.com
shen.94@osu.edu liu.ren@us.bosch.com

Abstract

The open-vocabulary image segmentation task involves partitioning images into semantically meaningful segments and classifying them with flexible text-defined categories. The recent vision-based foundation models such as the Segment Anything Model (SAM) have shown superior performance in generating class-agnostic image segments. The main challenge in open-vocabulary image segmentation now lies in accurately classifying these segments into text-defined categories. In this paper, we introduce the Universal Segment Embedding (USE) framework to address this challenge. This framework is comprised of two key components: 1) ***a data pipeline*** *designed to efficiently curate a large amount of segment-text pairs at various granularities, and 2)* ***a universal segment embedding model*** *that enables precise segment classification into a vast range of text-defined categories. The USE model can not only help open-vocabulary image segmentation but also facilitate other downstream tasks (e.g., querying and ranking). Through comprehensive experimental studies on semantic segmentation and part segmentation benchmarks, we demonstrate that the USE framework outperforms state-of-the-art open-vocabulary segmentation methods.*

1. Introduction

Open-vocabulary image segmentation [7, 20, 35, 36] aims to partition images into semantically meaningful segments and classify them with arbitrary classes defined by texts. Recent advances in vision foundation models such as the Segment Anything Model (SAM) [15] have shown superior performance in grouping image pixels into semantically meaningful segments at various granularities (e.g., object, part, and subpart). However, the existing open-vocabulary image segmentation methods [7, 20, 35, 36] face challenges in fully utilizing image segments generated by foundation models. For instance, end-to-end methods such as side adapter network (SAN) [36] cannot take image segments generated by foundation models as input or prompt to assign class labels. While two-stage methods (e.g., OVSeg [20]) decouple the image segmentation and classification, they are still limited in classifying segments at various granularities because of limited human annotations [2].



a) USE data pipeline b) USE model

Figure 1. The proposed framework consists of two major components: a) data pipeline that generates segment-text pairs from image datasets and b) segment embedding model.

In our study, we introduce a Universal Segment Embedding (USE) framework to tackle the identified challenges. The goal of USE is to take an image and various segments as input and generate an embedding vector for each segment that aligns with its corresponding text descriptions. These segment embeddings can then be utilized for classifying the segments in a zero-shot manner, similar to the CLIP [29] model used for image classification. Inspired by the recent advances in vision-language foundation models [21, 28], we develop the USE framework with a data-centric approach. Specifically, we introduce a **data pipeline** (Figure 1a) designed to autonomously generate segment-text pairs at various granularities without human annotations. In addition, we develop a lightweight universal segment embedding **model** (Figure 1b) that can be trained efficiently on the large scale of segment-text pairs.

Data Pipeline. Training data with a large scale of high-quality segment-text pairs plays an indispensable role in achieving a high-performing USE model. Inspired by the foundation model-powered data-centric approaches [28], we build a data pipeline that leverages a set of vision or vision-language foundation models to extract segment-text pairs from unlabeled images. Given an image, our data pipeline starts by generating detailed descriptions of the objects and parts in the image with a Multimodal Large Language Model (MLLM) [32]. Then, we detect the most relevant bounding box for each object/part with a phrase grounding model [22]. In the end, the segments of the objects and parts are generated based on the bounding boxes to collect segment-text pairs.

Model. We develop the USE model by leveraging the capabilities of pre-trained foundation models with *minimal trainable parameters*. The USE model consists of two major components, including an image encoder that is adapted from pre-trained vision foundation models and a lightweight segment embedding head that generates segment embeddings for input segments. Note that the output of the image encoder can be reused with different segments, and the lightweight segment embedding head can generate embeddings efficiently.

We conducted extensive experiments on open-vocabulary semantic segmentation and part segmentation benchmarks to demonstrate the advances of the proposed data pipeline and model. Our framework not only achieves state-of-the-art performance but also has flexibility in handling different open-vocabulary recognition tasks.

In summary, the contributions of this paper are threefold:

- We propose a carefully designed data pipeline that can autonomously generate high-quality segment-text pairs at various granularities without human annotations.
- We propose a lightweight segment embedding model that can generate high-quality segment embeddings, which are well-aligned with text descriptions. Hence, it enables various zero-shot image segmentation tasks such as semantic, instance, and part segmentation. In addition, the embeddings offer efficient querying of image segments by text.
- Substantial performance improvements are observed with our approach over the state-of-the-art open-vocabulary image segmentation methods on different tasks including semantic and part segmentation.

2. Related Work

Multi-Modality Representation Learning. Recently, learning from large-scale image-text data (e.g., CLIP [29]) has shown promising results in connecting visual concepts with textual descriptions. Pre-trained CLIP [29] has endowed many computer vision tasks with the capability of open-vocabulary recognition by learning a joint representation of image and text. These computer vision tasks include but are not limited to image segmentation [10, 20, 34, 36], object detection [8, 37], and image captioning [24]. However, the multi-modality representation learning for segment-text data is still under-explored with very few existing work [20, 34]. OVSeg [20] proposes a mask-adapted CLIP that fine-tunes CLIP on a collection of masked image regions to produce mask-aware image embeddings. Unfortunately, OVSeg fails to connect rich semantic information, such as object attributes, with the masked regions. It also has the limitation that the background information outside the masked region is completely ignored during the generation of segment embeddings. Unlike OVseg, the USE model can learn more expressive segment embeddings enriched with detailed text descriptions, including color, shape, size, etc. In addition, the segment embeddings generated by the USE model will take the context information outside the masked region into account given the detailed text descriptions.

