A Simple Framework for Open-Vocabulary Segmentation and Detection

Hao Zhang[1,2*], **Feng Li**[1,2*], **Xueyan Zou**[4], **Shilong Liu**[2,5]
Chunyuan Li[3], **Jianwei Yang**[3†], **Lei Zhang**[2†]
[1]The Hong Kong University of Science and Technology.
[2]International Digital Economy Academy (IDEA).
[3]Microsoft Research at Redmond. [4]University of Wisconsin-Madison.
[5]Dept. of CST., BNRist Center, Institute for AI, Tsinghua University.

{hzhangcx, fliay}@connect.ust.hk {xueyan}@cs.wisc.edu {liusl20}@mails.tsinghua.edu.cn {jianwyan,chunyl}@microsoft.com {leizhang}@idea.edu.cn

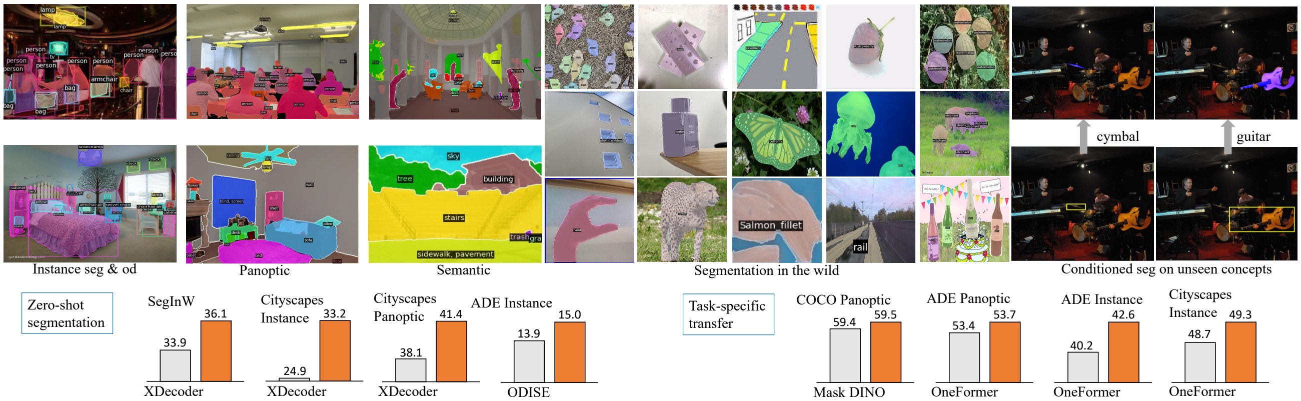


Figure 1: Top row: visualizations of *OpenSeeD* for open-vocabulary instance segmentation and detection, panoptic and semantic segmentation, instance segmentation in the wild, and conditioned segmentation given referring box location and concept. Bottom row: Our *OpenSeeD* model outperforms previous state-of-the-art methods (listed below each gray bar) on eight benchmarks in zero-shot and task-specific transfer settings.

Abstract

We present OpenSeeD, a simple ***Open****-vocabulary* ***Se****gm****e****ntation and* ***D****etection framework that jointly learns from different segmentation and detection datasets. To bridge the gap of vocabulary and annotation granularity, we first introduce a pre-trained text encoder to encode all the visual concepts in two tasks and learn a common semantic space for them. This gives us reasonably good results compared with the counterparts trained on segmentation task only. To further reconcile them, we identify two discrepancies: i) task discrepancy – segmentation requires extracting masks for both foreground objects and background stuff, while detection merely cares about the former; ii) data discrepancy – box and mask annotations are with different spatial granularity, and thus not directly interchangeable. To address these issues, we propose a decoupled decoding to reduce the interference between foreground/background and a conditioned mask decoding to assist in generating masks for given boxes. To this end, we develop a simple encoder-decoder model encompassing all three techniques and train it jointly on COCO and Objects365. After pre-training, our model exhibits competitive or stronger zero-shot transferability for both segmentation and detection. Specifically, OpenSeeD beats the state-of-the-art method for open-vocabulary instance and panoptic segmentation across 5 datasets, and outperforms previous work for open-vocabulary detection on LVIS and ODinW under similar settings. When transferred to specific tasks, our model achieves new SoTA for panoptic segmentation on COCO and ADE20K, and instance segmentation on ADE20K and Cityscapes (The bottom row in Fig. 1 shows a comparison of the performance of OpenSeeD and previous SoTA methods). Finally, we note that OpenSeeD is the first to explore the potential of joint training on segmentation and detection, and hope it can be received as a strong baseline for developing a single model for both tasks in the open world. Code will be released at https://github.com/IDEA-Research/OpenSeeD.*

[*] Equal contribution. List in random.
[†] Equal advisory contribution.

This work is developed during an internship at IDEA.

1. Introduction

Developing vision systems that can be transferable to novel concepts or domains has emerged as an important research topic in the community. In the light of strong zero-shot transferability demonstrated in the seminal work CLIP [40], a number of researchers have attempted to build advanced open-vocabulary models by leveraging large-scale image-text pairs for fine-grained vision tasks such as detection [10, 12, 22, 30, 57] and segmentation [7, 17, 48, 59].

Arguably, core vision tasks like detection and segmentation are fairly distinct in their vocabulary sizes and spatial granularities of supervision, as illustrated in Fig. 2 (a). For example, the commonly used public detection dataset Objects365 [42] contains box annotations for 365 concepts in around 1.7M images, while mask annotations in COCO [33] cover merely 133 categories in 0.1M images. Previous works have explored different ways of leveraging a large amount of image-text data for open-vocabulary detection or segmentation, such as distilling the visual-semantic representations from multi-modal foundation models [12, 57], designing fine-grained or augmented contrastive learning methods [39] or utilizing pseudo-labeling techniques [30, 56]. To the best of our knowledge, most (if not all) of them focused on how to improve the performance for either detection or segmentation. Moreover, transferring weak image-level supervision to fine-grained tasks usually requires sophisticated designs to mitigate the huge granularity gap and is vulnerable to noises in image-text pairs. This leads to a natural question: *can we bridge detection and segmentation that are cleaner and have a closer gap to attain a good open-vocabulary model for both?*

Taking one step back, marrying detection and segmentation had been previously explored in two main ways. On one hand, Mask R-CNN [15] is one of the first works that proposed to jointly learn detection and instance segmentation on COCO. On the other hand, it is shown that detection models pre-trained on Objects365 can be feasibly transferred for COCO panoptic segmentation [29]. However, as depicted in Fig. 2 (b), the former method requires the model to be trained on the same dataset containing aligned box and mask annotations, while the latter method follows pre-train-then-fine-tune protocol, leading to two separate closed-set models. In this work, we are the first to propose jointly learning from detection and segmentation data, and more importantly serving an open-vocabulary model for both tasks (Fig. 2 (b) bottom). Achieving this goal requires answering two critical questions: i) how to transfer the semantic knowledge across detection and segmentation data; ii) how to bridge the gap between box and mask supervision. First, the vocabulary shares commons but also bear substantial differences between the two tasks. We need to accommodate the two vocabularies and further go beyond towards *open* vocabulary. Second, semantic and panoptic segmentation tasks require segmenting not only foreground objects (things like “dog” and “cat”.) but also background concepts (stuff like “sky” and “building”), while detection task solely cares about foreground objects. Third, box supervision by nature is coarser than mask supervision. We can convert masks into boxes but hardly vice versa.

