Prompting Foundational Models for Omni-supervised Instance Segmentation

Arnav M. Das[1]*† Ritwick Chaudhry[2]* Kaustav Kundu[2] Davide Modolo[2]
[1]University of Washington, Seattle [2]AWS AI Labs

arnavmd2@uw.edu, {ritwic, kaustavk, dmodolo}@amazon.com

Abstract

Pixel-level mask annotation costs are a major bottleneck in training deep neural networks for instance segmentation. Recent promptable foundation models like the Segment Anything Model (SAM) and GroundedDINO (GDino) have shown impressive zero-shot performance in segmentation and object detection benchmarks. While these models are not capable of performing inference without prompts, they are ideal for omnisupervised learning, where weak labels are used to derive supervisory signals for complex tasks. In our work, we use SAM and GDino as teacher models and prompt them with weak annotations to create high-quality pseudomasks. These pseudomasks are then used to train student instance segmentation models, which do not require prompts at inference time. We explore various weak annotations, such as bounding boxes, points, and image-level class labels, and show that a student model can achieve roughly 95% of a fully-supervised model's performance while reducing annotation costs by 7×. We show the effectiveness of our approach on challenging instance segmentation benchmarks such as COCO [15], ADE20K [30], Cityscapes [9]. Our approach can be used to reduce annotation cost to train instance segmentation models, making it more accessible to a wider range of applications.

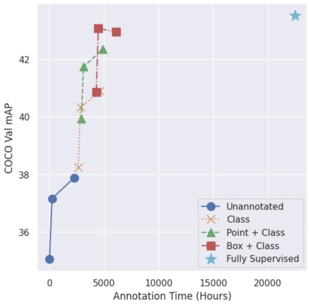

Figure 1. The results of training a Mask2Former [6] with pseudomasks derived from weak forms of annotations with a teacher model. We consider the settings where 0%, 1%, and 10% of masks are available, and are used to finetune the teacher.

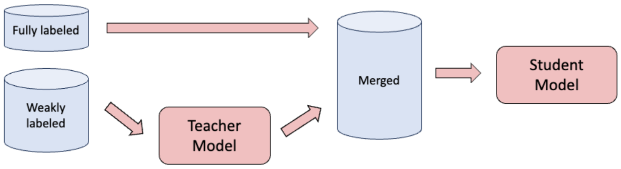

Figure 2. Here we show an overview of the omnisupervised setting. Annotations from the weakly labeled pool are converted to masks and merged with the fully labeled pool. The joint dataset is used to train a student model. The fully labeled data is also used to finetune the teacher model to further improve performance (not shown).

1. Introduction

Instance segmentation is an important computer vision task essential for applications such as autonomous driving and robotics. However, the high cost of instance segmentation annotations hinders the ability to create large-scale annotated datasets. While omni-supervised learning has been successfully applied in object detection [22, 26] to mitigate annotation costs using weaker forms of annotation (such as points, image tags, etc.), extending this approach to instance segmentation, where annotations are even more expensive, remains a crucial endeavor. Omni-supervised learning approaches are built on the student-teacher model [18, 24], where the teacher's predictions are filtered by weaker annotations to provide accurate pseudo-labels for training the student model. This method has proven effective in improving object detection performance, achieving a better cost-accuracy trade-off. In this work, we show that the instance mask predictions from the teacher model can be refined with weaker forms of annotation (such as image tags, points, etc.) to generate accurate pseudo-masks for training the student model, thereby improving instance segmentation performance.

In recent years, alongside the advancement of omni-supervised learning, foundational models (FMs) have emerged with remarkable zero-shot capabilities. Segment Anything Model (SAM) [14] was recently introduced as a FM for image segmentation. SAM was shown to gener-

*These authors contributed equally to this work.
†Work done during an internship with AWS AI Labs.

ate accurate segmentation masks conditioned on geometric prompts (such as points or bounding boxes). Similarly, Grounded DINO (GDino) [16] utilizes text queries to produce precise bounding boxes for instances corresponding to the query.

In this study, we investigate the use of FMs as teacher models capable of generating accurate pseudo-masks for training the student model in instance segmentation tasks. However, SAM's reliance on geometric prompts like points and bounding boxes limits its ability to identify pseudo-masks with both high precision and recall. Ambiguities arise, for instance, when a point prompt on a person could refer to a particular body part, a single individual, or a group. Similar ambiguities exist with prompts derived from image tags. To mitigate such ambiguities and enhance the quality of pseudo-labels, we introduce two ways to fuse more information to the prompts. In addition to geometric details, semantic information, such as class labels, serves as a complementary signal for the prompts. We leverage CLIP [21] to convert class labels into semantic embeddings and employ a class-aware module to merge the semantic prompt with the geometric prompt, thereby reducing ambiguity for the SAM decoder to produce accurate pseudo-masks. When bounding box annotations are unavailable, a merged point and semantic prompt may struggle to disambiguate overlapping instances of the same category due to occlusion. The lack of information about the extent of the object, increases ambiguity for the SAM decoder to predict the correct pseudo-mask. In such cases, we use GDino to convert the class labels into bounding boxes, which can refined the prompt to the SAM decoder in such cases.