Open-Vocabulary Image Segmentation. Driven by the increasing demands of real-world visual tasks, such as autonomous driving, the significance of open-vocabulary image segmentation is growing rapidly. The existing methods can be classified into two categories: end-to-end approaches [33, 34, 36, 40] and two-stage approaches [5, 12, 20, 35]. The two-stage approaches first generate class-agnostic segment proposals and then classify segments into text-defined categories, whereas the end-to-end approaches often generate class-specific segments in an end-to-end manner. Our approach aligns with the two-stage paradigm. Compared with the previous two-stage methods, our approach can take segments of various granularities as input and generate the corresponding embeddings. Meanwhile, we propose a foundation model-powered data pipeline to generate a large scale of segment-text pairs, which enhances the zero-shot ability of our model.

Improving Image-Text Datasets. The careful curation of high-quality image-text pairs is the secret sauce behind the remarkable performance of large-scale pre-trained multimodal models like CLIP [29]. Inspired by this observation, researchers have recently conducted extensive research on improving the quality of image-text datasets, which can further improve the performance of open-vocabulary computer vision tasks. The existing work can be categorized into two classes: data filtering [3, 23] and data improvement [6, 18, 39]. Data filtering aims to improve the efficiency and robustness of model training by filtering noisy image-text pairs, while data improvement focuses on improving the alignment of image and text data. In order to avoid filtering out images with rich visual concepts, we designed a data improvement approach as part of our data pipeline. Similar to [39], we leverage MLLMs to infuse more informative visual concepts into image captions. Furthermore, we propose to augment the image captions by

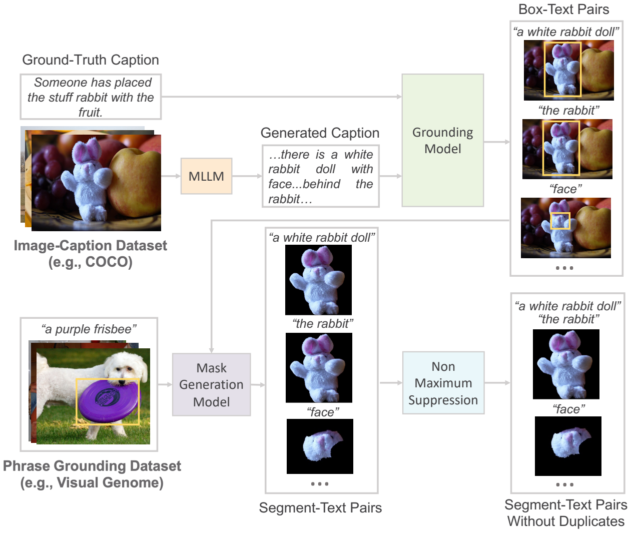


Figure 2. The overview of our data pipeline, which automatically constructs segment-text pairs at different levels of granularity. We design a unified data pipeline that curates data from different types of data sources while taking advantage of multiple foundation models to streamline the process.

meticulously describing the parts of objects in the image, thereby enriching the semantics of captions at multiple levels of granularity.

3. Method

In this paper, we propose a novel open-vocabulary image segmentation framework, USE, which consists of two key components: a data pipeline (Section 3.1) and a universal segment embedding model (Section 3.2). Specifically, the data pipeline aims to automatically curate large-scale segment-text pairs with fine-grained object descriptions at multiple levels of granularity; the universal segment embedding model generates segment embeddings that are aligned with text embeddings in the joint space of vision and language. Details of the two components are as follows.

3.1. USE Data Pipeline

In this section, we introduce our data pipeline to automatically curate segment-text pairs whose semantics are closely aligned. We carefully designed the data pipeline in a way that both the segments and text encapsulate information at multiple levels of granularity, with the purpose of enhancing the open-vocabulary recognition ability of our model.

The proposed data pipeline can be generalized for curating data from multiple types of data sources including image-only datasets (e.g., CIFAR-100 [17]), image-caption datasets (e.g., COCO [2], SBU [27], and CC3M [30]), and image with phrase grounding boxes (e.g., Visual Genome [16]). This unified data pipeline consolidates the segment-text pairs extracted from different image datasets and generates a collection of segments for each image where each segment can have multiple text descriptions associated with it. More importantly, this data pipeline is fully automatic and can be easily scaled up to billions of images.

The high-level overview of the proposed data pipeline is presented in Figure 2. It can be decomposed into three major modules: (a) an image captioning module that generates detailed descriptions of the image at different levels of granularity, (b) a referring expression grounding module that produces box-text pairs based on the images and captions, and (c) a mask generation module that converts box-text pairs into segment-text pairs. A detailed illustration of our data pipeline is discussed in the following sections.