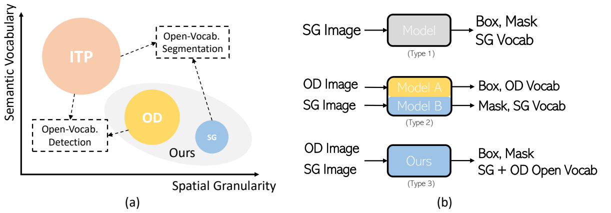


Figure 2: (a) Semantic vocabulary sizes and spatial granularities comparison across different vision tasks/datasets. “ITP” means image-text pairs; “OD” means object detection and “SG” means segmentation. Our *OpenSeeD* is the first open-vocabulary model that jointly learn on segmentation and detection (gray region). (b) Different types of methods connect object detection and segmentation.

To the end, we propose *OpenSeeD*, a simple encoder-decoder framework to reconcile the two tasks by mitigating the aforementioned problems. Concretely, we first exploit a single text encoder to encode all concepts occurring in the data and train our model to align the visual tokens with the semantics in a common space. Second, we explicitly divide the object queries in the decoder into two sub-types: foreground and background queries, where the first group is responsible for foreground objects from both segmentation and detection while the second group is only for background stuffs in segmentation. Third, we introduce conditioned mask decoding which learns to decode masks from ground-truth boxes from segmentation data and generates the mask assistant for detection data. As a result, our *OpenSeeD* is able to learn from separate detection and segmentation data seamlessly and achieves outstanding or competitive zero-shot and transfer performance across various tasks/datasets. Fig 1 shows a visualization of our model on instance, panoptic and semantic segmentation tasks. It also shows the segmentation results on datasets that largely differ from our training data such as the SeginW datasets and demonstrates the conditioned segmentation ability of *OpenSeeD. Given the encouraging results, we hope our work can contribute as the first strong baseline for developing a single open-vocabulary model for both tasks.*

Contributions. To summarize, our main contributions are:

- We are the first to present a strong baseline model that can jointly learn from detection and segmentation data towards an open-vocabulary model for both tasks.
- We identify the discrepancies in two tasks/datasets and propose separate techniques including shared semantic space, decoupled decoding, and conditioned mask assistance to mitigate the issues.
- By jointly training our model on segmentation and detection data, we achieve new state-of-the-art perfor-

mance for zero-shot task transfer across a variety of segmentation datasets, and competitive performance for zero-shot object detection.

2. Related Work

Generic Segmentation and Detection. Detection and segmentation have been long-standing problems in the vision community [8, 9, 38, 62]. Both tasks require understanding what and where the visual concepts are but with different spatial granularities. Generic segmentation mainly includes instance, semantic and panoptic segmentation [3, 15, 21], with respect to different semantics. Recently, Detection Transformer (DETR) [1] that is based on Transformer [44] has achieved significant progress in many detection [28, 34, 37, 55, 60] and segmentation models [5, 18, 29, 32]. However, all these methods are constrained to a limited vocabulary size.

Open-Vocabulary Segmentation. Many open-vocabulary segmentation models [7, 11, 17, 26] leverages large pretrained vision-language models (*e.g.*, CLIP [19] or ALIGN [40]) to distill or transfer the visual-semantic knowledge. Apart from using foundation models, DenseCLIP [41] and GroupViT [46] show that fine-tuning from a foundation model or training from scratch can also yield superior zero-shot performance. Recently, X-Decoder [61] proposes to unify all types of segmentation tasks and several vision-language tasks for open-vocabulary segmentation. In ODISE [47], the authors study a new way of using a text-to-image diffusion model as the backbone for open-vocabulary segmentation. Unlike the previous works, our model instead explores connecting segmentation and detection which have cleaner data and closer gap between each other.

Open-Vocabulary Detection. Similarly, some open-vocabulary detection models directly leverage foundation models for distillation or transfer like OV-DETR [54] and VILD [13]. Recently, GLIP [30] formulates detection as a special grounding problem to unify detection and phrase grounding tasks. These grounding data help improve the alignment between phrases and regions for open detection. RegionCLIP [57] and DetCLIP [51] generate pseudo box labels from image-text pairs for more generalized detection.

Weakly-Supervised Segmentation. Weakly-supervised segmentation typically only uses box annotation as supervision to generate segmentation. Prominent methods design teacher models or weak supervision loss, like BoxInst [43], Box2Mask [31], DiscoBox [25] and Mask Auto-Labelers [24]. All these models are with closed-set and usually inferior to models with segmentation supervision. In contrast, we attempt to leverage as much supervision as possible from both segmentation and detection for an open-vocabulary model.

Learning from Box and Mask. There are primarily two ways to learn from both box and mask. The first one is to train on a single dataset with both box and mask annotations. Prominent methods include Mask R-CNN [15] and HTC [2]. However, they are constrained to foreground instances. The second way is to pretrain with only box supervision and then transfer to segmentation. For example, HTC [2] and Mask DINO [29] can both learn from large-scale detection data and then be fine-tuned to a specific segmentation dataset. However, such a pretrain-and-finetune protocol leads to two separate models that are only capable of either detection or segmentation. Moreover, both models are closed-set and thus not transferable to novel concepts.

3. Method

Given segmentation and detection datasets, *OpenSeeD* is aimed at learning an open-vocabulary model for both tasks. Formally, let $\mathcal{D}_m = \{I_i, (\mathbf{c}_i, \mathbf{m}_i)\}_{i=1}^M$ denote the segmentation dataset of size M and $\mathcal{D}_b = \{I_j, (\mathbf{c}_j, \mathbf{b}_j)\}_{j=1}^N$ the detection dataset of size N, where $\mathbf{c}$ are the visual concepts in an image, and $\mathbf{m}$ and $\mathbf{b}$ the corresponding masks and boxes, respectively. Suppose $\mathcal{V} = \{c_1, ...c_K\}$ be the vocabulary of unique K visual concepts appearing in $\mathcal{D}_m \cup \mathcal{D}_b$. The goal of *OpenSeeD* is learning to *detect and segment* visual concepts in $\mathcal{V}$ and beyond.