The contributions of this work can be summarized with the following:

- We propose new prompting strategies for foundation models that allow us to generate high quality pseudo-masks from weaker forms of annotation.
- We propose new prompting strategies for SAM that allow us to generate high quality pseudo-masks from weaker forms of annotation.
- We demonstrate through extensive experiments that near fully supervised performance can be achieved on the COCO [15], ADE20K [30] and Cityscapes [9] datasets as shown in Figure 1.

2. Related Works

Adapting SAM. Some concurrent works have explored adapting SAM to other domains, but consider settings where geometric prompts are assumed to be available at inference time [5, 19, 27]. Other works assess or improve the performance of SAM in the few-shot setting [17, 29], and are complementary to our framework. New techniques that improve SAM's label efficiency can be used to improve the teacher model in our framework, and further improve the quality of pseudolabels when very few fully labelled samples are available.

Annotation Type	Time per Image		
	COCO	ADE20K	Cityscapes
none	0s	0s	0s
class	80s	100s	8s
point + class	88.7s	110.9s	24.7s
box + class	130.4s	163.3s	123.5s
mask + class	684.8s	859.5s	1341.7s

Table 1. We report the average labeling time for a single image in the COCO dataset with every type of annotation considered in this work. We follow [4, 7, 15, 20, 26] for the time estimates, and include detailed calculations in the Appendix.

Weakly Supervised Instance Segmentation. Given the high cost of obtaining annotations for masks, many works have sought to leverage other forms of annotations as weak labels for segmentation. [10, 23] propose approaches that use CLIP [21] to perform semantic segmentation, though applications to these towards instance segmentation have not been explored. [2, 3, 31, 32] have used image level labels to generate pseudomasks. These approaches typically use the Class Activation Mask (CAM) to identify salient regions of the image, and apply some sort of post-processing/refinement step to convert attribution maps to pseudomasks. Other approaches [8, 12, 13, 25] use bounding box annotations as weak labels for instance segmentation. [7] considers using both bounding box annotations and several point annotations to perform instance segmentation and are the first to demonstrate competitive performance with fully supervised instance segmentation on the COCO dataset. Instance segmentation with point annotations alone, however, remains underexplored though some works consider semantic segmentation with point supervision [4]. Unlike our framework, prior weakly supervised instance segmentation approaches do not leverage the impressive zero-shot capabilities of foundation models such as SAM. Moreover, while previous weakly supervised instance segmentation techniques are tailored towards leveraging one particular type of weak annotation type, we consider an omnisupervised setting where many forms of weak annotations are considered.

Omnisupervised Object Detection. Omnisupervised object detection approaches seek to use any useful form of annotation to maximize detection performance [22, 26]. Our setting is most similar to that of Omni-DETR proposed by [26], where a small fraction of the dataset is assumed to be fully annotated with bounding boxes, and the remainder of the dataset is labelled with some form of weak annotation (point, class tag, etc.). Omni-DETR proposed to generate pseudolabels using a bipartite matching based fil-

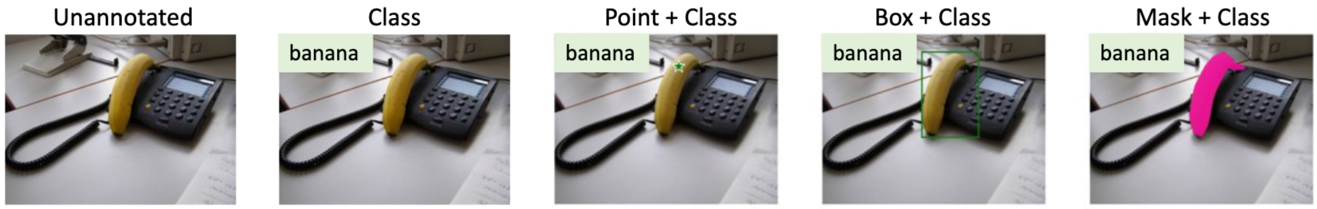

Figure 3. This figure displays the types of per-instance annotations considered in this work. The mask + class annotation represents the strongest form of annotation, and is required for fully supervised instance segmentation.

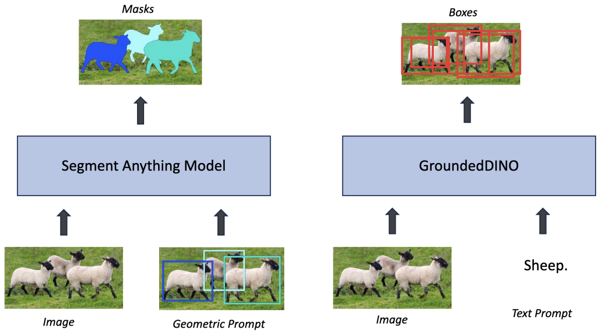

Figure 4. Here we display the two promptable foundation models we use in this work. SAM (left) produces a mask for a given geometric prompt (box or point) in an image. GDino (right) produces a set of boxes given an image and a text prompt.

tering mechanism. However, unlike our approach, Omni-DETR's framework can only be applied to object detection models that use a DETR style bipartite matching loss, and cannot be adapted to any arbitrary architecture. Moreover, our approach specifically considers instance segmentation whereas [26] only considers object detection.