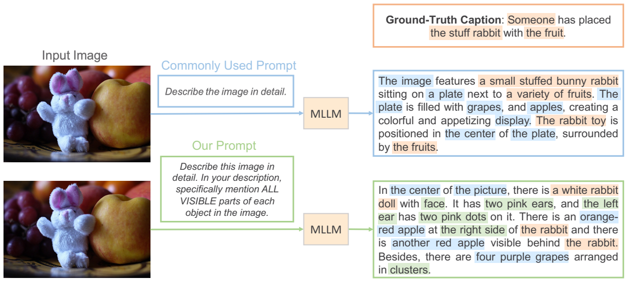

Ground-Truth Caption: Someone has placed the stuff rabbit with the fruit.
Input Image
Commonly Used Prompt
Describe the image in detail.
MLLM
The image features a small stuffed bunny rabbit sitting on a plate next to a variety of fruits. The plate is filled with grapes, and apples, creating a colorful and appetizing display. The rabbit toy is positioned in the center of the plate, surrounded by the fruits.
Our Prompt
Describe this image in detail. In your description, specifically mention ALL VISIBLE parts of each object in the image.
MLLM
In the center of the picture, there is a white rabbit doll with face. It has two pink ears, and the left ear has two pink dots on it. There is an orange-red apple at the right side of the rabbit and there is another red apple visible behind the rabbit. Besides, there are four purple grapes arranged in clusters.

Figure 3. The examples of the ground-truth caption, the caption generated with the commonly used image captioning prompt, and the caption generated with our prompt. Our prompt can guide the MLLM to generate captions with more fine-grained object parts.

Multi-Granularity Image Captioning. Our data pipeline starts with generating descriptions of objects (or parts) as well as their attributes from images. The quality and diversity of the descriptions play an important role in extracting segment-text pairs that cover objects in images as much as possible. We initially start with web-crawled or human-generated image captions (e.g., COCO [2], SBU [27], CC3M [30]) following previous work [28, 29]. However, we observe that these captions either lack descriptions about object attributes or only focus on the main objects in the image (see the ground-truth caption in Figure 3). This motivates us to generate image captions with richer semantic information. To this end, we leverage the recent advances of MLLMs such as CogVLM [32], Kosmos-2 [28], and LLaVA [21]. For all the MLLMs, the design of the text prompt is important for guiding the MLLMs to generate captions with desired properties. In order to obtain detailed descriptions of objects and parts in images, we prompt the MLLMs as follows:

"Describe this image in detail. In your description, specifically mention ALL VISIBLE parts of each object in the image."

Compared with the commonly used image captioning prompts (e.g., "Describe the image in detail."), our prompt allows MLLMs to not only describe the objects along with their attributes but also mention the visible parts of each object presented in the image. As shown in Figure 3, the caption generated with our prompt specifically mentions "*face*" and "*two pink ears*" along with detailed descriptions of the color of the apple, while the caption generated with the commonly used prompt fails to include this level of fine-grained details about the image. In our experimental study, we chose to employ CogVLM as the MLLM for generating multi-granularity captions.

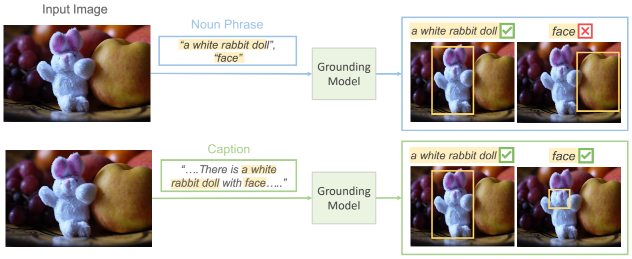


Figure 4. The examples of generated box-text pairs if we query the grounding model with either the entire caption or individual noun phrases. Querying with the entire caption can help to accurately identify object parts by considering more context information.

Referring Expression Grounding from Captions. Given the captions from different sources (i.e., ground-truth captions and MLLM-generated captions), the next step is extracting referring expressions from the captions and identifying their corresponding image regions represented by bounding boxes. Inspired by Kosmos-2 [28], we first extract the noun phrases using spaCy [11] and then expand the noun phrases as referring expressions. For example, from a caption ("*There is an orange-red apple at the right side of the rabbit and there is another red apple visible behind the rabbit.*"), we can obtain the noun phrases (*"an orange-red apple", "the right side", "the rabbit", "another red apple"*). We further expand the noun phrases to referring expressions by recursively traversing the children of noun phrases in the dependency tree and concatenating them. For the above example, the referring expressions we obtained after expanding noun phrases are (*"an orange-red apple", "the right side of the rabbit", "the rabbit", "another red apple visible behind the rabbit"*). Clearly, referring expressions could capture more context information regarding the objects. Existing open-vocabulary segmentation models that contain segment-text curation pipelines [7, 20] have a limited understanding of the text, either only including nouns (e.g., *"apple", "side", "rabbit"*) from the caption, or including adjectives and nouns separately (e.g, *"apple", "side", "rabbit", "orange-red", "red", "visible", "right"*). Compared with their approaches, the training data curated by our data pipeline will encapsulate richer semantics such that our open-vocabulary recognition ability can be enhanced and the predicted segments can be more consistent with the text query.