To achieve the goal, we propose a general encoder-decoder design and employ a text encoder for our *OpenSeeD*, as shown in Fig. 3. Our model takes as input an image I and the vocabulary $\mathcal{V}$ and output a set of predictions including masks $\mathbf{P^m}$, boxes $\mathbf{P^b}$, and classification scores $\mathbf{P^c}$. As a whole, $\langle \mathbf{P^m}, \mathbf{P^b}, \mathbf{P^c} \rangle = \mathsf{OpenSeeD}(I, \mathcal{V})$. More specifically, our model consists of one image encoder $\mathsf{Enc_I}$, one text encoder $\mathsf{Enc_T}$, and one decoder Dec. Given an image I and the vocabulary $\mathcal{V}$, we first encode them by $\mathsf{Enc_I}$ and $\mathsf{Enc_T}$, respectively:

$$\mathbf{O} = \mathsf{Enc_I}(I), \mathbf{T} = \mathsf{Enc_T}(\mathcal{V}) \tag{1}$$

where the image features $\mathbf{O} \in \mathcal{R}^{H \times W \times C}$, and the text features $\mathbf{T} = \{t_1, t_2, ..., t_K\}$. Afterward, the decoder takes L queries $\mathbf{Q} \in \mathcal{R}^{L \times C}$ as inputs and cross-attends the image features to get outputs:

$$\begin{aligned} \langle \mathbf{P^m}, \mathbf{P^b}, \mathbf{P^s} \rangle &= \mathsf{Dec}\left(\mathbf{Q}; \mathbf{O}\right) \\ \mathbf{P^c} &= \mathsf{Sim}(\mathbf{P^s}, \mathbf{T}) \end{aligned} \tag{2}$$

where $\mathbf{P^s}$ is the decoded semantics. The visual-semantic matching scores $\mathbf{P^c}$ is derived from $\mathsf{Sim}(\mathbf{P^s}, \mathbf{T})$ by calculating the similarity scores between $\mathbf{P^s}$ and $\mathbf{T}$, which is used to compute the loss during training and predict the category during inference.

3.1. Basic Loss Formulation

In this basic formula, we attempt to reconcile the two tasks by promoting a shared visual-semantic space without touching other issues. For multiple tasks and datasets, our

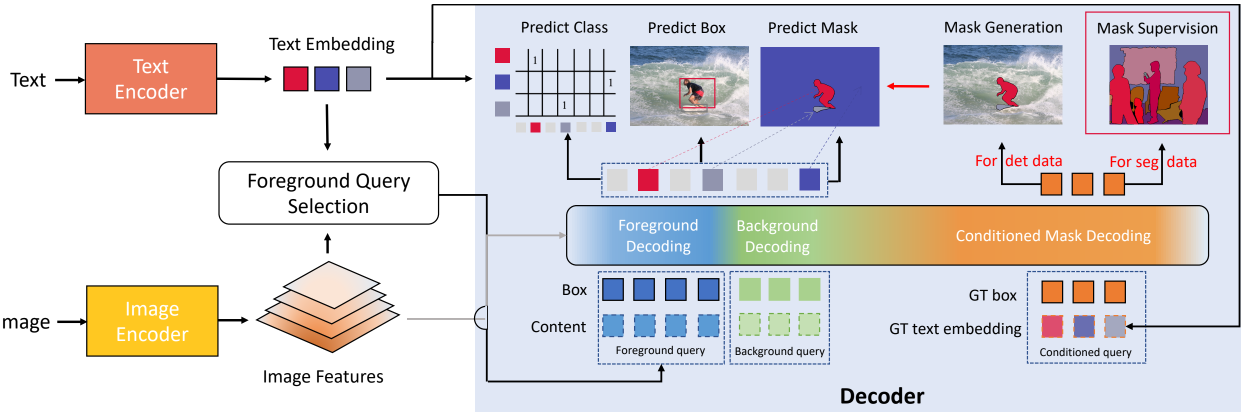


Figure 3: The framework of *OpenSeeD*. The framework consists of one image encoder, one text encoder, and our designed decoder with foreground, background, and conditioned mask decoding capability. "GT" means ground truth, and "GT text embedding" is the embedding encoded by the text encoder.

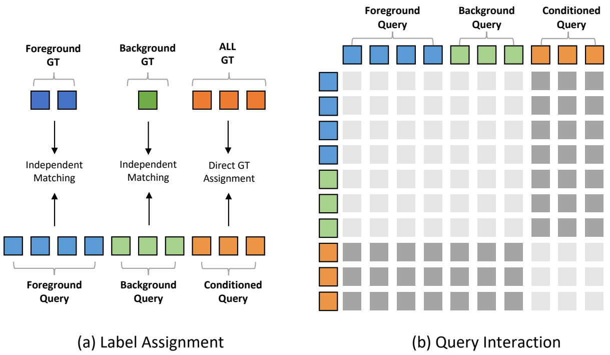
loss function can be written as follows:

$$\mathcal{L}_{all} = \sum_{I,(\mathbf{c},\mathbf{m})\in\mathcal{D}_m} \overbrace{\Big(\mathcal{L}_m(\mathbf{P^m},\mathbf{m}) + \mathcal{L}_b(\mathbf{P^b},\hat{\mathbf{b}}) + \mathcal{L}_c(\mathbf{P^c},\mathbf{c})\Big)}^{\text{Segmentation loss}} + \sum_{I,(\mathbf{c},\mathbf{b})\in\mathcal{D}_b} \underbrace{\Big(\mathcal{L}_b(\mathbf{P^b},\mathbf{b}) + \mathcal{L}_c(\mathbf{P^c},\mathbf{c})\Big)}_{\text{Detection loss}} \tag{3}$$

For clarity, we omit the weight for each loss term. Note that for the segmentation task, we can derive accurate boxes $\hat{\mathbf{b}}$ from masks $\mathbf{m}$ and use them to compute the box loss as in term $\mathcal{L}_b(\mathbf{P^b},\hat{\mathbf{b}})$. By summing over all the terms, our model can achieve a reasonably good open-vocabulary performance. Furthermore, it can be pre-trained end-to-end with detection and segmentation data, allowing it to perform open-vocabulary segmentation and detection using a single set of weights.

Despite building a strong baseline, we must consider the intrinsic discrepancies between the two tasks, as previously discussed. Semantic and panoptic segmentation require the recognition of both foreground and background, while detection focuses solely on localizing foreground objects. As a result, using the same queries for both tasks creates conflicts that can significantly degrade performance. Additionally, good box predictions are typically indicative of good masks, and vice versa. Separately training the box and mask head on detection and segmentation data obstructs the synergy of spatial supervision from both datasets.

To address the aforementioned discrepancies, we introduce a new decoder design for our *OpenSeeD*. We divide the queries $\mathbf{Q}$ into three types: L_f foreground queries $\mathbf{Q_f}$, L_b background queries $\mathbf{Q_b}$ and L_d conditioned queries $\mathbf{Q_d}$, and propose query-specific computations for each type. In the following, we will describe how we decouple the foreground and background decoding to address the task discrepancy in Sec. 3.2, and employ the conditioned mask decoding to tackle the data discrepancy in Sec. 3.3.



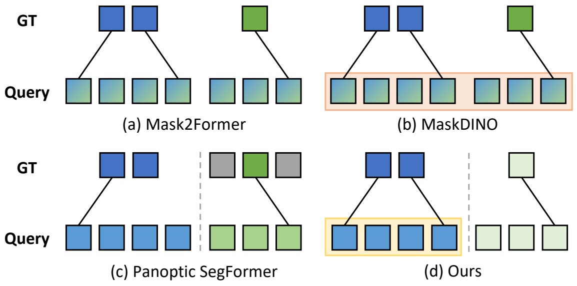
(a) Label Assignment

(b) Query Interaction

Figure 4: a) Label assignment for different queries. "gray" means unused for detection data without background stuff. Best viewed in color. b) Query interaction of different queries (foreground, background, and conditioned query) in *OpenSeeD*. All data has background queries including detection data. "dark" color means blocking interaction.