3. Method

Promptable foundation models for various dense prediction tasks have been recently released [1, 14, 16], and are capable of achieving strong zero-shot performance when provided with free-form text or weak geometric prompts. Among them, those of interest to this paper are: (i) Segment Anything Model (SAM) [14], which is a foundation model for image segmentation; it takes as input an image and geometric prompts (either a bounding box or a set of points) and generates masks; and (ii) GroundedDINO (GDino) [16], which is a foundation model for object detection; it takes raw text as input prompts and generates bounding boxes enclosing the concept(s) specified in the text prompt.

Our work seeks to use these foundation models to generate pseudomasks for instance segmentation in a setting where we assume the presence of human generated masks on a small proportion of the training set, and weaker forms of annotations on the remaining subset. Our pipeline consists of two parts: (i) finetuning SAM on a small number of fully annotated images and (ii) enhancing whatever form of weak annotation is available for a particular image with GDino predictions to construct a high quality prompt for finetuned SAM. We refer to the combination of promptable foundation models that are used to produce pseudomasks as the teacher network. A separate student network is then trained on a combination of human annotated and teacher annotated masks.

In the next sections we first present our teacher design (sec. 3.1) and later the different types of weaker supervisions that we consider in this paper, along with our pipelines to prompt GDino and SAM, accordingly. In details, we consider the case where no extra annotations are added (sec. 3.2), when only the class label is provided for the whole image (sec. 3.3), when the class label is provided, along with a point on the object (sec. 3.4) and finally the case when the class labels is provided along and a bounding box enclosing the object (sec. 3.5).

3.1. Teacher Architecture

We prompt SAM to produce the pseudomasks from weak forms of annotation, and use models such as CLIP and GDino to enhance prompt quality. We also modify and finetune SAM to support class prompts. Note that not all of the components are necessary for each form of annotation (e.g GDino is not necessary when ground truth boxes are available). The overview of our teacher model is shown in Figure 5, and discussed in more detail below.

Combining GDino and SAM. We use GDino's predicted boxes as inputs to SAM's prompt encoder. However, GDino predicts a large set of boxes, many of which are noisy or redundant. Therefore, we use a series of heuristics including filtering and/or matching to curate the boxes to use as the actual prompts to SAM. If weak geometric prompts (i.e. points) are available, then we design a two-stage prompt refinement procedure, where first only the points are used as prompts to SAM to generate pseudomasks. These pseudomasks are then matched with GDino outputs, and then conjunctively used as refined prompts to SAM. If no geometric prompts are available, we use filtering techniques such as confidence thresholding and NMS. These techniques form the core of our approach, since the quality of pseudomasks from SAM are heavily dependent on the quality of the prompts.

Class-Aware SAM. The currently released versions of SAM do not support text prompts. In our setting, the class information can be extremely useful in defining the extents of the objects. We therefore train a ControlNet style module to incorporate class information [28] into SAM, which is then merged with the geometric prompts. Specifically, we create two copies of the prompt encoder where one is kept frozen and the other is trained, denoted as $\mathcal{F}_{locked}$ and $\mathcal{F}_{train}$ respectively. Then, given a geometric prompt P_{geom} and a class prompt P_{cls} the following is used to recover the final prompt embedding Z_{merged}:

$$Z_{text} = \mathcal{F}_{train}(P_{geom}) + CLIP_{text}(P_{cls}) \quad (1)$$

$$Z_{merged} = \mathcal{G}_{zero}(Z_{text}) + \mathcal{F}_{locked}(P_{geom}) \quad (2)$$

where $CLIP_{text}$ is a pretrained CLIP text encoder [21] and $\mathcal{G}_{zero}$ is a fully connected layer whose weights are initialized to all zeros following [28]. Z_{merged} is then used as an input to the SAM decoder.

We leverage ground truth masks available on the small fully annotated subset (when available) to train this lightweight module. The importance of this component has been highlighted through an ablation experiment in Table 5.

3.2. Setting 1: No Annotations

In the case where we want to generate pseudomasks for unannotated images, we prompt GDino, apply filtering to recover a set of viable boxes, and use the boxes as prompts to a version of SAM that is finetuned on the available annotated data. This is similar to what is done in [1]. We describe each step in detail below:

Teacher Training. We first finetune SAM on all available fully annotated images, using only boxes and classes as prompts. The boxes are generated from the ground truth masks.

Recovering GDino Boxes. Assuming there are K classes in a given dataset, we use the following template: *[Class 1]. [Class 2]. ...[Class K].* to generate a text prompt for GDino. In the template, *[Class i]* corresponds to the name of the i'th class in the dataset. This prompting approach assumes that we have access to the label space of the dataset, but have no access to image specific information. We apply confidence filtering and NMS to suppress extraneous boxes.

Pseudomask Generation. Pseudomasks are generated by passing the GDino boxes through the finetuned SAM that was trained in the initial step. GDino class predictions are used as the class prompt in this case.