In order to obtain the bounding boxes associated with the extracted referring expressions, we adopt the open-vocabulary grounding models (e.g., Grounding DINO [22] and CoDet [14]). Note that some of the MLLMs [28] also offer the grounding capability, however, the generated bounding boxes are less accurate than the specialized grounding models. In this work, we use the Grounding Dino as an example. Given the image caption, there are two possible approaches to collecting bounding boxes associated with the noun phrases: querying with the noun phrases individually like what previous method [28] did or querying with the entire caption and then matching the boxes with the phrases. We observe that querying with the entire caption allows the grounding model to capture the comprehensive referring relationships implicitly encapsulated in the caption. In particular, when querying for object parts, the context is extremely important. For example, as shown in Figure 4, the rabbit face can be accurately located when querying with the entire caption, while the face is mistakenly assigned with a bounding box containing the apple if we query with the noun phrase *"face"* alone. Hence, we decided to query the grounding model with the entire caption and match the boxes with the phrases as follows. For each predicted box, we first identify the token with the highest probability score and associate the box with the noun phrase that contains the identified token. As a result, we generate a collection of box-text pairs for the next step. Note that we also extend box-phrase pairs to box-expression pairs and store both because the description of an image region can be ambiguous and from multiple levels of detail.

Mask Generation with Box Prompt. Given the box-text pairs generated by the referring expression grounding model mentioned above or directly from human annotations (e.g., Visual Genome [16]), the next step is to convert the bounding boxes into masks. We employ the image segmentation model SAM [15] which takes a bounding box as a prompt and outputs the mask of the best object that tightly fits with the box. For each box, the SAM will generate multiple masks, and we only choose the one with the highest stability score (predicted by the SAM). Similar to SAM, we perform two post-processing steps over the chosen masks including filling the small holes and removing the isolated small components. We notice that for some text with vague meanings (e.g., a room, the atmosphere), the bounding boxes often cover the entire image. If the size of the box is greater than 90% of the image size, we directly use the mask of the entire image as the corresponding segments without using SAM. Then, a collection of segment-text pairs can be obtained and merged via mask-based non-maximum-suppression (NMS). We use NMS to remove duplicate masks for each image because different text descriptions may refer to the same object in the image. After NMS, all the text descriptions associated with the duplicate masks will be merged and assigned to the corresponding mask.

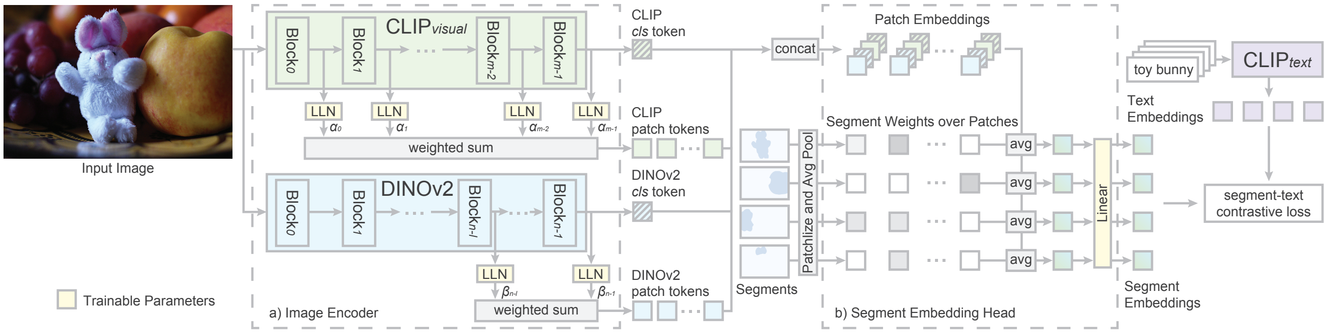


Figure 5. Architecture of the USE model, which consists of: a) an image encoder to extract image features for local patches and b) a segment embedding head maps the image features to segment embeddings that are aligned with text descriptions. The USE model is trained with segment-text contrastive loss using the segment and text embeddings.

3.2. USE Model

Inspired by recent advancements in multi-modal foundation models (e.g., LLaVA [21], CogVLM [32]), we introduce the USE model, which leverages the capabilities of pre-trained foundation models (i.e., CLIP [29] and DINOv2 [26]) with minimal trainable parameters. The architecture of the USE model is illustrated in Figure 5, comprising two major components: a) an image encoder that extracts image features by adapting the pre-trained foundation models, and b) a segment embedding head that generates segment embeddings based on the input segments and maps the segment embeddings to the vision-language space. In the subsequent sections, we first provide a detailed description of these two components and then discuss the training and loss of the model.