3.2. Bridge Task Gap: Decoupled Foreground and Background Decoding

Without loss of generality, we have defined the visual concepts that appear in instance segmentation and detection as foreground, while the stuff categories in panoptic segmentation are considered background. To mitigate the task discrepancy, we perform foreground and background decoding with foreground queries $\mathbf{Q_f}$ and background queries $\mathbf{Q_b}$, respectively. Specifically, for these two query types, our decoder predicts two sets of outputs, including $\langle\mathbf{P}_f^m,\mathbf{P}_f^b,\mathbf{P}_f^c\rangle$ for foreground and $\langle\mathbf{P}_b^m,\mathbf{P}_b^b,\mathbf{P}_b^c\rangle$ for background. We also divide the ground truths in segmentation dataset into two groups: $(\mathbf{c}_f,\mathbf{m}_f)$ and $(\mathbf{c}_b,\mathbf{m}_b)$, and then perform two independent Hungarian Matching processes for these two sets correspondingly, as shown in Fig.4 (a). Consequently, both foreground and background decoding are used for segmentation, while only foreground decoding is used for detection. As a result, our basic loss



(a) Mask2Former (b) MaskDINO

(c) Panoptic SegFormer (d) Ours

Figure 5: Compare different ways of handling foreground (fg) and background (bg) queries. Note: blue: fg. green: bg. mixed bule&green: treat fg and bg equally. Shaded queries are acquired by query selection from image features, others are learnable. a)&b) Mask2Former and MaskDINO teat fg and bg equally. They differ in that queries are learnable for Mask2Former while selected for MaskDINO. c) PanopticSegFormer discriminates fg and bg, but assign each bg query to a fixed category. d) Ours. The bg queries are lightgreen as they are not matched to GT in detection data.

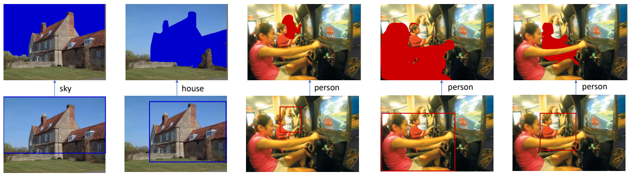


Figure 6: Background and foreground masks generated with boxes and text as the condition for sample images in ADE20K [58].

function in Eq. (3) is reformulated to:

$$\mathcal{L}_{all} = \sum_{I,(\mathbf{c},\mathbf{m})\in\mathcal{D}_m} \Bigg(\overbrace{\Big(\mathcal{L}_m(\mathbf{P_f^m}, \mathbf{m}_f) + \mathcal{L}_b(\mathbf{P_f^b}, \hat{\mathbf{b}}_f) + \mathcal{L}_c(\mathbf{P_f^c}, \mathbf{c}_f)\Big)}^{\text{Segmentation loss for foreground}}$$

$$+ \overbrace{\Big(\mathcal{L}_m(\mathbf{P_b^m}, \mathbf{m}_b) + L_b(\mathbf{P_b^b}, \hat{\mathbf{b}}_b) + \mathcal{L}_c(\mathbf{P_b^c}, \mathbf{c}_b)\Big)}^{\text{Segmentation loss for background}} \Bigg)$$

$$+ \sum_{I,(\mathbf{c},\mathbf{b})\in\mathcal{D}_b} \underbrace{\Big(\mathcal{L}_b(\mathbf{P_f^b}, \mathbf{b}) + \mathcal{L}_c(\mathbf{P_f^c}, \mathbf{c})\Big)}_{\text{Detection loss for foreground}} \tag{4}$$

where $\hat{\mathbf{b}}_f$ and $\hat{\mathbf{b}}_b$ are derived from $\mathbf{m}_f$ and $\mathbf{m}_b$, respectively. Based on such explicit decoupling, our model maximizes the cooperation of foreground supervision from both detection and segmentation datasets and significantly reduces the interference between foreground and background categories. Though decoupled, we note that these two types of queries share the same decoder and interact with each other with self-attention, as shown in Fig. 4 (b). Below we explain how the foreground and background queries are determined.

Language-guided foreground query selection. Open-vocabulary setting differs from the conventional closed-set setting in that a model is required to localize a large number of foreground objects far beyond the training vocabulary. However, the fact is that our decoder contains a limited number of foreground queries (a few hundred typically), making it hardly handle all possible concepts in the image. To address this issue, we propose a method called *language-guided foreground query selection* to adaptively select queries with respect to given text concepts as shown in Fig. 3 left part. Given the image features $\mathbf{O}$ and text features $\mathbf{T}$, we employ a lightweight module to predict the box and score for each feature:

$$\mathbf{E}^b = \mathsf{Head}(\mathbf{O}), \mathbf{E}^c = \mathsf{Sim}(\mathbf{O}, \mathbf{T}) \tag{5}$$

where Head is the box head. Then we select L_f top-ranked entries from $\mathbf{E}^b$ and $\mathbf{O}$ according to the scores in $\mathbf{E}^c$. These selected L_f image features and boxes are then fed to the decoder as the foreground queries (blue squares in Fig. 3). By selecting only the text-related tokens as decoder queries, we mitigate the problem of decoding irrelevant semantics and provide better query initialization. Such an adaptive way of proposing foreground queries enables our model to effectively transfer to novel vocabulary during test scenarios.

Learnable background queries. Different from foreground queries, we use learnable query embeddings for our background queries for two reasons. Firstly, query selection does not work well because the selected reference points often extend beyond large and non-convex background regions, leading to suboptimal results. Secondly, background stuff has a relatively smaller number of categories than the foreground, and a single image typically contains a few different stuffs (*e.g.*, "sky", "building"). As a result, using learnable queries for our model can sufficiently and effectively handle background stuff categories and generalize well to open-vocabulary settings. The background queries are marked by green squares in Fig. 3.

Comparison with previous works. In Fig. 5, we show a comparison between our approach and others on handling foreground and background. Mask2Former [5] and MaskDINO [29] treat foreground and background equally when conducting panoptic segmentation, resulting in suboptimal mask average precision (AP) for foreground objects compared to the same model solely trained on foreground classes (instance segmentation). Panoptic Segformer [32] separates foreground and background queries, but their background queries have fixed semantics, with each query corresponding to a pre-defined background category, limiting their ability to handle open-vocabulary categories. In contrast, our approach proposes foreground queries through a language-guided selection mechanism, and our background queries are fully learnable, eliminating the restrictions of a predefined vocabulary.

3.3. Bridge Data Gap: Conditioned Mask Decoding

Our ultimate goal is to bridge the data gap by using a single loss function to train multiple tasks, resulting in the

Table 1: Results for models trained on COCO without and with conditioned mask decoding. Models are evaluated on COCO and ADE20K validation set. "final" and "early" means the model at the final and early training stage, respectively. "Convert box into mask" means we directly convert the GT boxes into rectangle masks for evaluation.