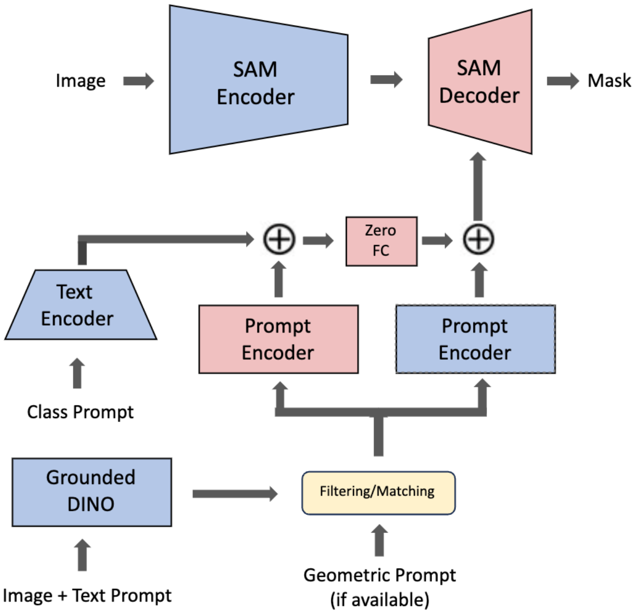

Figure 5. Here we show the generalized teacher model architecture used for all experiments. Modules in red are finetuned, while modules in blue are frozen. Note that the class prompt and text prompt are distinct: the class prompt is provided at an instance level whereas the text prompt is provided at an image level.

3.3. Setting 2: Class Annotations

When class annotations are present, we follow a procedure very similar to the no annotation case to generate pseudomasks. The key difference is that we modify the text prompt used for GDino, so that only the classes present in a given image are used.

3.4. Setting 3: Class Annotations + Point

In this section, we consider the setting where we have a small number of fully annotated images and a large number of images where each instance is annotated with a single positive point selected at random and a class tag. To produce masks from point annotations, we have three steps: 1) teacher training, 2) prompt refinement, and 3) mask generation.

Teacher Training. For this step, we simply finetune SAM on all available fully annotated images. We train two teacher models: one that is trained on only point prompts (P-SAM) and another that is trained on both box and point prompts (BP-SAM). We then generate an initial set of pseudomasks using P-SAM from ground truth point prompts.

Prompt Refinement. We find that P-SAM produces very noisy masks with such weak annotations even after tuning, so we seek to leverage GDino to produce boxes that we can use to prompt BP-SAM.

Algorithm 1: Prompt Refinement

```
Input: Predicted boxes $\mathcal{B}_{dino} \in \mathbb{R}^{N \times 4}$, confidence
       scores $\mathcal{C}_{dino} \in \mathbb{R}^{N}$, predicted labels
       $\mathcal{Y}_{dino} \in \mathbb{R}^{N \times K}$, ground truth points
       $\mathcal{P}_{gt} \in \mathbb{R}^{N' \times 2}$, SAM predicted boxes
       $\mathcal{B}_{sam} \in \mathbb{R}^{N' \times 4}$, ground truth labels
       $\mathcal{Y}_{gt} \in \mathbb{R}^{N'}$
Parameter: $\alpha$, $\beta$, $\tau_c$.
Output: Merged boxes $\mathcal{B}_{merged} \in \mathbb{R}^{N' \times 4}$
Initialize cost matrix $C \in \mathbb{R}^{N \times N'}$ to all zeros;
for $i = 1$ to $N$ do
    if $\mathcal{C}_{dino}[i] < \tau_c$ then
        $C_{ij} \leftarrow \infty$ ;
        continue;
    end
    for $j = 1$ to $N'$ do
        if $\mathcal{Y}_{dino}[i] \neq \mathcal{Y}_{gt}[j]$ or
          PointNotInBox($\mathcal{P}_{gt}[j], \mathcal{B}_{dino}[i]$) then
            $C_{ij} \leftarrow \alpha$
        end
        iou$_{ij}$ ← ComputeIOU($\mathcal{B}_{dino}[i], \mathcal{B}_{sam}[j]$)
         $C_{ij} \leftarrow C_{ij} - \text{iou}_{ij} - \beta \mathcal{C}_{dino}[i]$
    end
end
Solve $\sigma^* \leftarrow \text{argmin}_{\sigma} \sum_{j \in N'} C_{\sigma(j)j}$ with Hungarian
  matching algorithm;
Initialize $\mathcal{B}_{merged}$ ;
for $j = 1$ to $N'$ do
    if $\mathcal{Y}_{dino}[\sigma(j)] \neq \mathcal{Y}_{gt}[j]$ or
      PointNotInBox($\mathcal{P}_{gt}[j]$, $\mathcal{B}_{dino}[\sigma^*(j)]$) then
        $\mathcal{B}_{merged}[j] \leftarrow \mathcal{B}_{sam}[j]$
    end
    else
        $\mathcal{B}_{merged}[j] \leftarrow \mathcal{B}_{dino}[\sigma^*(j)]$
    end
end
return $\mathcal{B}_{merged}$
```

For a given image, GDino outputs box proposals $\mathcal{B}_{dino} \in \mathbb{R}^{N \times 4}$, corresponding confidence scores $\mathcal{C}_{dino} \in \mathbb{R}^{N}$, and class predictions $\mathcal{Y}_{dino} \in \mathbb{R}^{N \times K}$. We are provided with ground truth point annotations $\mathcal{P}_{gt} \in \mathbb{R}^{N' \times 2}$, and ground truth labels $\mathcal{Y}_{gt} \in \mathbb{R}^{N' \times K}$, where N' is the total number of instances in an image. We also have the teacher generated boxes from P-SAM, $\mathcal{B}_{sam} \in \mathbb{R}^{N' \times 4}$. Since typically $N' \neq N$, we propose a novel strategy to merge $\mathcal{B}_{sam}$ and $\mathcal{B}_{dino}$ to select as prompts.