Image Encoder. Given an input image x, we exploit pre-trained vision transformers (ViTs) to extract patch embeddings $\mathbf{z} \in \mathbb{R}^{N \times D}$, where N is the number of image patches and D is the embedding dimension. To capture local features from image patches for the segmentation task, we use the *multi-level feature merging* introduced in COMM [13], which uses both CLIP and DINOv2 to extract the embeddings. Specifically, given the CLIP model CLIP$_{visual}$ and an input image x, we extract patch embeddings from all transformer blocks $\text{CLIP}_{visual}(x) = [\mathbf{c}^0, \mathbf{c}^1, \dots, \mathbf{c}^{m-1}]$, where m is the number of transformer blocks. To align embeddings from different blocks, we apply a linear-layernorm module (LLN) [13] to patch embeddings of each block. The LLN is a layer norm layer followed by a linear layer. Then, we merge the patch embeddings from different blocks by weighted sum $\overline{\mathbf{c}} = \sum_{i=0}^{m-1} \alpha_i \cdot \text{LLN}(\mathbf{c}^i)$, where the block scales α_i are learned during training. The DINOv2 patch embeddings $\overline{\mathbf{d}}$ are also extracted with the same approach. Note that we only extract patch embeddings from the last l blocks of DINOv2 because the shallow features lead to significant performance degradation [13]. Hence, the DINOv2 patch embeddings are $\overline{\mathbf{d}} = \sum_{i=n-l}^{n-1} \beta_i \cdot \text{LLN}(\mathbf{d}^i)$. In order to capture global image features, we also obtain the image embeddings from the cls tokens of CLIP and DINOv2, denoted as $\hat{\mathbf{c}}$ and $\hat{\mathbf{d}}$. In the end, the output of our image encoder is the patch-wise concatenation of the extracted embeddings as $\mathbf{z} = [\overline{\mathbf{c}}, \hat{\mathbf{c}}, \overline{\mathbf{d}}, \hat{\mathbf{d}}]$. It is worth mentioning that both CLIP and DINOv2 are frozen during training. The only trainable parameters in the image encoder are the LLN modules and the block scales (i.e., α_i and β_i).

Segment Embedding Head. Given arbitrary segments as prompt, the embedding head aims to extract segment embeddings from the patch embeddings $\mathbf{z}$ and map them to the joint space of vision and language. Specifically, given a segment s, we first calculate the segment's area within each patch and then normalize it with the patch size to determine the segment's weight within each patch. Then, we use these weights to compute the weighted average of the patch embeddings. Finally, the average embedding is mapped to the vision-language space with a linear layer and serves as the segment embedding $\mathbf{s}$. Note that we use simple mask pooling and linear projection, which are lightweight and cost-effective to train over a large scale of segment-text pairs. More sophisticated designs such as prompt encoder [15] and cross attention [1] can also be considered, which we leave for future work.

Training and Loss. After obtaining the segment embeddings $\mathbf{s}_{0,1,\dots,k-1}$ of a set of segments. We compute the text embeddings $\mathbf{t}_{0,1,\dots,k-1}$ of the corresponding texts. Then we use the segment-text contrastive loss to train the model as:

$$\begin{aligned} L = -\frac{1}{2k} \sum_{i=0}^{k-1} \Bigg[&\log \frac{\exp(\mathbf{s}_i \cdot \mathbf{t}_i / \tau)}{\sum_{j=0}^{k-1} \exp(\mathbf{s}_i \cdot \mathbf{t}_j / \tau)} \\ &+ \log \frac{\exp(\mathbf{s}_i \cdot \mathbf{t}_i / \tau)}{\sum_{j=0}^{k-1} \exp(\mathbf{s}_j \cdot \mathbf{t}_i / \tau)} \Bigg], \end{aligned} \tag{1}$$

where τ is the temperature parameter that scales the logits. Note that a segment may correspond to multiple text descriptions in the training data. At each training iteration, we randomly sample a text description for each segment in the mini-batch to compute the text embedding.

Dataset	#pairs	#pairs w/ NMS	#expressions
COCO (OVSeg) [20]	1.3M	-	0.3M
COCO	5.6M	1.3M	0.9M
VG	5.0M	2.9M	3.1M

Table 1. The number of segment-text pairs and unique expressions generated by the proposed data pipeline.

4. Experiments

4.1. Datasets

Training Data. We collect training data using the proposed data pipeline from two datasets including COCO [2] and Visual Genome (VG) [16]. For COCO, we use all training images with captions, which contain 118k images and 590k captions. We also use CogVLM-17B [32] to generate detailed captions for these training images. The hyperparameters of CogVLM-17B are set as follows: $temperature = 0.8$, $Top\,P = 0.4$, and $Top\,K = 5$. The images and the captions are fed into grounding DINO [22] with Swin-T backbone to generate bounding boxes of reference expressions. The box threshold is set to 0.05, and the NMS threshold is set to 0.7 for grounding DINO. The bounding boxes are then fed to SAM [15] with ViT-H backbone to generate the corresponding segments. Most of the hyperparameters for SAM are set as the default value, except the IoU threshold and the stability score threshold are both reduced to 0.6 to obtain more segments. Similar segments are merged using NMS with an IoU threshold of 0.7, and the corresponding expressions are merged into a list. For VG, we use the human-annotated box-text pairs from the training data directly and convert the boxes into segments using SAM with the same hyperparameter setting as COCO. The numbers of segment-text pairs and unique expressions are shown in Table 1. Compared with OVSeg [20], our data pipeline generates 4 times segment-text pairs and 3 times unique expressions on the COCO dataset, because OVSeg only focuses on nouns.