Method	COCO Mask AP	ADE Mask AP
OpenSeeD (T)	45.1	8.6
OpenSeeD (T) w conditioned train & eval (final)	53.2	**46.4**
OpenSeeD (T) w conditioned train & eval (early)	42.2	14.8
Convert box into mask	16.2	25.4

following loss function [1]:

$$\mathcal{L}_{all} = \sum_{I,(\mathbf{c},\mathbf{m},\mathbf{b})\in\mathcal{D}} \mathcal{L}_m(\mathbf{P^m},\mathbf{m}) + \mathcal{L}_b(\mathbf{P^b},\mathbf{b}) + \mathcal{L}_c(\mathbf{P^c},\mathbf{c}) \tag{6}$$

Here, $\mathcal{D}$ represents the union of segmentation and detection datasets. However, the loss function requires mask annotations for detection data and box annotations for segmentation data, leading to a discrepancy in the granularity of spatial supervision between the two tasks. As we discussed earlier, we can easily convert an object mask m to a box $\hat{b}$, which augments the original segmentation data $\mathcal{D}_m = \{I_i, (\mathbf{c}_i, \mathbf{m}_i)\}_{i=1}^M$ into $\hat{\mathcal{D}}_m = \{I_i, (\mathbf{c}_i, \mathbf{m}_i, \hat{\mathbf{b}}_i)\}_{i=1}^M$. For detection data $\mathcal{D}_b$, however, we are only given coarse location (box) and category. Then an interesting question comes – *can we obtain its mask given these priors?* To address this problem, we resort to the segmentation data which contains rich mappings from label&box to mask, *i.e.*, $(c, b) \rightarrow m$ and propose conditioned mask decoding to learn the mappings as shown in Fig. 3 right-most part. Given the ground-truth concepts and boxes, $(\mathbf{c}, \mathbf{b})$, we employ the decoder to decode the mask:

$$\mathbf{P^m} = \mathsf{Dec}\left((\mathbf{t}, \mathbf{b}); \mathbf{O}\right) \tag{7}$$

where t is the text features extracted for the concepts. Based on Eq. (7), the question becomes, *can we learn from segmentation data a good mapping which generalizes well to detection data with different categories?*

Mapping Hypothesis Verification. To answer the question, we conduct a pilot study. We train a model which learns to decode masks conditioned on GT concepts and boxes on COCO [4], and then evaluate the conditioned decoding performance on ADE20K [58]. The results are shown in Table 1. Comparing the top two rows, we can find mask decoding conditioned on the GT concept and box significantly improves the quality (mask AP from 8.6 to 46.4), which even reaches a similar level to COCO (46.4 *v.s.* 53.2). These results indicate that our learned mask decoding generalizes well to a new dataset with novel categories. To further verify, we visualize decoded masks in Fig. 6.

Interactive Segmentation. The above study implies a new interface of image segmentation. Apart from segmenting an image from scratch, users can give a hint about the object location by drawing a box (click four points), and our *OpenSeeD* can generate its mask with fairly high quality. This capacity can potentially help accelerate the annotation of segmentation data, especially for those with boxes. We leave a comprehensive study on this as future work.

Conditioned Mask Decoding Training. Based on the verified hypothesis, we add all the GT boxes and labels as the conditioned queries to simultaneously learn foreground/background decoding and conditioned mask decoding, as shown in Fig. 3. It unifies all our tasks and enables *OpenSeeD* to learn more generalized conditioned decoding in the joint semantic space. Based on this, we can literally derive the pseudo masks $\hat{\mathbf{m}}$ for object detection data and obtain augmented $\hat{\mathcal{D}}_b = \{I_i, (\mathbf{c}_i, \hat{\mathbf{m}}_i, \mathbf{b}_i)\}_{i=1}^N$. Below we elaborate on how these pseudo masks are used for training.

Conditioned Mask Generation to Guide Detection Data. The trained conditioned mask decoding component can also be used to assist detection data as segmentation guidance. We propose two methods to utilize the generated mask to guide our model training, *online mask assistance* and *offline mask assistance*. For online mask assistance, we only train one model and generate the masks on the fly. Instead of directly using the generated masks as mask supervision, we use the masks to assist in matching predictions and GT instances because the mask quality is not strong enough for supervision. especially in the early stage (shown in Tab. 1 third row). As for offline mask assistance, we train our model with conditioned mask decoding until convergence and generate mask annotations for detection data. The annotated dataset can be used to train a segmentation model. Considering detection data only has instance-level annotations, the generated masks are expected to improve instance segmentation in both cases. More details about these two methods are discussed in the Appendix.

Comparison with Denoising Training. Compared with models [28, 29, 55] using denoising training (DN), *conditioned mask decoding* differs in two aspects. First, their design choices are different. DN adds noise to the GT boxes and labels for reconstruction, but our model learns to generate masks conditioned on the GT priors. Second, their design purposes are different. DN is designed to accelerate training convergence (understanding), while our method aims to generate masks for detection data (generation).

4. Experiment

4.1. Experimental Setup

Datasets and Settings. In our experiments, we jointly pretrain on two types of data, including panoptic segmentation and object detection. For panoptic segmentation, we use

[1] We do not differentiate foreground and background terms here, while in practice, foreground and background terms are divided as in Equation 4. In addition, this loss function is considered ideal and we are unable to achieve it in the online mask assistance method described later.

Table 2: **One suit of weights** for open-vocabulary segmentation on multiple datasets in a zero-shot manner. Our model is pre-trained on COCO and Objects365 data. "SEG" indicates segmentation data (COCO), "DET" indicates detection data (Objects365), and "ITP" indicates image-text pairs/referring/captioning data. The values in gray are supervised results. ⋆ X-Decoder (L) is not open-source, so we cannot evaluate its performance on LVIS.

Method	Training Data			ADE				Cityscapes			LVIS		BDD		SCAN-20		SCAN-41	SUN
	SEG	DET	ITP	PQ	mask AP	box AP	mIoU	PQ	mask AP	mIoU	mask AP	box AP	PQ	mIoU	PQ	mIoU	mIoU	mIoU
MSeg (B) [23]	✓	✗	✗	33.7	32.6	–	19.1	46.9	24.8	51.1	–	–	–	44.9	–	33.4	–	29.6
MDETR [20]	✗	✓	✓	–	–	–	–	–	–	–	–	24.2	–	–	–	–	–	–
LSeg+ (B) [26]	✓	✗	✗	–	–	–	18.0	–	–	–	–	–	–	–	–	–	–	–
ZegFormer (B) [16]	✓	✗	✗	–	–	–	16.4	–	–	–	–	–	–	–	–	–	–	–
OpenSeg (B) [12]	✓	✗	✗	–	–	–	21.1	–	–	–	–	–	–	–	–	–	–	–
OpenSeg (B) [12]	✓	✗	✓	–	–	–	26.4	–	–	–	–	–	–	–	–	–	–	–
MaskCLIP (L) [7]	✓	✗	✗	15.1	6.0	–	23.7	–	–	–	–	–	–	–	–	–	–	–
ODISE (H) [47]	✓	✗	✓	23.5	13.9	–	28.7	–	–	–	–	–	–	–	–	–	–	–
GLIP (T) [30]	✗	✓	✗	–	–	–	–	–	–	–	–	18.5	–	–	–	–	–	–
X-Decoder (T) [61]	✓	✗	✓	18.8	9.8	–	25.0	37.2	16.0	47.3	9.6	–	16.4	42.4	30.7	37.8	21.7	34.5
OpenSeeD (T)	✓	✓	✗	**19.8**	**14.1**	**17.0**	22.9	**37.3**	**26.2**	46.1	**19.4**	**21.8**	**17.2**	**44.8**	**39.7**	**45.1**	**25.2**	**39.0**
X-Decoder (L) [61]	✓	✗	✓	21.8	13.1	–	29.6	38.1	24.9	52.0	⋆	–	17.8	47.2	39.5	49.5	29.7	43.0
OpenSeeD (L)	✓	✓	✗	19.7	**15.0**	**17.7**	23.4	**41.4**	**33.2**	47.8	**21.0**	**23.0**	**19.4**	**47.4**	**42.2**	48.7	27.4	41.9