The procedure merges the two sets of bounding boxes to create $\mathcal{B}_{merged} \in \mathbb{R}^{N' \times 4}$ is shown in Algorithm 1, where we use Hungarian matching to assign GDino boxes to instances and replace invalid GDino boxes with SAM boxes.

Pseudomask Generation. In this step, we simply use $\mathcal{B}_{merged}$ and $\mathcal{P}_{gt}$ to prompt the BP-SAM teacher from step 1 and use the corresponding masks as pseudolabels.

3.5. Setting 4: Class Annotations + Bounding Box

In the setting where box + class annotations are available, GDino is unnecessary. In this case, a teacher is trained on box prompts, and the remaining boxes are fed into SAM directly.

4. Experiments

4.1. Experimental settings

Implementation details. We now dive deep into the parameters and settings of the components of our approach.

Teacher Training. The SAM decoder and prompt encoder are the only components that are finetuned in all settings. We use the class-aware components in all settings except for the case when box + class annotations are available. We use a combination of cross entropy loss and dice loss for our loss as done in [6]: $\mathcal{L}_{mask} = \lambda_{ce}\mathcal{L}_{ce} + \lambda_{dice}\mathcal{L}_{dice}$. We use $\lambda_{ce} = 5$ and $\lambda_{dice} = 5$. Also following [6], we compute the loss on a subset of 112×112 points as opposed to using the full mask to improve training efficiency. We train the teacher on either 1% or 10% of the data, and both models are trained for the same number of steps (not epochs), following [11]. We train all models with a batch size of 32 for 36874 steps (equivalent to 10 epochs on 10% of the data), with the AdamW optimizer. An initial learning rate of 10^{-5} is used, and is dropped to 10^{-6} at step 27655. Each teacher model with the specs provided is trained on 8 Nvidia A10 GPU's.

Pseudomask Generation. To generate pseudomasks, we use a confidence threshold of 0.3 and an NMS threshold of 0.9 using GDino confidence scores for the no annotation and class annotation cases. For the point annotation case, we use $\alpha = 5$, $\beta = 2$, and $\tau_c = 0.1$. For images that are fully annotated, we do not use pseudomasks and only use ground truth masks.

Student Training. We fix the student training procedure in all of our settings, as we did with the teacher model. For the student model, we use a Mask2Former model with a ResNet-50 backbone [6] trained for a total of 50 epochs with batch size 32 on 8 Nvidia A10 GPU's. The decoder architecture, loss function, post-processing steps and all associated hyperparameters are kept identical to what is reported in [6].

Datasets. We evaluate the efficacy of our approach on three widely used segmentation datasets, namely MS-

% Mask	Weak Annotation Type	COCO AP	COCO APs	COCO APm	COCO APl	ADE20K AP	ADE20K APs	ADE20K APm	ADE20K APl	Cityscapes AP	Cityscapes APs	Cityscapes APm	Cityscapes APl
0%	none	35.1	17.7	39.1	54.9	9.5	2.8	11.4	19.6	17.3	4.5	16.6	36.0
	class	38.2	20.2	42.5	57.8	19.5	7.6	23.1	34.0	26.4	5.5	20.0	47.5
	point + class	**39.9**	**21.1**	**44.3**	**58.7**	23.3	9.3	**26.4**	38.0	22.1	4.4	18.0	39.0
	box + class	**40.9**	**22.5**	**44.5**	**60.2**	**24.3**	**9.5**	**27.3**	40.2	30.5	7.1	26.3	**56.0**
1%	none	37.2	18.4	40.5	58.0	9.6	3.1	11.1	18.7	18.4	5.4	19.2	35.7
	class	**40.3**	20.3	**44.0**	**61.1**	20.6	6.5	23.3	36.6	27.7	5.3	23.7	48.8
	point + class	**41.7**	**21.0**	**44.9**	**62.7**	**24.5**	8.9	**27.3**	40.0	28.9	4.5	26.2	53.8
	box + class	**43.1**	**21.9**	**46.5**	**63.6**	**25.4**	9.2	**28.4**	**42.3**	31.7	**7.8**	27.5	**56.8**
10%	none	37.9	19.0	41.6	**59.0**	11.9	3.2	13.9	23.8	21.2	5.8	20.8	41.1
	class	**40.9**	**20.7**	**44.5**	**62.1**	21.5	7.6	25.0	37.6	28.7	6.7	24.0	53.1
	point + class	**42.2**	**21.4**	**45.3**	**63.2**	**25.2**	**9.9**	**28.0**	**41.9**	31.7	6.6	27.5	**57.9**
	box + class	**42.9**	**22.5**	**46.3**	**64.2**	**26.5**	**10.3**	**29.0**	**44.3**	**34.6**	7.6	**30.0**	**60.6**
100%	-	43.5	23.0	46.6	65.0	26.7	10.4	29.1	45.0	35.6	8.7	32.6	60.0

Table 2. **Omnisupervised results on COCO, ADE20K and Cityscapes**. All results report the performance of a Mask2Former student model with a ResNet-50 backbone. Settings that achieve $>$ 90% of the AP attained by a fully supervised model are shown in **bold**.