Test Data. We evaluate the USE Model on two tasks, including open-vocabulary semantic segmentation and open-vocabulary part segmentation. For open-vocabulary semantic segmentation, we evaluate our model on ADE20K [38] and Pascal Context [25] datasets. ADE20K is a large-scale dataset for scene understanding with 20K training images and 2K validation images. We use the validation set with two sets of categories for evaluation, one set includes 150 frequently used categories (ADE-150) and the other set contains a full list of 847 categories (ADE-847). Pascal Context is a dataset for semantic understanding with 4,998 training and 5,105 validation images. We also use the validation set with two sets of categories for evaluation including one with 59 categories (PC-59) and the other one with 459 categories (PC-459). For open-vocabulary part segmentation, we perform the experiments on the PartImageNet [9] dataset, which contains 16,540 training images and 2,957 validation images. We use the validation set for evaluation, which contains 40 part categories. It is worth mentioning that our model is not trained on any of the training images mentioned above. Moreover, none of the category names are known before testing.

4.2. Implementation Details

We employ the ViT-L/14 CLIP model pre-trained on 336×336 resolution and the ViT-L/14 distilled DINOv2 model in the image encoder. For the CLIP model, we collect patch tokens from all transformer blocks and for DINOv2 we only use the patch tokens output from the last 6 transformer blocks. The embeddings of expressions are generated by the same ViT-L/14 CLIP model with 4 prompt templates including: *a photo of* {}, *This is a photo of* {}, *There is* {} *in the scene*, and *a photo of* {} *in the scene*. During training, the input images are augmented with random image resizing with a scaling factor from 0.5 to 2 and random cropping with a size of 560×560. The USE model is trained on the generated segment-text pairs for 5 epochs with a batch size of 32. The temperature τ from the segment-text contrastive loss is set to 30 for all experiments. We set the initial learning rate to 0.001 and decay it with a polynomial learning rate policy with a power of 0.9. The AdamW optimizer is used with a weight decay of 0.01.

4.3. Open-Vocabulary Semantic Segmentation

We evaluate our method with open-vocabulary semantic segmentation using class-agnostic masks. The class-agnostic masks are generated by prompting SAM with a regular grid of point prompts followed by filtering and merging duplicate masks via NMS. For each mask, we first obtain its embedding using our model and then compute the similarities between the segment embedding and the text embeddings of the target classes. Here, we adopt the prompt template used in [10] to generate text embeddings as the class names are mostly nouns. The similarities are then converted to probabilities with softmax. To generate semantic segmentation maps, we calculate the class prediction of each pixel by aggregating the probabilities of all segments that cover the pixel and taking the class with the highest probability.

We compare the performance of our method with the state-of-the-art open-vocabulary semantic segmentation methods [4, 7, 19, 20, 34–36] on the ADE20K and Pascal Context datasets. The performance is evaluated with the mean Intersection over Union (mIoU) across all classes. For methods that were evaluated with different CLIP models [20, 35, 36], we use results from the ViT-L/14 CLIP model for comparison. For other methods, we use the high-

Method	Type	Training Data	VL-Model	ADE-150	ADE-847	PC-59	PC-459	Average
LSeg+ [19]	end2end	COCO	ALIGN EN-B7	18.0	3.8	46.5	7.8	19.0
ZegFormer [4]	end2end	COCO	CLIP ViT-B/16	16.4	-	-	-	-
OpenSeg [7]	end2end	COCO	ALIGN EN-B7	28.6	8.8	48.2	12.2	24.4
ODISE [34]	end2end	COCO	Stable Diffusion	29.9	11.1	57.3	14.5	28.2
SAN [36]	end2end	COCO	CLIP ViT-L/14	32.1	12.4	57.7	**15.7**	29.4
SimSeg [35]	two-stage	COCO	CLIP ViT-L/14	21.7	7.1	52.2	10.2	22.8
OVSeg [20]	two-stage	COCO	CLIP ViT-L/14	29.6	9.0	55.7	12.4	26.6
OVSeg+SAM	two-stage	COCO	CLIP ViT-L/14	27.5	8.8	51.2	12.3	24.9
USE+SAM (ours)	two-stage	COCO†	CLIP ViT-L/14	**37.0**	**13.3**	**57.8**	14.7	**30.7**
USE+SAM (ours)	two-stage	COCO,VG	CLIP ViT-L/14	37.1	13.4	58.0	15.0	30.9

Table 2. **Open-vocabulary semantic segmentation benchmarks measured by mIoU.** Our method outperforms the state-of-the-art two-stage methods by a large margin on all datasets. Our method also achieves the best average performance compared with all previous methods. † We use all segment-text pairs from COCO images including the annotations from VG.

est performance number of each method for comparison. We first train our model on the segment-text pairs from the COCO images for fair comparison. Similar to other two-stage methods [20, 35], we also train an extra model on the COCO ground truth annotations and use predictions from both models to make the final prediction. Table 2 shows the benchmarking results on the ADE20K and Pascal Context datasets. We observe that our method consistently outperforms the state-of-the-art two-stage methods on all datasets by a large margin. Note that OVSeg's performance declines when using SAM segments, indicating that SAM segments at varying granularities are even more challenging to classify. Meanwhile, our method archives the best average performance compared with the state-of-the-art end-to-end methods while preserving the flexibility of taking segments at various granularities as prompt. Extra performance boost can also be observed when training on COCO plus VG images.