COCO2017 [33] with segmentation annotations (around 110k images). For object detection, we use Objects365 [42] (660k images for v1 and 1700k images for v2). We use Objects365v1 for training and ablating our tiny model and Objects365v2 only for training our large model. We evaluate our models on all tasks covered by pretraining, including semantic, instance, panoptic segmentation, and object detection. In particular, we benchmark on more than 60 datasets covering a wide range of domains on zero-shot segmentation and detection.

Implementation Details. We build on Mask DINO [29] to implement our model. Mask DINO is a unified detection and segmentation framework which simultaneously predicts box and mask. We follow [29] to use 300 latent queries and nine decoder layers for thing categories in instance segmentation and add 100 panoptic queries for stuff categories. For the visual backbone, we adopt pretrained Swin-T/L [35] by default. We also use Focal-T [49] in our ablation studies following [61]. For the language backbone, we adopt the pretrained base model in UniCL [50]. Particularly, our model only uses these pretrained backbones and does not use other image-text pairs or grounding data for pretraining [30, 61]. During pretraining, we set a minibatch for segmentation to 32 and detection to 64, and the image resolution is 1024 × 1024 for both segmentation and detection. During fine-tuning, we use 512 × 1024 for Cityscapes [6] and 640 × 640 for ADE20K [58] by default. Following the balanced sampling strategy in [50, 61], the segmentation data are always sampled for a consistent number of epochs, regardless of the total number of detection data. We use AdamW [36] as the optimizer. We pre-train our model on the joint dataset for 30 epochs. The learning rate is set to 0.0001, which is decayed at 0.9 and 0.95 fractions of the total number of steps by 10. *Unless otherwise specified, we use online mask assistance during our pretraining by default.*

4.2. Open-Vocabulary Benchmarking

After pretraining our *OpenSeeD* on COCO and Objects365, we evaluate it on a wide range of datasets in a zero-shot manner. Following [61], we cover six commonly used segmentation datasets, including indoor scenes (ADE20K [58]), outdoor scenes (Cityscapes [6]), and driving scenes (BDD100K [52]). In addition, we evaluate both segmentation and detection performance on LVIS [14]. We report PQ, mask AP, and mIoU for panoptic, instance, and semantic segmentation, respectively, and use box AP for detection. The results are shown in Table 2.

We first compare with previous works on segmentation tasks. Overall, our model achieves significantly better performance on instance segmentation and comparable performance for panoptic and semantic segmentation. Compared with state-of-the-art methods ODISE [47] and X-Decoder [61], *OpenSeeD* achieves **1.1** and **1.9** mask AP improvements on ADE20K, respectively. This gap is even larger on Cityscapes and LVIS. Our *OpenSeeD* outperforms X-Decoder by **10.2** and **8.3** mask AP with a tiny and large model on Cityscapes, respectively. On LVIS, we evaluate the mask AP with the released X-Decoder tiny model and the comparison shows **9.8** mask AP improvement with our *OpenSeeD* tiny model. These results indicate that the proposed joint learning method can effectively transfer the instance-level knowledge in detection data for instance segmentation. Compared with instance segmentation, both panoptic and semantic segmentation requires the segmentation of background stuff, which is fully absent in the detection data. Despite that, our *OpenSeeD* still outperforms X-Decoder for panoptic segmentation on 3 out of 4 datasets (except for ADE20K), and achieves comparable semantic segmentation performance. The results on three segmentation tasks suggest that detection data significantly bene-

Table 3: Task-specific transfer of *OpenSeeD* to different segmentation and VL tasks. We directly evaluate the COCO performance without finetuning. Note: "−" denotes the model does not have number reported or does not have the ability for the specific task. ⋆ means it is the test set results. The results in the bracket are trained with 1280×1280 image size. Note that the results of GLIPv2 and X-Decoder on COCO are fine-tuned while those of *OpenSeeD* are reported without fine-tuning.

Method	Type	ADE				Cityscapes			COCO			
		PQ	mask AP	box AP	mIoU	PQ	mask AP	mIoU	PQ	mask AP	box AP	mIoU
Mask2Former (T) [5]	Closed-set	39.7	26.4	28.8	47.7	63.9	39.1	80.5	53.2	43.3	46.1	63.2
Mask2Former (B) [5]		–	–	–	53.9	66.1	42.8	82.7	56.4	46.3	49.5	67.1
Mask2Former (L) [5]		48.1	34.2	36.4	56.1	66.6	43.6	82.9	57.8	48.6	52.1	67.4
OneFormer (L) [18]		48.6	35.9	–	57.0	67.2	45.6	83.0	57.9	48.9	–	67.4
MaskDINO (L) [29]		–	–	–	–	–	–	–	58.3	50.6	56.2	67.5
Pano/SegFormer (B) [45]		–	–	–	51.0	–	–	–	55.4	–	–	–
kMaX-DeepLab (L) [53]		48.7	–	–	54.8	–	–	–	58.1	–	–	–
GLIPv2 (T) [56]	Open-vocabulary	–	–	–	–	–	–	–	–	42.0⋆	–	–
GLIPv2 (B) [56]		–	–	–	–	–	–	–	–	45.8⋆	–	–
GLIPv2 (H) [56]		–	–	–	–	–	–		–	48.9⋆	–	–
X-Decoder (T) [61]		41.6	27.7	28.8	51.0	61.3	36.2	78.7	52.6	41.3	43.6	62.4
OpenSeeD (T)		**47.2**	**35.1**	**39.4**	**52.2**	**63.9**	**38.2**	**80.3**	**55.4**	**47.6**	**52.0**	**64.0**
X-Decoder (L) [61]		49.6	35.8	–	58.1	65.6	42.2	81.7	56.9	46.7	–	67.5
OpenSeeD (L)		**53.1 (53.7)**	**42.0 (42.6)**	**46.4 (46.9)**	**58.6 (58.4)**	**69.2**	**49.3**	**84.5**	**59.5**	**53.2**	**58.2**	**68.6**

Table 4: One suit of weights on the SeginW benchmark in a zero-shot manner. [Chunyl: How about we simple put the dataset name vertical?]