Method	Per Instance Annotation	Backbone	Segmentation Style	AP	APs	APm	APl
BoxInst [25]	box + class	R-50	CondInst	32.1	15.6	34.3	43.5
BoxTeacher [8]	box + class	R-50	Mask-RCNN	35.0	19.0	38.5	45.9
Pointly-Supervised IS [7]	box + class + 10 Points	R-50	Mask-RCNN	36.9*	-	-	-
SAM Pseudolabels (ours)	box	R-50	Mask2Former	**40.9***	22.5*	44.5*	**54.2***
BoxInst [25]	box + class	R-101	CondInst	35.0	17.1	37.2	48.9
BoxTeacher [8]	box + class	R-101	Mask-RCNN	37.6	16.9	38.7	52.1
Pointly-Supervised IS [7]	box + class + 10 Points	R-101	Mask-RCNN	38.5*	-	-	-
BoxTeacher [8]	box + class	Swin-B	Mask-RCNN	40.6	**23.4**	**44.9**	**54.2**

Table 3. **Comparisons to Baselines on COCO**. We compare the performance of a student model trained on zero-shot SAM pseudolabels alone, against previous SOTA weakly supervised instance segmentation models. Numbers marked with an asterisk (*) are reported on COCO val2017, while all other numbers are reported on COCO test2017. R-50 and R-101 correspond to the ResNet-50 and ResNet-101 architectures respectively.

COCO [15] (80 ”things” categories), ADE20K [30] (100 ”things” categories) and Cityscapes [9] (8 ”things” categories). Following the experimental setting of [26], we randomly sample 1% and 10% of the training images of each datasets as “fully-supervised”. The remaining subsets are assumed to be annotated with weaker form of annotations. We evaluate the trained student models on the validation sets of the respective datasets.

4.2. Main Results

We compare the performance of our Mask2Former with a ResNet-50 backbone trained on images with (pseudo)-masks derived from several different forms of weak annotations in Table 2. We observe that when masks are available for a small proportion of images, we can attain near fully supervised performance while drastically reducing annotation costs through weaker forms of annotations. In particular, we observe that using boxes as the main form of annotation with masks only on 10% images can lead to more than 97% of the AP obtained by a fully supervised model across the three datasets. Furthermore, with masks on only 1% of the images, the student can achieve around 99%, 95% and 90% relative to the fully supervised student on COCO, ADE20K and Cityscapes respectively. Notably, these result in cutting down the annotation costs by at least 5 times.

Using points can lead to even higher savings in the annotation cost (more than 7x for COCO and ADE20K and larger than 50x for Cityscapes). The powerful combination of the foundation models can enable students to achieve 97%, 95% and 89% of the fully supervised performance. on the three datasets respectively, with only 10% of the images being fully annotated, with the rest having only one point annotated for each instance. With only 1% of the images having masks, the student models can correspondingly achieve 96%, 92% and 81%.

With even weaker form of annotations such as only class information without any localization information or in a completely unsupervised manner, the student is still able to achieve modest performance relative to the fully supervised scenario. In the setting with only class information and no ground truth masks, the student can achieve 73% to 88% of the fully supervised performance, ranging over different datasets. With the availability of masks on 10% of the training images, the student performance correspondingly ranges from 80% to 94%.

We observe that significant performance gains are made when we annotate 1% of the images with masks when compared to when there are no masks available. Fine-tuning the SAM model along with the learning the ability to incorporate class-aware information is highly effective for generating high quality pseudo-labels, while requiring only a small set of fully annotated samples. This is especially evident with the performance on Cityscapes, where 1% of the samples amount to less than 30 images. Fine-tuning on those 30 images, can lead to the student performance improving by over 30% (+6.8 points) in terms of mAP. This behavior can be attributed to zero-shot SAM often under-segmenting and over-segmenting instances, especially with minimal localization information such as points. This is qualitatively demonstrated in Figure 6.

We also compare some of the SOTA weakly supervised methods with a Mask2Former student model trained on zero-shot SAM's pseudolabels in Table 3. We find that using foundation models as teachers, even without finetuning, provides a significant improvement over previous approaches. The only approach that comes close to the results presented in this work is BoxTeacher with a Swin-B backbone, which has about 3× more parameters than ResNet-50 backbone used by the student model we evaluate.

4.3. Ablation study

Does Finetuning Help? We decouple the effects of simply adding more ground truth masks to the training set and finetuning SAM. We find that adding ground truth masks alone without finetuning SAM does not improve performance, in terms of AP on COCO val2017 that is attained by a student model (Table 4). Qualitatively (Figure 6), we observe that finetuning SAM allows the teacher model to pick up class information that is specific to the dataset. SAM is trained without any class information, so it has a tendency to over-segment images with point prompts and flip the foreground/background with box prompts. Finetuning on a small number of these points allows the model to resolve the ambiguities present in the weaker geometric prompts.