Method	Datasets	All	Quadruped			
		(40)	head	body	foot	tail
USE (ours)	COCO	6.2	8.8	2.6	2.6	18.5
VLPart [31]	Pascal Part	4.5	17.4	0.1	0.0	2.9
VLPart [31]	+ ImageNet	5.4	23.6	3.4	0.8	1.2
VLPart [31]	+ ImageNet w/ Parts	7.8	35.0	15.2	3.5	8.9

Table 3. **Open-vocabulary part segmentation benchmarks on PartImageNet measured by mAP.** Here, $mAP_{mask}@[0.5, 0.95]$ of all 40 parts and Quadruped's parts are presented. Specifically, our method is trained on COCO datasets that do not contain any human-annotated part segments. In contrast, VLPart is first trained on human-annotated part data, Pascal Part. Then, image-level annotations and part-level annotations on ImageNet are added to the training data sequentially.

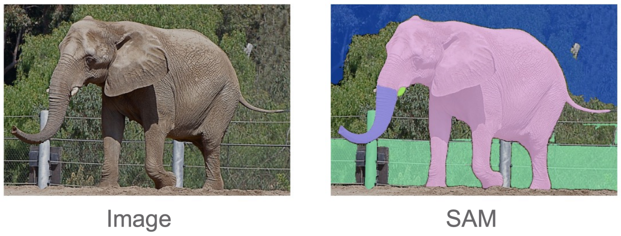


Figure 6. Illustrative example of class-agnostic masks generated by SAM. SAM fails to capture the elephant's head because the boundary lines between the head and the neck are very blurry.

4.4. Open-Vocabulary Part Segmentation

In addition to semantic segmentation, the effectiveness of our method is also evaluated with open-vocabulary part segmentation on PartImageNet dataset [9]. To begin with, the class-agnostic masks are generated using SAM. The similarities between the segment embeddings and text embeddings of the target classes are obtained with the same approach discussed in Section 4.3. Because the class-agnostic masks are generated by prompting SAM with uniformly sampled points over the entire image, most of the proposed masks do not contain any object parts. Instead, it may contain the entire object in the foreground or the objects in the background. Therefore, we combine the classes of the parts with a list of common background classes to perform classification. Specifically, we include the 91 COCO stuff classes and 11 super-categories from PartImageNet. During inference, we only evaluate the masks whose most similar classes are one of the target part categories.

To evaluate the performance of our model, we compare it against VLPart [31] which is specifically designed and trained for open-vocabulary part segmentation. To assess our open-vocabulary recognition ability on parts against

Image Encoder	ADE-150 (mIoU)	ADE-847 (mIoU)
CLIP	30.2	10.3
DINOv2	31.9	10.2
CLIP + DINOv2	**32.6**	**11.3**

Table 4. **Ablation study** on the choice of the pre-trained backbone. Combining CLIP and DINOv2 gives the best mIoU.

Architecture	ADE-150 (mIoU)	ADE-847 (mIoU)
w/o cls token	31.4	10.0
w/ cls token	**31.9**	**10.2**

Table 5. **Ablation study** on architecture design of the image encoder. Only DINOv2 is used in the study.

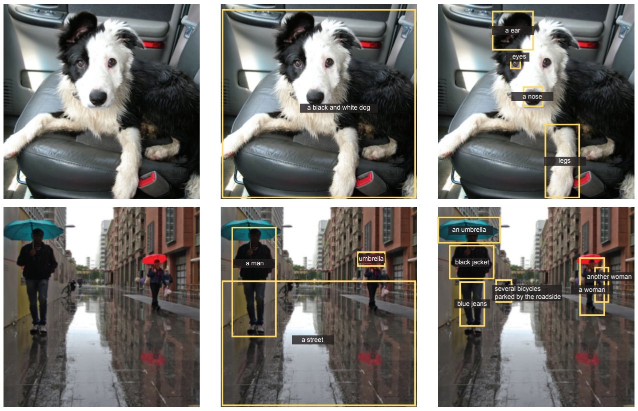


Figure 7. More fine-grained objects and parts can be extracted from MLLM augmented captions compared with ground truth captions.

VLPart, we choose to compare with their cross-dataset generalization performance on the PartImageNet dataset, i.e., the VLPart model is trained on datasets other than PartImageNet. Following VLPart, we adopt $mAP_{mask}@[0.5, 0.95]$ as our evaluation metric. As indicated in Table 3, our model outperforms VLPart trained on Pascal Part (human-annotated part data) and VLPart trained on Pascal Part + ImageNet over all 40 categories by 1.7 and 0.8, respectively, even though our model was not trained on any human-annotated part data and have not seen any images from ImageNet during training. Compared with the VLPart trained on Pascal Part + ImageNet + ImageNet w/ Parts (4th row in Table 3), our performance is slightly worse. However, it's important to note that this VLPart model is not under an open-vocabulary setup, as it relies on known target classes of the downstream task and incorporates part segments derived from human annotations.

In terms of the detailed metrics of Quadruped, our model achieved sufficiently high mAP on the body, foot, and especially tail parts, but our model does not perform well in terms of the head. This is caused by the limitation of SAM because SAM is mostly edge-oriented and thus hardly differentiates two parts if the boundary edges between them are blurry. For example, as shown in Figure 6, the elephant's trunk has clear edges, whereas the boundary lines between the elephant's head and the elephant's neck are fuzzy and thus can hardly be distinguished by SAM. It is worth mentioning that, our method is flexible enough to take the class-agnostic masks generated by any image segmentation model. Hence, segmentation models that are specifically designed for parts can be used to improve our open-vocabulary part segmentation performance.