Model	Med.	Avg	Air-Par.	Bottles	Br. Tum.	Chicken	Cows	Ele.-Sha.	Eleph.	Fruits	Gar.	Gin.-Gar.	Hand	Hand-Metal	House-Parts	HH.-Items	Nut.-Squi.	Phones	Poles	Puppies	Rail	Sal.-Fil.	Stra.	Tablets	Toolkits	Trash	W.M
X-Decoder (T) [61]	15.2	22.7	10.5	19.0	1.1	12.0	12.0	1.2	65.6	66.5	28.7	7.9	0.6	22.4	5.5	50.6	62.1	29.9	3.6	48.9	0.7	15.0	41.6	15.2	9.5	19.3	16.2
OpenSeeD (T)	21.5	**33.9**	12.2	27.4	5.0	68.7	21.5	0.3	73.3	72.9	7.3	6.2	92.4	62.3	0.5	55.0	63.6	2.4	4.6	63.8	5.4	15.6	85.3	32.0	4.8	14.5	51.0
X-Decoder (L) [61]	22.3	32.3	13.1	42.1	2.2	8.6	44.9	7.5	66.0	79.2	33.0	11.6	75.9	42.1	7.0	53.0	68.4	15.6	20.1	59.0	2.3	19.0	67.1	22.5	9.9	22.3	13.8
OpenSeeD (L)	38.7	**36.1**	13.0	39.7	2.1	82.9	40.9	4.7	72.9	76.4	16.9	13.6	92.7	38.7	1.8	50.0	40.0	7.6	4.6	74.6	1.8	15.0	82.8	47.4	15.4	15.3	52.3

Table 5: One suit of weights on ODinW benchmark. Average and median AP across 35 datasets are reported for simplicity.

Model	Pretrain Data	Average	Median
MDETR [20]	GOLDG, REFC	10.7	3.0
GLIP-T [30]	Object365	11.4	1.6
OpenSeeD (T) (ours)	Object365, COCO	14.2	3.1
OpenSeeD (L) (ours)	Object365, COCO	**15.2**	**5.0**

fit the instance-level understanding while image-text pairs mainly augment the semantic understanding for semantic segmentation. In addition to segmentation, *OpenSeeD* also produces reasonably good detection performance. Compared with GLIP, our *OpenSeeD* (T) outperforms GLIP (T) (setting A) for zero-shot detection on LVIS (**21.8** *v.s.* 18.5), where both only use Objects365 as the pretraining detection dataset. *At last, we highlight that our model is the first one that can be pretrained with segmentation and detection data jointly and perform zero-shot transfer to both tasks.*

4.3. Direct and Task-Specific Transfer

After pretraining, our model can be directly transferred to downstream segmentation and detection tasks. In Table 3, we compare our model with both closed-set and open-vocabulary methods. Remarkably, our model achieves SOTA performance **59.5** PQ for COCO panoptic segmentation *without any further fine-tuning*. After data-specific fine-tuning, *OpenSeeD* establishes a new SOTA on ADE20K panoptic (**53.7** PQ) and instance segmentation (**42.6** AP) when trained with 1280 × 1280 image size. In addition, we also achieve a new SOTA on Cityscapes instance segmentation (**48.5** AP). These results indicate the jointly pretrained open-vocabulary model can also be well transferred to closed-set detection and segmentation.

Table 6: The performance of *OpenSeeD* on close-set segmentation. We train *OpenSeeD* with only COCO ('w/o DET') data to compare with the top-performed close-set segmentation model MaskDINO. We also train *OpenSeeD* on COCO and Object365 together to verify the ability to utilize detection data. * none of our designed components are used.

Method	ADE				COCO			
	PQ	mask AP	box AP	mIoU	PQ	mask AP	box AP	mIoU
MaskDINO-ResNet-50* [29]	–	–	–	–	53.0	44.3	48.8	60.6
OpenSeeD-ResNet-50* w/o DET	–	–	–	–	52.7	44.0	47.9	60.2
OpenSeeD-SwinT* w/o DET	16.0	9.9	10.7	18.6	54.0	45.6	49.0	62.1
OpenSeeD-SwinT	**19.8**	**14.1**	**17.0**	**22.9**	**55.2**	**47.3**	**51.9**	**63.7**
OpenSeeD-FocalT* w/o DET	15.0	8.7	9.2	17.5	54.0	45.1	47.5	62.5
OpenSeeD-FocalT	**18.5**	**13.4**	**15.6**	**21.7**	**55.4**	**47.3**	**51.0**	**63.4**

4.4. Segmentation and Detection in the Wild

To investigate the generalization ability of *OpenSeeD* for segmentation, we evaluate our model on more domain-specific datasets. We conduct a zero-shot evaluation of our model on the Segmentation in the Wild (SeginW) benchmark [61], which includes 25 datasets. As this benchmark focuses on instance segmentation, we report the average and median mAP of all the datasets following the common practice. The results in Table 4 indicate that combining detection supervision significantly improves the segmentation performance by more than **10** AP under the same setting.

To further study the object detection ability of *OpenSeeD*, we follow GLIP [30] to evaluate detection performance on Object Detection in the Wild (ODinW) benchmark [27]. It collects over 35 datasets and is closer to real-world scenarios. We report the average and median mAP of all 35 datasets. With the jointly pretrained model weights, we directly evaluate this challenging benchmark in a zero-shot manner. Under the same setting that only uses Objects365 as the detection data for training, our tiny model outperforms GLIP-T (setting A) by 2.8 AP in average.

Table 7: Ablation of the effectiveness of pseudo annotations in offline mask assistance for our open-vocabulary model. We evaluate the model performance on ADE20K and COCO. ”-anno” denotes with annotations and ”w/o anno” denotes without annotations.

Method	ADE				COCO			
	PQ	mask AP	box AP	mIoU	PQ	mask AP	box AP	mIoU
OpenSeeD-FocalT w/o anno	18.4	12.6	14.5	21.4	55.1	46.8	50.3	62.7
OpenSeeD-FocalT-anno	18.2	**14.1(+1.5)**	**16.2(+1.7)**	21.4	55.2	**47.5(+0.7)**	**51.3(+1.0)**	63.4
OpenSeeD-SwinT w/o anno	19.8	14.1	17.0	22.9	55.2	47.3	51.9	63.7
OpenSeeD-SwinT-anno	**20.4(+0.6)**	**14.8(+0.7)**	**17.1**	**24.0(+1.1)**	55.5	**47.9(+0.6)**	**52.4(+0.5)**	63.9

4.5. Ablation

Ablation on our basic components. We first ablate on our basic model to verify whether each basic component works as expected. We evaluate our model on the COCO closed-set panoptic segmentation task. The first two rows in Table 6 show that our model can achieve a comparable closed-set panoptic segmentation performance as MaskDINO using the same ResNet-50 backbone. It is also shown in the last four rows that our framework can significantly improve the segmentation performance by combining tasks and utilizing detection data across different backbones.