Does SAM need Class Prompts? We now investigate the effect of class-aware SAM. We find that having class information incorporated into the prompts helps SAM resolve ambiguities, particularly in the point prompt case. In particular, we notice that having access to class information alleviates under and over segmentation, especially in the case of weak geometric prompts. This has been shown qualitatively in Figure 8. The improvements in teacher mIOU scores is shown in Table 5. We find modest improvements with using class-aware SAM with box prompts.

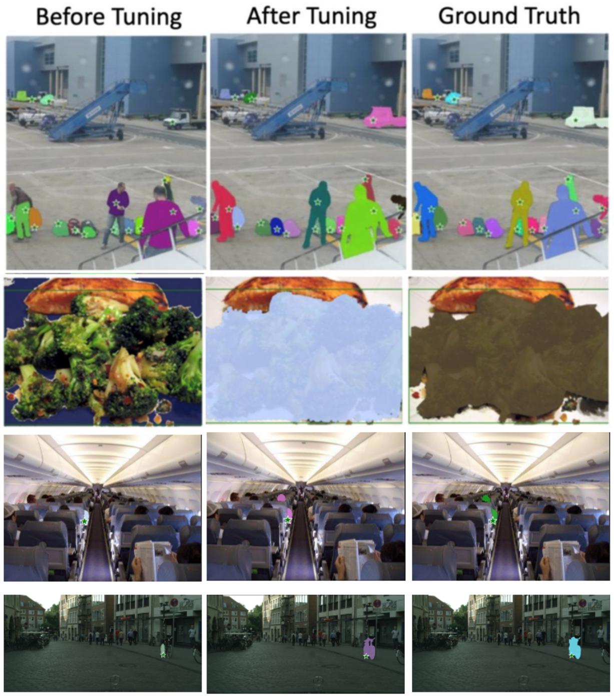

Figure 6. Effect of finetuning SAM on 1% of training data. The first two rows are generated by using point prompts and box prompts respectively on COCO val2017. The third row is on ADE20K while the last row is from Cityscapes. In these cases, SAM is able to resolve ambiguities in the prompt after tuning on a small amount of data.

Weak Anno. Type	0% Mask	1% Mask	10% Mask
none	35.1	35.4	36.6
class	38.2	38.3	39.2
point + class	39.9	40.4	40.0
box + class	40.9	40.8	41.3

Table 4. Student model AP when zero-shot SAM is used to generate pseudomasks. Adding fully annotated images alone brings only minor improvements.

Prompt Refinement. Since SAM natively supports point prompts, it is natural to ask if it is necessary to include the prompt refinement step which selects GDino boxes based on P-SAM's predictions. In Table 5, we observe that there is a considerable jump in teacher mIOU when GDino is incorporated in both. We also visualize how P-SAM's predicted boxes are improved when they are enhanced with GDino's boxes in Figure 7.

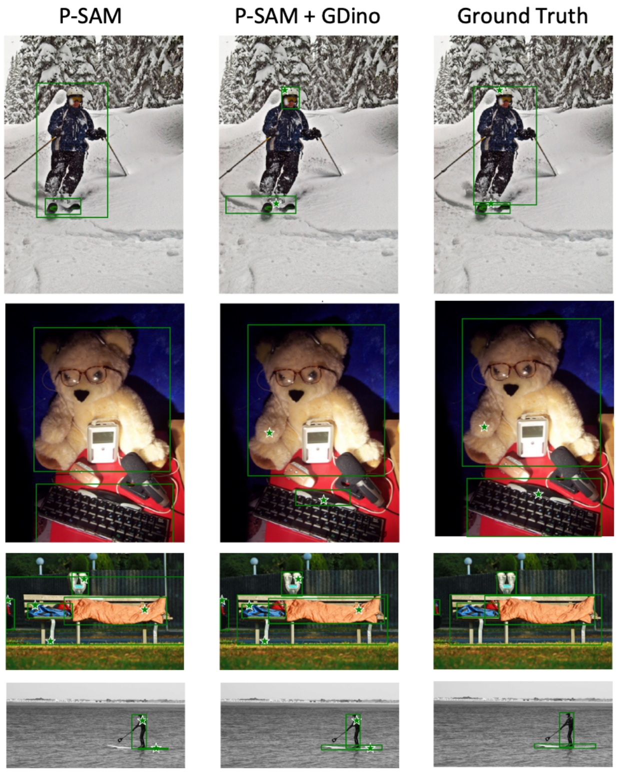

Figure 7. Effect of applying prompt refinement to P-SAM's predictions by merging its boxes with GDino boxes. With the use of matching, we can construct higher quality prompts for SAM.

% Mask	Class Aware?	Point	Point + GDino	Box
1%	×	64.0	65.7	80.7
	✓	67.5	70.0	80.8
10%	×	64.9	66.1	81.3
	✓	69.0	70.7	81.4

Table 5. Teacher performance (mIOU) for different prompts, demonstrating the utility of incorporating the class-aware module. With point prompts, before and after merging with GDino boxes, we realize a clear performance boost. The increase in performance by including class prompts with boxes is minor.