4.5. Ablation Study

We study the choice of the pre-trained backbone on the ADE20K dataset and the open-vocabulary semantic segmentation task. We train the model on the COCO images and set the crop size of images to 336×336 during training to reduce computation costs. The evaluation results are shown in Table 4, which shows that performance gains can be obtained by combining CLIP and DINOv2.

Compared with COMM [13], we propose to concatenate the cls token with the patch tokens when extracting image features for the embedding head. We study the influence of the cls token on the ADE20K dataset for open-vocabulary semantic segmentation. The model is trained with the same hyperparameter setting as the previous study. The performance number with and without using the cls token is shown in Table 5. We can see that the mIoU is improved consistently by including the cls token.

We qualitatively compare the objects extracted from ground truth captions and MLLM-augmented captions in Figure 7. More fine-grained objects and parts can be captured by MLLM-augmented captions compared with ground truth captions. For example, the eye, nose, ear, and leg of the dog.

5. Conclusion

This paper presents the USE framework for open-vocabulary image segmentation. By integrating a carefully designed data pipeline and a lightweight embedding model, the USE framework effectively classifies image segments in a zero-shot manner without human annotations. Our approach leverages pre-trained foundation models, optimized for efficiency and scalability. Extensive experiments demonstrate the superiority of the USE framework over existing methods in semantic and part segmentation. We hope this work can shed some light on building foundation models for open-vocabulary image segmentation and segment-based representation learning.

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, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, pages 23716–23736, 2022. 5

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 1, 3, 6

[3] Liangliang Cao, Bowen Zhang, Chen Chen, Yinfei Yang, Xianzhi Du, Wencong Zhang, Zhiyun Lu, and Yantao Zheng. Less is more: Removing text-regions improves CLIP training efficiency and robustness. *arXiv preprint arXiv:2305.05095*, 2023. 2

[4] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *CVPR*, pages 11583–11592, 2022. 6, 7

[5] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with MaskCLIP. In *ICML*, pages 8090–8102, 2023. 2

[6] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving CLIP training with language rewrites. *arXiv preprint arXiv:2305.20088*, 2023. 2

[7] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, pages 540–557, 2022. 1, 4, 6, 7

[8] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022. 2

[9] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, and Alan Yuille. PartImageNet: A large, high-quality dataset of parts. *arXiv preprint arXiv:2112.00933*, 2021. 6, 7

[10] Wenbin He, Suphanut Jamonnak, Liang Gou, and Liu Ren. CLIP-S4: Language-guided self-supervised semantic segmentation. In *CVPR*, pages 11207–11216, 2023. 2, 6

[11] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength natural language processing in python. 2020. 4

[12] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *CVPR*, pages 7020–7031, 2022. 2

[13] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Xiaopeng Zhang, Jin Li, Hongkai Xiong, and Qi Tian. From CLIP to DINO: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023. 5, 8

[14] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, pages 11144–11154, 2023. 4

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, pages 4015–4026, 2023. 1, 4, 5, 6

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 3, 4, 6

[17] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 3

[18] Zhengfeng Lai, Haotian Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, Yinfei Yang, and Meng Cao. From scarcity to efficiency: Improving CLIP training via visual-enriched captions. *arXiv preprint arXiv:2310.07699*, 2023. 2

[19] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 6, 7

[20] 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 *CVPR*, pages 7061–7070, 2023. 1, 2, 4, 6, 7

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 3, 5

[22] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2, 4, 6

[23] Pratyush Maini, Sachin Goyal, Zachary C. Lipton, J. Zico Kolter, and Aditi Raghunathan. T-MARS: Improving visual representations by circumventing text feature learning. *arXiv preprint arXiv:2307.03132*, 2023. 2

[24] Ron Mokady, Amir Hertz, and Amit H. Bermano. Clip-Cap: CLIP prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2

[25] 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 *CVPR*, pages 891–898, 2014. 6

[26] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5

[27] Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. Im2Text: Describing images using 1 million captioned photographs. In *NeurIPS*, pages 1143–1151, 2011. 3

[28] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 1, 2, 3, 4

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1, 2, 3, 5

[30] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018. 3

[31] Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Going denser with open-vocabulary part segmentation. *arXiv preprint arXiv:2305.11173*, 2023. 7

[32] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. CogVLM: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. 2, 3, 5, 6

[33] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic segmentation emerges from text supervision. In *CVPR*, pages 18134–18144, 2022. 2

[34] 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 *CVPR*, pages 2955–2966, 2023. 2, 6, 7

[35] 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 *ECCV*, pages 736–753, 2022. 1, 2, 6, 7

[36] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, pages 2945–2954, 2023. 1, 2, 6, 7

[37] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. RegionCLIP: Region-based language-image pretraining. In *CVPR*, pages 16793–16803, 2022. 2

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, pages 633–641, 2017. 6

[39] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. ChatGPT asks, BLIP-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv preprint arXiv:2303.06594*, 2023. 2

[40] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. Generalized decoding for pixel, image, and language. In *CVPR*, pages 15116–15127, 2023. 2