Ablation on the offline mask supervision. As we discussed earlier, the proposed conditioned mask decoding can be used either in online or offline manner. Here, to investigate the effectiveness of offline setting, we generate pseudo annotations with our large models and then use them to tune the tiny models. Concretely, we first generate pseudo masks on Objects365 conditioned on boxes with our *OpenSeeD* (L) model. Given the masks, we conduct experiments on two models, including the open-vocabulary model (*OpenSeeD*) and the closed-set model (MaskDINO) by using the pseudo masks for supervision. As shown in Table 7, when evaluating our models, we find that the mask AP and box AP are significantly improved. When conducting experiments on MaskDINO, we perform object detection for the setting without pseudo annotation and instance segmentation for the setting with pseudo annotations during pretraining. After pretraining, we do a zero-shot evaluation on COCO. We also use the pretrained model for fine-tuning. The results in Table 8 indicate that in both settings, pseudo-annotations improve performance. It is also shown that the box AP can also be improved accordingly with extra mask annotations. In addition, we are the first to report zero-shot segmentation performance on COCO which is comparable with the full-shot performance of other methods.

Table 8: Ablation of the effectiveness of pseudo annotations in offline mask assistance for close-set segmentation models. We conduct experiments with maskDINO pre-trained on Object365 with Mask DINO. The task is instance segmentation and object detection for the settings with or without pseudo annotations, respectively. ”O365+anno” denotes Objects365 data with pseudo-annotations of masks.

Method	Pre-training Dataset	COCO zeroshot		COCO finetune	
		mask AP	box AP	mask AP	box AP
GLIP A(Swin-T) [30]	O365	–	42.9	–	52.9
GLIP B(Swin-T) [30]	O365	–	44.9	–	53.8
MaskDINO-R50	O365	–	42.4	46.7	52.0
MaskDINO-R50	O365+anno	**41.4**	**44.3(+1.9)**	**48.5(+1.8)**	**54.3(+2.3)**

Table 9: Ablation of the decoupled decoding and the online mask assistance. We keep other settings the same and only remove one design at a time. For the setting without decoupled decoding, we let foreground and background queries be selected together from image features.

Method	ADE				COCO			
	PQ	mask AP	box AP	mIoU	PQ	mask AP	box AP	mIoU
OpenSeeD-SwinT	19.8	14.2	17.2	23.1	55.2	47.3	51.9	63.7
– online mask assistance	19.4	**13.6(-0.6)**	**16.7(-0.5)**	22.9	55.0	47.5	52.3	63.9
– decoupled decoding	19.0	**13.1(-1.1)**	**14.9(-2.3)**	22.4	55.0	**46.4(-0.9)**	**49.5(-2.4)**	63.9

Ablation on decoupled decoding and online mask assistance. In Table 9, we conduct experiments to show the effectiveness of our proposed components by removing them one at a time. In the second row, after removing online mask assistance, the instance mask and box performance on the open-segmentation dataset ADE20K drops by 0.6 AP and 0.5 AP, respectively. In the third row, when we remove the decoupled decoding, the performance of the mask and box is further impacted by a large margin (-1.1 AP and -2.3 AP for ADE mask and box). Both results suggest that the proposed techniques help to mitigate the gap between segmentation and detection data.

5. Conclusion

We have presented *OpenSeeD*, a simple open-vocabulary segmentation and detection framework, which jointly learns from different segmentation and detection datasets with a single model. To bridge the task gap between foreground objects and background stuff, we propose a decoupled decoding method with language-guided foreground query selection. We also jointly train a conditioned mask decoding task, which provides an interactive segmentation interface during inference and helps bridge data gap for detection data during training. The result indicates our unified model significantly improves open-segmentation while keeping a reasonable detection performance. The jointly pre-trained model can also be seamlessly transferred to improve close-vocabulary performance.

Limitations. In this work, we aim at exploring the potential of training an open-vocabulary model for both segmentation and detection. *OpenSeeD* does not utilize either referring/-grounding data or large-scale image-text pairs to further enrich our training data and semantic coverage. We leave a grander joint training to future work.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3

[2] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4974–4983, 2019. 3

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 3

[4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6

[5] 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. 3, 5, 8

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 2. sn, 2015. 7

[7] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022. 2, 3, 7

[8] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 3

[9] King-Sun Fu and JK Mui. A survey on image segmentation. *Pattern recognition*, 13(1):3–16, 1981. 3

[10] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Open vocabulary object detection with pseudo bounding-box labels. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 266–282. Springer, 2022. 2

[11] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021. 3

[12] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 2, 7

[13] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *Learning*, 2021. 3

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 7

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 3

[16] Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac, and Aida Nematzadeh. Decoupling the role of data, attention, and losses in multimodal transformers. *TACL*, 2021. 7

[17] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7031, 2022. 2, 3

[18] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. *arXiv preprint arXiv:2211.06220*, 2022. 3, 8

[19] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 3

[20] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 7, 8

[21] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 3

[22] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. *arXiv preprint arXiv:2209.15639*, 2022. 2

[23] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2879–2888, 2020. 7

[24] Shiyi Lan, Xitong Yang, Zhiding Yu, Zuxuan Wu, Jose M Alvarez, and Anima Anandkumar. Vision transformers are good mask auto-labelers. *arXiv preprint arXiv:2301.03992*, 2023. 3

[25] Shiyi Lan, Zhiding Yu, Christopher Choy, Subhashree Radhakrishnan, Guilin Liu, Yuke Zhu, Larry S Davis, and Anima Anandkumar. Discobox: Weakly supervised instance segmentation and semantic correspondence from box supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3406–3416, 2021. 3

[26] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 3, 7

[27] Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Yong Jae Lee, Houdong Hu, Zicheng Liu, et al. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *arXiv preprint arXiv:2204.08790*, 2022. 8

[28] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 3, 6

[29] Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and seg-

mentation. *arXiv preprint arXiv:2206.02777*, 2022. 2, 3, 5, 6, 7, 8

[30] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 2, 3, 7, 8, 9

[31] Wentong Li, Wenyu Liu, Jianke Zhu, Miaomiao Cui, Xian-Sheng Hua, and Lei Zhang. Box-supervised instance segmentation with level set evolution. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 1–18. Springer, 2022. 3

[32] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1280–1289, 2022. 3, 5

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 7

[34] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022. 3

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 7

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[37] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021. 3

[38] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021. 3

[39] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. *arXiv preprint arXiv:2212.04994*, 2022. 2

[40] 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. 2, 3

[41] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022. 3

[42] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 2, 7

[43] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5443–5452, 2021. 3

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[45] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 8

[46] 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. 3

[47] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. *arXiv preprint arXiv:2303.04803*, 2023. 3, 7

[48] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. *arXiv preprint arXiv:2302.12242*, 2023. 2

[49] Jianwei Yang, Chunyuan Li, and Jianfeng Gao. Focal modulation networks. *arXiv preprint arXiv:2203.11926*, 2022. 7

[50] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *CVPR*, 2022. 7

[51] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pretraining for open-world detection. 2022. 3

[52] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018. 7

[53] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *European Conference on Computer Vision*, pages 288–307. Springer, 2022. 8

[54] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021. 3

[55] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr

with improved denoising anchor boxes for end-to-end object detection, 2022. 3, 6

[56] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022. 2, 8

[57] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. 2, 3

[58] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5, 6, 7

[59] Chong Zhou, Chen Change Loy, and Bo Dai. Denseclip: Extract free dense labels from clip. *arXiv preprint arXiv:2112.01071*, 2021. 2

[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3

[61] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. *arXiv preprint arXiv:2212.11270*, 2022. 3, 7, 8

[62] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019. 3