5. Conclusion and Future Work

In this work, we demonstrate the potential of promptable foundation models to significantly reduce the annotation cost for instance segmentation. We propose an architecture agnostic framework that leverages these foundation models to generate pseudolabels, and demonstrate that far weaker forms of annotation can be used to train a student model that can attain near-fully supervised performance. Future research directions may include the following:

- We consider only one form of weak annotation in each of our settings, but different instances may be suited for different forms of annotations. This was briefly investigated in the context of object detection by [26], but devising a principled method of determining the optimum annotation type per instance is yet to be explored.
- All of our modifications improve the teacher component, while fixing the training strategy. However, using techniques that improve the student model's robustness to label noise during training could significantly improve our results.
- Techniques that enhance SAM's few shot segmentation performance would greatly benefit omnisupervised segmentation, by further reducing our framework's dependency on ground truth masks.

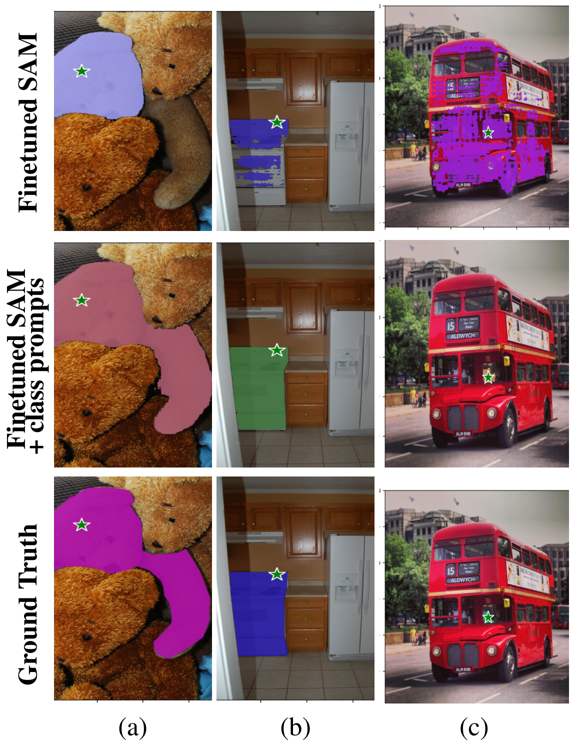

Figure 8. Effect of using class aware prompts on SAM (fine-tuned with point + class annotations). The class labels are (a) teddy bear, (b) oven, (c) person.

References

[1] Grounded segment anything. https://github.com/IDEA-Research/Grounded-Segment-Anything, 2023. 3, 4

[2] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[3] Aditya Arun, C. V. Jawahar, and M. Pawan Kumar. Weakly supervised instance segmentation by learning annotation consistent instances, 2020. 2

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li

Fei-Fei. What's the point: Semantic segmentation with point supervision, 2016. 2, 1

[5] Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam fails to segment anything? – sam-adapter: Adapting sam in underperformed scenes: Camouflage, shadow, medical image segmentation, and more, 2023. 2

[6] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *CoRR*, abs/2112.01527, 2021. 1, 5

[7] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2617–2626, 2022. 2, 6, 1

[8] Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Qian Zhang, and Wenyu Liu. Boxteacher: Exploring high-quality pseudo labels for weakly supervised instance segmentation, 2023. 2, 6

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1, 2, 6

[10] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Maskclip: Masked self-distillation advances contrastive language-image pretraining, 2023. 2

[11] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023. 5

[12] Cheng-Chun Hsu, Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2

[13] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Weakly supervised semantic labelling and instance segmentation. *CoRR*, abs/1603.07485, 2016. 2

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 1, 3

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1, 2, 6

[16] 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, 2023. 2, 3

[17] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. *arXiv preprint arXiv:2305.13310*, 2023. 2

[18] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *International Conference on Learning Representations*, 2020. 1

[19] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images, 2023. 2

[20] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation, 2017. 2, 1

[21] 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 *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 4

[22] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Alexander G. Schwing, and Jan Kautz. Ufo^2: A unified framework towards omni-supervised object detection. *CoRR*, abs/2010.10804, 2020. 1, 2

[23] Gyungin Shin, Weidi Xie, and Samuel Albanie. Namedmask: Distilling segmenters from complementary foundation models, 2022. 2

[24] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 1

[25] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. *CoRR*, abs/2012.02310, 2020. 2, 6

[26] Pei Wang, Zhaowei Cai, Hao Yang, Gurumurthy Swaminathan, Nuno Vasconcelos, Bernt Schiele, and Stefano Soatto. Omni-detr: Omni-supervised object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9367–9376, 2022. 1, 2, 3, 6, 8

[27] Junde Wu, Yu Zhang, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation, 2023. 2

[28] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 4

[29] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot, 2023. 2

[30] 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. 1, 2, 6

[31] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[32] Yi Zhu, Yanzhao Zhou, Huijuan Xu, Qixiang Ye, David Doermann, and Jianbin Jiao. Learning instance activation maps for weakly supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2